```
ÚÄÄÄÄÄÄ InformationÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
³                          ºººÛ   ºººººÛ  ºººººÛ   ººÛ       ººÛ     ³
ÃÄÄÄÄÄÄ Systems ÄÄÄÄÄÄÄÄÄÄÄ ºÛ ÄÄ ºººÛ ÄÄ ºººÛ ÄÄÄ ººººÛ ÄÄ ºººÛ ÄÄÄÄ´
³                           ºÛ    ºººººÛ  ºººººÛ   ºººººººººººÛ     ³
ÃÄÄÄÄÄÄ Security ÄÄÄÄÄÄÄÄÄÄ ºÛ ÄÄÄÄÄ ººÛ ÄÄÄ ºÛ Ä ººÛ ÄººÛ ÄººÛ ÄÄÄÄ´
³                          ºººÛ   ºººººÛ  ºººººÛ   ººÛ       ººÛ     ³
ÀÄÄÄÄÄÄ Monitor ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Dedicated to the pursuit of security awareness.............

===
Volume 3 Number 4   October 1993
===

IN THIS ISSUE:


```
**************************************
*                                    *
*  Public Debt Connects to Internet  *
*         by Joe Kordella            *
*                                    *
**************************************
```

Over the past few years, Public Debt computer users have seen a steady increase in the resources made available to them through the various networks to which they are attached. Through the FRCS-80 network it is possible to share mainframe applications developed by Public Debt with our partners at many of the Federal Reserve Bank sites. Our own PDLAN network allows us to share files within our workgroups and among our several sites in Washington and Parkersburg.

Recently, the AIS Security Branch within the Office of Automated Information Systems (OAIS), expanded the range of such resources available to Public Debt personnel by establishing a gateway to the "Internet". The Internet was born about 20 years ago. At that time one of its antecedents, called the ARPAnet, was essentially an experimental network designed to support military research. Sometime later, ethernet technology and Local Area Networks (LANS) became commercially available. Organizations which invested in such tools quickly saw the advantage of connecting their local LANS to the larger ARPAnet and other

similar networks. Benefits included access to shared information and greatly expedited communications throughout the country and the world. Over time, more and more networks were connected to each other and the resultant network of networks became known as the "Internet".

The Security Branch's gateway allows Public Debt users to exchange E-mail with Internet users throughout the world. Users on the system located in Parkersburg can receive mail from individuals throughout the world as user@aisecur.bpd.treas.gov (where "user" is the individual's authorized ID on the Security Branch system.) The gateway also provides access to Internet "News Groups". News groups are the Internet equivalent of CompuServe "forums" or BBS "doors". They are essentially electronic meeting places for people of like interests to swap information and news items about a specific subject of interest. Security Branch's gateway carries news on a wide variety of computer and security related topics. Access to news groups gives Public Debt users access to world class resources, many of whom are willing to share their expertise in a spirit of cooperation and mutual help.

Those desiring additional information on the Public Debt e-mail and news gateway should contact the AIS Security Branch or send them email at kclancy@aisecur.bpd.treas.gov .

********** END OF ARTICLE **********

////////////////////////////////////
/ /
/ Computer Security Day, 1993 /
/ By The Editors /
/ /
////////////////////////////////////

The 6th annual nation-wide observance of Computer Security Day is set for December 1, 1993. The primary goal of Computer Security Day is to focus attention on the vital problem of computer security by encouraging management of computer professionals everywhere to bring extra attention to the issues of computer security.

Last year The Bureau of Public Debt participated by holding a contest to select the "Best Security Slogan" as submitted by the ISSM Newsletter readership. The slogans, plus the names of the submitters, were posted on the bulletin boards throughout Public Debt, also the slogans were printed in the ISSM Newsletter, along with photos of the participants.

This year the Bureau will hold a contest for the "Best Security Poster". The poster can relate to any computer security-related topic. Submit your posters to AIS Security Branch, Poster Contest, Room 107 by March 31, 1994. Posters will be posted on the bulletin boards throughout Public Debt, and all submitters will receive a prize.

********** END OF ARTICLE **********

