Fortinet-1021

April 1990 ISSN 0956-9979

VIRUS BULLETIN

THE AUTHORITATIVE INTERNATIONAL PUBLICATION
ON COMPUTER VIRUS PREVENTION,
RECOGNITION AND REMOVAL

CONTENTS

EDITORIAL

Burger's Legacy

In October of last year *Virus Bulletin* published a review of a book entitled *Computer Viruses: A High Tech Disease* which was first published in 1987 by *Data Becker GmbH* of West Germany. The book by Ralf Burger contains numerous source code listings of viruses (most of which do not work) and includes a recognisable listing (albeit crippled) of the Vienna virus in assembler source form. Of this, Jim Bates, who reviewed the book for *Virus Bulletin* noted:

"This immediately provokes questions concerning the wisdom of publishing such listings in this way and thereby allowing any irresponsible individual to produce his own virus code without thought for the consequences. Drawing a parallel, I wonder what the general reaction would have been to a book listing details of how to make explosives or assemble bombs, however amateurish."

About two months before this review appeared, a computer virus called Vacsina was discovered in West Germany and disassembled by Christoph Fischer at the *University of Karlsruhe.* Vacsina incorporates certain elements contained in Burger's listing. This gave rise to suspicions that virus writers were using Burger's book. Vacsina, it now transpires, is the sixteenth in a series of fifty viruses which originate in Bulgaria. These viruses maintain upward compatibility which makes it possible to analyse the evolution of the series. Both Vacsina and Yankee Doodle (reportedly number 38 in the series) escaped into the 'wild' and are now causing infections in Eastern Europe and Germany.

According to Vesselin Bontchev, who has analysed the series, these viruses are written by an individual with the initials 'TP'. TP also wrote New Vienna versions VHP-367, VHP-353 and VHP-348 which were modified viruses based on Berger's source code listing. It is not possible to state with certainty that TP used Burger's listing to write these mutations or whether he used an existing variant of the Vienna virus based on it.

The 1260 virus (*see VB, March 1990*) which is believed to originate from a different source, is also based on the Burger listing, indicating that at least one other virus writer is using Burger's published source code. Two further viruses, Ghostballs and Lisbon, also use the book's source code.

This is certainly an indictment of Data Becker, the publisher of the book, and equally a confirmation of our book review which clearly stated: *"There is no doubt that some damage will result from attempted copies of the Vienna virus listing; and the pseudo-official status afforded to hackers and software terrorists may tempt some readers to try such activities".*

The explicit nature of Burger's book and the irresponsible attitude of his publisher was reconfirmed while researching the IBM Christmas Tree (CHRISTMA EXEC) program, details of which appear on page 9 of this month's edition. David Ferbrache, who maintains an archive database on attack programs at *Heriot-Watt University* directed me to page 193 of the book - it contained a complete source code listing for the program which, according to Burger was included for 'documentation purposes'!

In September 1989, the *International Federation of Information Processing* passed a unanimous resolution calling on publishers to refrain from publishing virus listings. It is now clear that published source code has been used to write computer viruses. Authors and publishers should take note and proceed in a responsible manner. Claims that such details are provided for educational or documentary purposes will no longer suffice.

Mac Threats: The ZUC Virus

A report of a new Macintosh virus has appeared on *Compuserve.* The ZUC virus attaches to the CODE 1 resource in each application. When active the virus will scan the entire disk or use the desktop APPL resources to locate applications to infect. ZUC bypasses software right-protection on volumes, locked resource forks, and protected CODE 0/1 resources. The virus causes a screen display consisting of the cursor moving diagonally down the screen two minutes after an infected application is launched.

Jeff Schulman (author of *Virus Detective* software) has published a search string which can detect ZUC.

```
Resource Start & Pos - 1256 & Data
082A#F1655#30832;For finding ZUC.Virus
```

Editor's note: Symantec UK has released SAM v 2.00 and has announced a newsline service which will provide instructions to update SAM as new viruses and Trojans appear. Tel UK 0628 776343. Meanwhile, Virex v.2.51 protects against both Trojans reported in last month's *VB* as well as a new Trojan called 'Virus Info'. (*see page 16*).