~~~~~~
~   ~
~ Analysis of Garden Variety Computer Viruses in 5 Minutes   ~
~ (Well, Almost 5 Minutes...)   ~
~   By George Smith, Ph.D.   ~
~   ~
~~~~~~

(George can be contacted on CompuServe at 70743,1711 or via internet at 70743.1711@compuserve.com)

Occasionally, as a network administrator you may run across a virus which isn't covered by any of your current protection schemes.

Lucky you!

In any case, analyzing the virus - once you've isolated it - need not be a traumatic affair, or even necessitate a call to an expert. In most instances, you are fully capable of handling the job. Don't let your mind be gripped by insecurity. Yes, I will say it again: "You, too, have the skill to analyze and disassemble computer viruses!" And this news piece will tell you how to get started.

If you've discovered a virus, your first goal was to get rid of it. However you found it, you've set your colleagues to work eliminating files you suspect or are sure are infected. But you might want more information. The need for analysis and disassembly - or reverse engineering of the virus to the point where you adequately understand its instructions and purpose - arises.

A real world example is the recent spread of the Butterfly virus within the Telemate communications program shareware archive.

Because Telemate is a popular program, nearly everyone who received original copies of the recent version of Telemate also received copies of the Butterfly virus.

Assume that you have users who use Telemate. All might have executed copies of the Butterfly virus. Simple VISUAL scrutiny of the Telemate programs with any common file viewing/listing utility (DOS, Windows, OS/2, PC Tools and Norton Utilities versions all include such tools) would have revealed the following:

```
0380  4E 8D B6 50 02 8D 96 2C-02 52 EB 3C B4 1A BA 80   N..P...,.R.<....
0390  00 CD 21 33 C0 33 DB 33-C9 33 D2 33 F6 33 FF BC   ..!3.3.3.3.3.3..
03A0  FE FF BD 00 01 55 33 ED-C3 0B DB 74 19 B5 00 8A   .....U3....t....
03B0  8E 47 02 B8 01 57 8B 8E-48 02 8B 96 4A 02 CD 21   .G...W..H...J..!
03C0  B4 3E CD 21 33 DB B4 4F-5A 52 B9 07 00 33 DB CD   .>.!3..OZR...3..
03D0  21 73 18 E9 9F 00 FF 47-6F 64 64 61 6D 6E 20 42   !s.....******* B
03E0  75 74 74 65 72 66 6C 69-65 73 FF 8B D6 B8 02 3D   utterflies.....=
03F0  CD 21 72 B5 8B D8 B4 3F-B9 04 00 8D 96 04 01 CD   .!r....?........
```

The above shows a portion of a program infected with the Butterfly virus. Note the text "******* Butterflies" (Ed note: text has been sanitized, code is

unchanged). This is not standard fare for any program and should raise an eyebrow, unless everyone on your staff is possessed of an unusual sense of humor. Programming a text searching tool for "******* Butterflies" would uncover any file with the embedded string on a searched disk, i.e, any file infected with the Butterfly virus.

In the real world, your job would have been done!

But you might suspect that not everyone in your building has gotten the alert, in which case you would expect to hear from Butterfly once or twice again. You might want to know some more information about the virus.

You would then use a commercially available disassembler to quickly translate the virus into its basic instructions. One assembler for the job is Sourcer (V Communications, Walnut Creek, CA), but there are others equally good.

The first step would be to take an original file infected with Butterfly and place it on an isolated machine for virus testing. In the same directory as the original Butterfly-infected file would be placed "bait" .COM and .EXE programs which contain nothing more than hexadecimal "00" or "90" words. (Utilities exist to create such programs. In addition, I have included the assembly language code for such a "bait" file at the end of this article.)

The reason for the bait file is so that the virus can be clearly seen in an infected file. Any instructions written by the disassembler will then belong ONLY to the virus. This simplifies analysis, since you won't have to interpret whether the disassembler's results refer to the infected file or the virus.

To infect the bait files, execute the virus infected file. If it is a direct action virus, it will add itself to one or more of the baits. A simple directory listing will reveal a file size change if this is the case. If the virus is a memory resident infector, you will have to execute the virus-infected file and then execute the baits consecutively. Because some viruses have what are called by the vulgar computer press "stealth characteristics," immediately doing a directory listing of the files may not show any change. Such a "stealth" virus, when present in memory, will confuse the machine sufficiently so that such a directory listing is useless.

Reboot the test machine CLEAN with a write-protected system disk. Now, do a directory listing. All changes in bait file size will appear unless the virus is a RARE overwriting stealth virus. These cases are so odd, I feel secure in saying you need not worry about them at all. So we won't.

Instructing the disassembler to analyze the Butterfly-infected file will, if we use Sourcer as an example, produce a summary of key virus intstructions labelled the "interrupt usage list."

It looks like this:

```
Interrupt 21h : DOS Services  ah=function xxh
Interrupt 21h :  ah=1Ah  set DTA(disk xfer area) ds:dx
Interrupt 21h :  ah=3Dh  open file, al=mode,name@ds:dx
Interrupt 21h :  ah=3Eh  close file, bx=file handle
Interrupt 21h :  ah=3Fh  read file, bx=file handle
Interrupt 21h :  ah=40h  write file  bx=file handle
Interrupt 21h :  ah=42h  move file ptr, bx=file handle
Interrupt 21h :  ah=4Fh  find next filename match
```

Interrupt 21h : ax=5701h set file date+time, bx=handle

Because you've used a bait file to examine the virus, these raw instructions belong to Butterfly. They are not as cryptic as they initially appear.

You may have already identified the individual in your organization who is the assembly language tinkerer. He can tell you what the above instructions mean. In lieu of that, you can use the "New Peter Norton Programmer's Guide to the IBM PC & PS/2" or the "MS-DOS Encyclopedia" for an interrupt usage list which contain easily read tables that translate the above interrupts and their functions into meaningful English.

Using either of these references, you see the analyzed program:

--opens files (function 3Dh) very common, a virus has to open a file before infecting it.

--read file (function 3Fh) very common, a virus has to read a portion of the file to determine if it has or has not already infected it.

--write to file (function 40h, the virus-programmer's magazine 40Hex is named after this), very common, a virus has to write its code out to the potential host.

--find next filename: match (function 4Fh) very common for direct action viruses like Butterfly. The filename function points to the file mask, *.COM, embedded in the virus code. The virus, therefore, seeks .COMfiles to infect.

For a virus, this is very straightforward. And it is a commonplace, real world example. Butterfly appears to do little more than look for .COMfiles to infect. As the virus doctor, you would be alert for functions which check system time, date, DOS version or any other particular variable on a machine. If such were also included in the above list, you would presumptively conclude it has NO use beneficial to your machines and might indicate an activation trigger which would cause the virus to do something even more unpleasant than merely replicate.

For example, such antisocial behavior would be shown by an appearance in the above list of an occurrence of interrupt 13h - an absolute write to the disk drive. In viruses, this is almost always associated with an attempt to destroy all the data on an affected machine. It is not critical to know when such an event is triggered. You SHOULD assume that it could happen any time the virus is called.

It's also quite possible you might encounter an encrypted virus. One example, a German virus called SANDRA, was quickly disassembled by many experts when it appeared early in 1993.

Using Sourcer to analyze SANDRA was a little different than Butterfly. The interrupt list, in this case, was nonexistent, because the majority of the virus was encrypted and hidden from cursory analysis by a dissasembler.

The initial Sourcer analysis looked like gibberish, a small segment of cryptic assembly code instructions, then some words that almost appeared to be English and quite an oodle of hexadecimal values arrayed in columnar "define byte" (or "db") format.

This immediately told the experienced that SANDRA was encrypted, and rather weirdly at that.

The next step, then, was to trick the virus into decrypting itself and then writing the "plain text" version to disk. This was simple in theory, only slightly more difficult in practice. Envision that the portion of the virus researchers wanted to execute was the decryptor loop, a small stretch of instructions which unscrambled the virus in memory. Might not that segment of cryptic assembly code that Sourcer produced on its first pass contain the keys to the decryptor? Yes, good guess! And it looked like this:

```
    seg_a           segment byte public
                    assume  cs:seg_a, ds:seg_a
                    org     100h

                    sandra          proc    far

   3C44:0100                    start:
   3C44:0100  F8                            clc                             ;
Clear carry flag
   3C44:0101  E8 002F                       call    sub_2 ;<----FIG. 1
   3C44:0104  FB                            sti   ; Enable interrupts
   3C44:0105  F8                            clc   ; Clear carry flag
   3C44:0106  <--execute to this address     jmp     loc_6   ;*(027C)
   3C44:0106  E9 73 01                        db      0E9h, 73h, 01h
   3C44:0109  3C              data_3          db      3Ch                     ;
xref 3C44:013D
   3C44:010A  00              data_4          db      0                       ;
xref 3C44:0149
```

You notice that SANDRA starts by calling a sequence of instructions dubbed "sub_2" (see FIG 1.) by Sourcer. Looking down the listing (which is not included here) you see that "sub_2" is another segment of plain-text assembly code. This was the viral unscrambler and when we returned from it, the virus was unencrypted and ready to do its work. The next job for SANDRA, then, was to begin its infection. Looking at the assembly commands above, you see SANDRA jumps (jmp) to a new location, which looked encrypted in the listing researchers started with.

The idea they uses was that by executing the virus right up to the "jmp," it was possible to get SANDRA to translate itself in memory without it looking for a file to infect, infecting that file and regarbling itself. This was an easy task to accomplish with any software debugger. I used the ZanySoft debugger program because it's almost idiot-proof and requires little input.

I started the ZanySoft debugger by typing:

```
C>ZD86
```

ZanySoft is menu driven. Using its "File" drop-down menu to load the SANDRA virus-infected file, I brought up its "Run" menu and double-clicked on the "go to xxxx:xxxx" command. This told ZanySoft to execute the loaded program to a

certain address - which it prompted me to supply -- and stop. The address needed was the one corresponding to the "jmp" in the above listing. Sourcer had supplied it, and it is ear-marked in the diagram: 0106.

By typing in 0106 at ZanySoft's prompt and hitting <enter>, the SANDRA virus was decrypted. Returning to the "Files" menu and selecting the option, "Write to .COM." wrote the SANDRA virus to the disk from memory, in its "plain-text" or unencrypted form.

Disassembling this version of SANDRA produced an interrupt table list similar to that obtained from Butterfly, because THIS time the virus was unencrypted, its instructions wide open to analysis.

There are many other variants on this theme. Some virus programmers attempt to disguise their creations with "tricks" which attempt to confuse disassemblers. I can say with some assurance that these attempts are not particularly successful and that the odds you will run into such an animal are less than being run over by car.

Is all this so mysterious? YES, I hear you say. Perhaps you feel a little overwhelmed. But if you sit back and look at the examples of Butterfly and SANDRA once again, even though you think you know next to nothing about assembly language or virus code, with persistence, you will be able to use a disassembler listing to make some informed deductions about any virus. And you'll be able to do it in about five minutes, with a little experience.

```
;500+ byte "bait" file suitable for trapping .COMfile infecting viruses
;Assemble with Turbo Assembler or shareware A86 assembler
;example command lines:  A86 bait.fil bait.com
;                     or TASM bait.fil
;                        TLINK /x /t bait

code   segment
       assume cs:code, ds:code, ss:nothing

       org 100h

  start:    jmp  term
            db   500 dup (90h) ;change number preceding "dup" to any value
  host:     db   'Hello, virus!',0  ;<---simple marker