VIRUS BULLETIN - CHANGE OF ADDRESS

Please note that from Monday, 2nd April 1990 our new address will be:

Virus Bulletin Ltd.
21 The Quadrant
Abingdon Science Park Tel 0235 555139
Abingdon OX14 3YS, UK. Fax 0235 559935

TECHNICAL EDITORIAL

Friday The Thirteenth: A Day Of Reckoning?

Computer viruses received considerable publicity last year, when the media predicted a major disaster on Friday the 13th October - the day when computer viruses would attack all the personal computers of the world.

Many of the experts in the field did their best to explain that the situation was far from being that serious, but were incapable of calming peoples' nerves. The positive side of this hysteria was the number of backups taken before October 13th, in many cases by people who normally do not take any backups of their work.

As expected, no major virus outbreak struck on October 13th, which unfortunately convinced some people that computer viruses were just a 'media- phenomenon'.

Friday the 13th is coming up again now in April, so this article will examine the three 'Friday the 13th' viruses and try to determine whether there is any cause for alarm.

The South African Virus

The South African virus is one of the oldest viruses around, but nevertheless, it is very rare. The main reason for this is that the virus is rather primitive and will not spread as effectively as many later viruses. The virus does not stay resident, but when an infected program is run it will seek out one or more programs to infect. The virus appends itself to the infected file, and adds a JMP at the beginning.

Several variants of the virus have been reported. The length is 415-419 bytes depending on which assembler was used to create it. If an infected program is executed on October 13th, it will be deleted by the virus, but one variant, Virus-B with a length of 544 bytes is also known, where the deletion part has been disabled, as the virus is intended for demonstration purposes only.

As reports of infections by this virus are few and far between, it should not be the cause of any great concern.

Jerusalem

Jerusalem (often called the Friday the Thirteenth or the Black Friday virus) is among the most common viruses and is circulated worldwide. As described in the July 1989 edition of the *Virus Bulletin*, it activates on every Friday the 13th, deleting programs run on that date. The Jerusalem virus will be the major source of concern on April 13th. Detection and disinfection programs for this virus are readily available and the use of such software is strongly advised. As this virus is so well known, it will not be discussed further.

Datacrime

The Datacrime virus caused most of the hysteria last October, despite the fact that it was not nearly as widespread as the Jerusalem virus. The reason was that it would do more damage, because instead of deleting just a single program, it would reformat the entire disk. (*For a full description of Datacrime, see the August 1989 edition of VB*).

As expected the virus only struck in isolated locations on October 13th 1989 and did not cause the widespread devastation predicted by the media.

The major reason the virus had such a limited geographical distribution, was simply that it had not been given enough time to spread. With the exception of viruses like Tenbyte, (described in this edition) it usually takes a virus at least a year to spread around the globe. The Datacrime virus displayed a message when it activated, saying that it had been released on March 1st, 1989, which may well be true.

Unlike the other two viruses described in this article, Datacrime does not activate on every Friday the 13th. It is set to activate on October 13th every year and stay active until the end of the year. It not possible, therefore, to bypass this virus by changing the current date on the computer to October 14th.

Strictly speaking, Datacrime is **not** a 'Friday the 13th' virus, but it is included here because of its central role in the events of last October. The important question now is whether we have anything to fear from this virus on October 13th of this year.

Datacrime reported its presence on all infected computers last year by low-level formatting the hard disk. It is probable that the original virus has been completely eradicated. However, somebody may have released it again this year in the hope that it will become sufficiently widespread to cause panic on October 13th 1990.

Other variants of the virus may also be around. An improved variant, called Datacrime II appeared within months of the original virus being discovered. Unlike the original virus it is able to infect .EXE files as well as .COM files. The same is true for the latest version, DataCrime II-B, which has been reported to activate on a different date, although we are not yet able to confirm this, as the virus is still awaiting disassembly.

VIRUS REPORT

Fridrik Skulason

Tenbyte Virus - Worldwide Distribution Within Hours

Note: *In the March edition of Virus Bulletin, this virus was listed as 'Valert'. Its name has been changed to 'Tenbyte'.*

Tenbyte is a resident .COM and .EXE file infecting virus, with a length of 1554 bytes. **It is not of particular interest from a technical viewpoint but the reason for examining it in detail is that it received a worldwide distribution recently.**

The virus was distributed in February 1990 via a US electronic mailing list called VALERT-L which is dedicated to urgent computer virus alerts. Most of the recipients of VALERT-L are on BITNET/EARN (US/Europe), but some of them use EUNET (Europe), INTERNET (US) and JANET (UK).

The method of distribution was very efficient - within a few hours of the original posting, the virus was in the hands of more than 500 people around the world.

It is highly unlikely that any of the recipients will execute the virus by mistake because it was clearly labelled as such. *However, it is probable that someone will distribute this virus deliberately, possibly in modified form.*

Tenbyte will activate on September 1 every year and will stay active until the end of the year (December 31). During its active period it will intercept any write operation and modify the data written, by removing the first ten characters.

Operation

When an infected program is executed, the virus will try to detect whether any program is monitoring INT 21H (single step) or INT 3H (breakpoint), with the purpose of frustrating anyone attempting to trace the execution of the virus with a debugger.

Just like the 'Stupid' virus from Israel, 'Tenbyte' installs itself into a fixed location in memory at 9800:0000. This is 32K below the the 640K boundary. The virus will not work at all on machines with less than 640K - they will just 'hang'. *Some programs use this part of memory, so they will overwrite the virus and also cause a system crash*. The virus is also invisible to any memory mapping utility.

Tenbyte next intercepts INT 21H, in a unique way. The interrupt table is not modified but the program currently in control of INT 21H is searched for a code fragment containing the following instructions:

```
cmp ah,SOME_VALUE
conditional branch to SOME_LABEL
```

If these instructions are found within a range of 128 bytes from the INT 21H entry point, they are replaced with a FAR JMP to the new INT 21H function. When the virus later transfers control back to the original program, it will either perform a FAR JMP to SOME_LABEL or the instruction following the conditional branch, depending on SOME_VALUE.

This makes it impossible to determine whether the virus is present by examining the interrupt table. However, the code described above is not present in all versions of DOS and on some computers this virus will not activate.

If the virus is installed successfully, it will restore the infected program to its original state and transfer control back to it.

INT 21H Routine

Tenbyte only intercepts two functions of INT 21H. Like most other resident viruses it intercepts function 4BH (load/execute program). It stores the original attributes and creation date of any program executed before checking the first two bytes to see whether the inspected file is .COM or .EXE. This is necessary because the file extension is not always valid - a 'true' .EXE file could have the extension .COM and vice versa.

To determine whether the program is already infected, the virus checks the two bytes at offset 2-3. If they contain 2E 01, the file will not be infected.

EXE files are infected by appending the viral code to the file and modifying the header as necessary, using a method similar to that used by a number of other viruses. COM files are infected in a somewhat unusual manner - the virus code is appended to file in the usual way, but instead of simply placing a 3 byte JMP at the beginning, the first 12 bytes are overwritten with a short routine which performs a FAR JMP to the start of the virus. COM files of less than 1000 bytes will not be infected in order to reduce the chance of the virus being detected.

Although the virus' length is 1554 bytes, infected files may grow by as much as 1570 bytes because the file is first padded so that its length becomes a multiple of 16 bytes. This makes it possible for the virus code to start at Offset 0 in the virus segment, both in .EXE and .COM files, which eliminates the need for complex relocation instructions.

The second function which Tenbyte intercepts is INT 40H (Write). If the current month is September, October, November or December, the first ten bytes of everything written to a file will be lost. This effect is produced by adding 10 to the address of the data which should be written. As the number of bytes is not changed, ten 'garbage' bytes will be added to the end.

This incident of virus distribution by electronic mail again emphasises the need to enforce strict control over the uploading, downloading and transfer of executable files (programs) on electronic mail and conferencing systems and bulletin boards. In this instance, the virus was clearly marked as such. However, there have been numerous cases of viruses and Trojans residing on BBSs. Anyone considering downloading a program from a BBS must understand that the integrity of that program is entirely dependent on the vigilance of the System Operator.