  term:
            mov  ah, 4Ch
            int  21h

            code ends
                 end start
```

Bibliography:

1. Hruska, Jan. "Computer Viruses And Anti-Virus Warfare". 1992. Simon & Schuster/Ellis Horwood.

2. Ludwig, Mark. "The Little Black Book of Computer Viruses." 1991. American Eagle, Inc. (Tucson, AZ).

3. Norton, Peter & John Socha. "Peter Norton's Assembly Language Book for the IBM PC." 1989. Brady Books.

********** END OF ARTICLE **********

??????????????????????????????????????
?   ?
?   When It Comes to Viruses....   ?
?   WHAT'S A USER TO DO?   ?
?   by The Editors   ?
?   ?
??????????????????????????????????????

When it comes to viruses, what is a user to do? The previous article on viruses may seem rather technical for the everyday computer user but may also demonstrate to some that understanding viruses is not as difficult as one might imagine. To the user of PC's in Public Debt, your interests probably rest in trying to understand how to protect yourself from viruses or learning how not to introduce viruses to others in Public Debt and those we interact with. Prevention can be as easy as contacting your ISSM to find out what types of controls they have put in place for your area and ensuring you are complying with the procedures they have established.

ISSMs throughout Public Debt have installed software for users, provided scanning of new diskettes before they are installed on user's machines and even published their own information on the topic.

ISSMs are responsible for establishing the virus protection programs in their areas. Give them a call if you have any questions. As a user, you also have a responsibility to report "virus-like" activity to your ISSM. The Insert in this newsletter contains the procedures put in place by the AIS Security Branch and Public Debt's ISSM Team for handling viruses. The sooner a possible virus is reported, the sooner a response team can be formulated and the problem resolved.

Do your part and know your responsibilities. Review the procedures and contact your ISSM with any suggestions or questions you may have.

********** END OF ARTICLE **********

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
%                                %
%      WELCOME ABOARD!           %
%       By The Editors           %
%                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A New Employee Joins the Ranks of the Security Branch We'd like to extend a welcome to Andy Brinkhorst, the newest member on the AIS Security Branch team. Andy comes to us from Farmers Home Administration, Department of Agriculture, where he was Assistant Information Resource Manager for the State Office in Lexington, Kentucky.

At FmHA, Andy was responsible for providing support and training for over 60 County and District offices, as well as developing systems for use at the State Office level. He also provided training and support to the State Office Staff, as well as serving as the Deputy Security Officer for FmHA in Kentucky.

Prior to his career in the public sector, Andy was self-employed as a consultant, providing computer and network support for individuals and small business operations. Andy started this business while in the final year of obtaining his B.S. degree in Computer Science/Information Systems from Marshall University in Huntington, WV. Andy says that even though the bluegrass of Kentucky is nice, he's happy to be back here, having grown up in Vienna and graduating from Parkersburg High School.

We're all glad that it was possible to bring a West Virginia native back home to the Mountain State, and wish him the best of luck in his new position.

********** END OF ARTICLE **********

```
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&                               &
&         Jim's Corner          &
&       By Jim Heikkinen        &
&                               &
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

FORMAL TRAINING: Fiscal year 93 training contracts are generally in place and I can announce tentative dates for the following classes: ACF2 (Washington) November 15-19 Novell Netware Security (Parkersburg) November/December 1993 SNA/APPN/APPC December 6-10 Voice Communications (Intro) November 15-19 Voice Communications (Advanced) November 29-December 3

AUDIO-VISUAL DEPT.

Best bet video for this quarter: "Invasion of the Data Snatchers" Five episodes on one 20-minute VHS cassette that highlight methods of data theft.

Best bet for late night reading: "Terminal Compromise" - by Winn Schwartau A fictional account of a series of computer terrorist attacks on the United

States. A blend of political extremists and technical mercenaries spin a web of deceit and intrigue that threatens this country's 70 million computers.

******************** END OF ARTICLE ********************

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                                    !
!  COMPUTER SPEAK - Computer Terms and Definitions   !
!                ISSM Staff                          !
!                                                    !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

ARPAnet n. A network established by the Advanced Research Projects Agency (ARPA) of the Department of Defense so that information can be exchanged between the computers of universities and defense contractors.

GATEWAY n. A connection between dissimiliar communications networks.

COMPUTER VIRUS n. A program that searches out other programs and 'infects' them by embedding a copy of itself in them. When these programs are 'run' they performed a pre-programmed set of instructions. For example, the program may erase all the data on your hard drive.