The VALERT-L incident has also demonstrated the speed with which viruses can be propagated by e-mail systems. Hundreds of users intentionally downloaded the Tenbyte virus and we can only assume that not all of these people will have done so out of innocent interest. The virus will almost certainly appear in the 'wild' and it is probable that mutations will occur, some of which may be destructive.

IBM PC VIRUSES

Amendments and additions to the *Virus Bulletin Table of Known IBM PC Viruses* as of 26 March 1990. For information on all known IBM PC viruses, refer to *Virus Bulletin*, March 1990.

Hexadecimal patterns can be used to detect the presence of the virus with the 'search' routine of disk utility programs such as The Norton Utilities or your favourite disk scanning program.

Datacrime II-B - CEN: Minor variant of Datacrime II

```
Datacrime II-B 2BCB 2E8A 0732 C2D0 CA2E 8807
43E2 F3BD; Offset 01B
```

Ohio - DR: Related to the Den Zuk virus, probably an older variant.

```
Ohio FAFA 8CC8 8ED8 8ED0 BC00 F0FB E845 0073;
Offset 02B
```

Old Yankee. Change of identification pattern.

```
Old Yankee 03F3 8CC0 8904 0E07 53B8 002F CD21
8BCB; Offset 009
```

Sunday - CER: Variant of Jerusalem. Infective length is 1631 bytes (EXE) and 1636 (COM). Activates on Sunday and displays the message "Today is SunDay! Why do you work so hard? All work and no play make you a dull boy.".

```
Sunday FCB4 FFCD 2180 FCFF 7315 80FC 0472 10B;
Offset 095
```

Taiwan - CN: Adds 708 bytes to the beginning of infected files and 395 bytes to the end of them. Easily detected because it often destroys infected programs making them 'hang'.

```
Taiwan 07E4 210C 02E6 21FB B980 0033 F6BB 8000;
Offset 0A0
```

Tenbyte, formerly Valert, - CER: Adds 1554 bytes to infected files. Will activate on September 1st of any year. It will corrupt data being written to disk. Was distributed by accident on the VALERT-L electronic mailing list.

```
Tenbyte 1E0E 1F8D 36F7 04BF 0001 B920 00F3 A42E;
Offset 0
```

Zero Bug / Palette - samples obtained are identical.

REPORTED ONLY

512 - confirmed as alias for 'Number of the Beast'

651 - confirmed as alias for 'Eddie-2'

5120 - ?

1702 - CR: A new variant of Cascade.

June 16th - ?

Saturday 14th - ?

Virdem - ?

WORM PROGRAMS

David Ferbrache

DECnet - Inecurity Through Default

The previous article in this series (*see VB, January, 1990*) addressed the UNIX ® operating system interconnected via the Internet, JANET and UUCP protocols. This article considers how the principles of security extend to DEC systems under VMS interconnected via the proprietary DECnet protocols.

Two major occurrences of worms propagating via DECnet networks have been reported, namely:

Father Christmas Worm 23rd December 1988
(distinct from the Bitnet chain letter)

Worms Against Nuclear Killers
17th October 1989

In both cases the carrier networks were the NASA SPAN and *Department of Energy* HEP networks. Both worms exploited default system configurations which included well known DECnet default accounts, coupled with a lack of auditing and constraint of remote operations.

Father Christmas Worm

This worm struck HEP/SPAN systems worldwide on the 23rd December 1988. The worm DCL command script, entitled HI.COM, was designed as a harmless prank. In particular it differed from the Internet worm by generating only one copy on any host system.

The worm executed a command procedure which:

1. established a process MAIL_178DC

2. attempted to propagate to a area/host node selected on the basis of permutated date/time

3. used the default DECnet account to copy the worm script to the chosen system

4. used the TASK 0 DECnet feature to remotely invoke the copied script file

The worm waited until midnight on 24th December 1988 before mailing the message:

```
HI,
How are you? I had a hard time preparing all the
presents.
It isn't quite an easy job. I'm getting more and
more letters ...
Now stop computing and have a good time at home!!
Merry Christmas and a Happy New Year
Your Father Christmas
```

The worm script additionally contained code to mail a message to node 20.117, user PSOLIDE containing the system id string.

WANK worm

On 16th October 1989 the SPAN network control node detected a worm replicating on the network, the worm only affected DEC VMS systems and propagated via DECnet protocols, not TCP/IP protocols. If a VMS system had other network connections, the worm did not take advantage of them. At least two versions of the worm exist and CERT warned that further versions may be in circulation. The worm:

1. Tested for a further copy of itself and if present the worm would delete its source script and terminate. All instances of the worm were identified by process names containing the characters 'NETW_'. A quick check for infection is thus to look for a process name starting with 'NETW_' using a SHOW PROCESS command.

2. Changed the default DECnet password to be a random string of at least twelve characters.

3. Mailed the new password to node 6.59 (SPAN), user GEMPAK.

4. Changed its name to be 'NETW_' with a trailing random numeric id.

5. If operating with SYSNAM privilege the system announcement banner is changed as shown in Figure 1.

6. If operating with SYSPRV privilege the worm disabled mail to the system account and then modified the system login command procedure to appear to delete all user files.

7. Attempts to modify the FIELD account password to allow logins from all locations with full privileges.

DECnet users are advised to upgrade to DECnet VAX version 5.2. This configuration provides options to purge default accounts and incorporates computer generated passwords. Version 5.2 invalidates both worm programs described in this article and comprises a significant enhancement to system security. Ed.

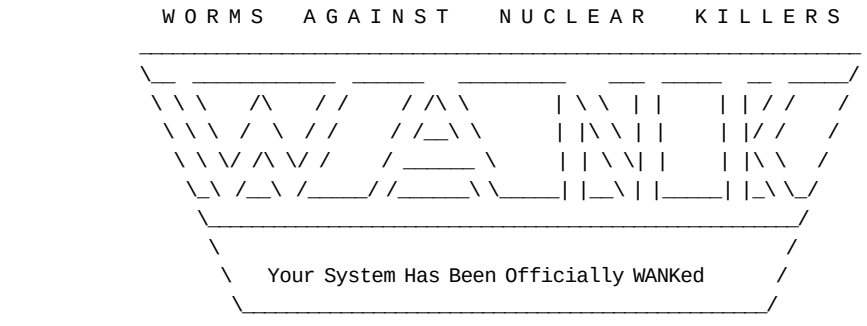

Figure 1.

8. Enters a loop whereby it will pick a node number at random. It uses PHONE to obtain a list of active users on the remote system and users on selected nodes are phoned at random. It also uses the RIGHTSLIST.DAT file and a list of default names to attempt remote access. Two forms of passwords are attempted, namely blank and the userid. (*Essential checks should be made for accounts with blank passwords, or passwords which are the same as userids. These should be invalidated. ed.*) If a privileged account is located (with access to SYSUAF.DAT) the worm is copied to that account, otherwise to a non-privileged account on the remote system.

Detection

Both the Father Christmas and WANK worms use distinctive process names as in-core signatures to prevent reinfection, namely "MAIL_178DC" and "NETW_XXXX". The standard SHOW PROCESS command will indicate the presence of such processes. In the case of the Father Christmas worm testing for HI.COM in the accounting files and NETSERVER logs is also advisable.

Prevention

In both cases the procedures recommended by the US Department of Defense can be adopted to limit external access:

1. Disable the default DECnet account altogether:

```
$ Run SYS$SYSTEM:NCP
Purge Executor Nonprivileged User Account Password
Clear Executor Nonprivileged User Account Password
^Z
```

This does however imply that all remote users must have a legitimate login id or proxy login on the machine.

2. Prevent DCL command procedures from being executed remotely:

```
$ Run SYS$SYSTEM:NCP
Clear Object Task All
^Z
```

then edit the SYS$MANAGER:STARTNET.COM to add

```
CLEAR OBJECT TASK ALL
```

after the line

```
SET KNOWN OBJECTS ALL
```

This will restrict users to execution of those command procedures defined in the DECnet permanent database. This action will not prevent copying of the worm to the system but will prevent its invokation by a remote user.