ISSM n. Information Systems Security Manager. Each area in Public Debt has a security manager assigned who is responsible for establishing security safeguards in their area of responsibility.

END USER n. The person that works directly with the computer equipment in order to complete their assigned job duties. This is the most important person in the computer security program. This person is you!

******************** END OF ARTICLE ********************

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                           X
X   Anti-Virus Procedures   X
X     By The Editors        X
X                           X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

1. End user encounters problems on his/her PC which suggest the possible presence of a virus. The PC is left on but the user should not interact with it further.
2. End user contacts his/her ISSM requesting guidance.
3. ISSM visits the end user's PC with a repair "kit" including a write-protected virus scanning disk. If the virus scanning reports the

presence of a virus, the ISSM will notify the Help Desk.
4. The Help Desk will immediately notify the Manager, AIS Security Branch by telephone and provide the name of the affected ISSM.
5. The Security Branch will direct all virus recovery steps by:
Calling together an emergency response team to manage recovery if necessary.

The team may consist of:

-LAN personnel.
-Communications personnel.
-LAN administrator for infected server.
-ISSM of infected area.
-Help Desk representative.
-Others that are required.

Prescribing the procedures for scanning other machines close to the infection;
Notifying the Network Section of the Communications Branch and Help Desk if the infected PC has access to the PD LAN server or mainframe;
Instructing the Network Section to isolate segments of the LAN which may be infected;
Entering necessary data in the Virus table of the SOMS system;
Compiling data related to the severity of the infection, the resources required to recovery from it and other pertinent information;
Contacting industry experts as required to develop and/or procure a strategy for recovering from the infection;
Notifying the ISSM community of the infection via the most expeditious means (i.e., E-Mail, BBS, Telephone) and alerting them to the potential for diminished network services.
6. If network resources are involved Network Section personnel will scan and clean network servers and report their findings to the Security Branch. Servers which were infected will not be placed back on-line without the approval of the Security Branch Manager.
7. PC resources which have been infected will be scanned with a write-protected disk by the ISSM owning those resouces. PCs which were infected will not be placed back on-line or logged into the network without the approval of the Security Branch Manager.
8. Once all infected resouces have been certified scanned and clean by the ISSMs and the Network Section, the Security Branch Manager will approve placing the servers and PCs back on-line.
9. The Security Branch will alert the Help Desk that virus affected resources are being placed back on-line. The Help Desk will make all appropriate notifications.
10. The Security Branch will issue a report to the Assistant Commissioner, OAIS, which summarizes the virus outbreak and associated cleanup efforts.
11. If a message notification is given to the Command Center (Help Desk) via automated cc:Mail virus administrator box refer to step 4 of this procedure.

********** END OF ARTICLE **********

>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>   >
>   TOKEN TRAINING STEPS   >
>   By The Editors   >
>   >
>>>>>>>>>>>>>>>>>>>>>>>>>>>>

1. Enter your logon ID and your password.
2. Turn your Token on..."EP" should appear in the window.
3. Enter your 4-digit P-I-N..."ECH" will appear in the window.
   (Remember...your P-I-N is secret...keep it safe!)
4. Enter the challange number from the PC. Press "E" on the token.
5. Enter the 8-digit number shown in the token window as your dynamic
   password.

********** END OF ARTICLE **********

.....
The AIS Security Branch Runs an Electronic BBS. Give us a call at (304) 480-6083. An electronic version of the ISSM is posted on the board and can be downloaded. Articles in the electronic version may include more detail in that we are not limited by space constraints as we are in the paper copy.

The Information Systems Security Monitor is a quarterly publication of the Department of Treasury, Bureau of the Public Debt, AIS Security Branch, 200 3rd Street, Parkersburg, WV 26101 (304) 480-6355 Editors:

Ed Alesius
Andy Brinkhorst
Kim Clancy
Mary Clark
Jim Heikkinen
Joe Kordella
.....

>>>>>>>>>>>>>>>>>> END OF NEWSLETTER <<<<<<<<<<<<<<<<<<