3. Create a separate account for the file access listener (FAL) with a separate UIC and directory area. This account is created with minimal privileges. Access to the FAL object from the default DECnet account is disabled. The sequence of commands would thus be:

- Run the AUTHORIZE utility to create a new account with minimal permission and privileges restricted to TMPMBX, NETMBX; and a login command script FALLOG.COM described below.
- Invoke the NCP command to define the file access listener object as being accessible by user FAL.
- Create a command script (FALLOG.COM) to carry out logging of all access via the FAL login, this script should include logging of the remote userid and node which is initiating the transfer.

As always the best advice is to read the manual entry, in this case the guide to VMS system security, DEC order AA-LA40A-TE 4/88. Chapter 7 of this work deals specifically with the security of a DECnet node.

Specific Blocks

Specific measures to combat the two worms described above, included in the case of HI.COM ensuring that the worm script file could not be created (similar to the /usr/tmp/sh condom suggested for the Internet worm), and in the case of the WANK worm creating a dummy process which the worm would recognise as an instance of itself. In general most worms do use a signature comprising a special file or process. This signature can be forged as a temporary measure to prevent infection while the attack is being analysed.

HI.COM

```
$ Set Default default-decnet-directory
$ Create HI.COM
$ Stop/ID=0
^Z
$ Set File/Owner=[1,4]/
Protection=(S:RWED,O:RWED,G:RE,W:RE)/Version=1
HI.COM
```

WANK Worm

```
$ Set Default SYS$MANAGER
$ Create BLOCK_WORM.COM
$ DECK/DOLLAR=END_BLOCK
$LOOP:
$ Set Process/Name=NETW_BLOCK
$ Wait 12:0
$ Goto loop
END_BLOCK
$ Run/Input=SYS$MANAGER:BLOCK_WORM.COM/Error=NL:/
Output=NL:/UIC=[1,4] - SYS$SYSTEM:LOGINOUT
```

Conclusion

In general DECnet is an extremely flexible network environment. **It is however vital that passwords are changed regularly, use of default accounts is minimised (in this case the FIELD and DECNET account passwords must be changed from their defaults), and that access privileges from remote systems are identifed and carefully regulated.**

In the aftermath of the DECnet worms the Defense Data Network Operations Center has produced a series of management and security bulletins which are available from the DDN Network Information Center. The Computer Emergency Response Team (CERT) will also be happy to advise and investigate any breaches of mainframe system security.

Internet Worm: Sentenced Deferred

Contact numbers and network addresses for these organisations are given below:

Computer Emergency Response Team
Email: cert@edu.cmu.sei
Telephone: USA 412-268-7090 (24 hours a day)

Defense Communications Agency,
DDN Security Co-ordination Center
Email: scc@mil.ddn.nic
Telephone: USA 800-235-3155

Management bulletins are available by anonymous ftp from the host mil.ddn.nic under pathname DDN-NEWS:DDN-MGT-BULLETIN-nn.TXT (where nn is the bulletin number), security bulletins are occasionally available in the public domain.

At this time no comparable UK governmental organisations exist. I can only hope that in the aftermath of the Internet, DECnet and AIDS Trojan incidents the requirement for a trusted, experienced, central authority to collate and organise countermeasures is realised. The rapid reporting of both security holes (especially important now that the UNIX operating system is expanding with many vendors supplying systems based closely on one of two ancestors, AT&T or Berkeley) and known threats to system integrity (such as the AIDS Trojan) is vital.

A Can of Worms

The IBM Christmas Tree

The first widely publicised replicating network program was the Christmas Tree (CHRISTMA EXEC) which paralysed the worldwide IBM VNet private electronic mail network on 17th December 1987.

It was written by a student at the University of Clausthal-Zellerfield, West Germany. He claimed that the program, written in the VM/CMS operating system language REXX, was designed to send Christmas greetings to his friends within the university. The first known infection was at 1300 GMT on 9th December at Clausthal-Zellerfield (EARN node DCZTU1).

However, the program escaped onto the European Academic Research Network (EARNet) which, ironically, had been established with IBM financial backing and assistance. EARNet is linked to BitNet in the United States which in turn is connected to VNet, a network used by IBM to communicate with customers and suppliers. The The first

infection on Vnet occurred on 15th December 1987. This network was brought to a temporary standstill two days later.

The program read the addresses of communication partners in the user files NAMES and NETLOG and sent a copy of itself to all addresses contained therein. Users found the message 'Let this exec run and enjoy yourself!' posted in their mailboxes. If the program was run the following greeting appeared:

```
'              *               '
'              *               '
'             ***              '
'            *****             '
'           *******            '
'          *********           '
'         *************              A'
'           *******            '
'          ***********                VERY'
'        ***************        '
'      *******************              HAPPY'
'          ***********          '
'        ***************             CHRISTMAS'
'      *******************      '
'    ************************         AND MY'
'        ***************        '
'      *******************            BEST WISHES'
'    ************************     '
'  *************************        FOR THE NEXT'
'             ******              '
'             ******                     YEAR'
'             ******
'                                  '
```

Further investigation of the EXEC file diplayed the line:

```
/*    browsing this file is no fun at all
        just type CHRISTMAS from cms */
```

The program messages terminated with ERASE CHRISTMAS EXEC by which time the program had propagated itself, like a chain letter, to all mailbox addresses contained in the user's address file. In common with PC viruses, this program needed to be executed or 'run' to replicate. Unlike a virus, the Christmas Tree did not incorporate any of its own code into a host program.

The Christmas Tree was the first program known to have caused wide area network paralysis and amply demonstrated the need to educate network users to be wary of running unfamiliar executables (EXEC, CLIST, program), however intriguing a filename or on-screen instruction might be.

Internet Worm: Sentence Deferred

This program was unleashed on the 2nd November 1988 and caused crashes on some 6,200 machines on the U.S.*Defense Advanced Research Projects Agency* (DARPA) Internet. The worm exploited holes in UNIX, common utilities such as *Sendmail* and was targeted at Sun workstations and DEC VAX minicomputers (*see VB, November 1989, pp 2*).

As reported in the *Virus Bulletin* last month Robert T. Morris Jnr has been convicted under the US*Computer Fraud & Abuse Act 1986*. This act makes it a felony to knowingly gain access to federal government computer systems and cause losses exceeding $1,000. Judge Howard Munson had originally scheduled sentencing for February 27th but the sentence date has now been deferred to May 4th. Morris, a Cornell post-graduate student, is the first to be convicted under this act and federal guidelines dictate that he spends at least six months in prison. The maximum sentence for this offence is five years imprisonment, a $250,000 fine plus restitution fees to affected users, which prosecution witnesses testified were equal to approximately $150,000. At no time during the trial did the prosecution indicate that a maximum penalty was being sought.

Morris told the jury that the Internet worm had been designed to make one copy of itself within each node on the network but that a bug in the programming had caused it to replicate out of control causing a denial of service attack. In his own words the program was a "dismal failure". The sentence which Morris receives on May 4th may prove pivotal in the legislative battle against the computer criminal. Calls for severe sentencing of Morris to deter future virus writers and pranksters have been numerous. "Each and every one of you" said lawyer Ellen Meltzer at the trial "must understand that the worm was not merely a mistake; it was a crime against the government of the United States".

Munson rejected defence arguments that the Syracuse, NY, jury had been swayed by prosecution references to Morris as being a 'terrorist'. Some thirteen months before the trial, 35 Syracuse students had been murdered in the terrorist bombing of Pan Am flight 103 over Lockerbie, Scotland.

Morris, currently suspended from Cornell University, is now working as a programmer at Harvard University where he received his bachelor's degree in 1988.

Perhaps the greatest irony surrounding the case is that Morris is the son of Robert T Morris Sr., chief scientist at the U.S. *National Security Agency's Computer Security Division.*

COMMENT

Dr. Peter Lammer

The appearance of second generation computer viruses has sparked a debate about the direction and nature of anti-virus software. In this article, Dr. Peter Lammer offers some thoughts on the future of IBM PC virus detection.

1260 Revisited

The advent of the 1260 virus (*VB, March, 1990*) is something of a milestone. Not only is the virus encrypted, like Cascade (*VB, September 1989*), but the only non-encrypted part of the virus (the decryption routine) is self-modifying. As a result, it is impossible to extract a suitable pattern which can be used reliably to search for the virus. The largest unchanging sequence in the decryption routine of the 1260 virus is three bytes long, which is insufficient for a search. Ten bytes is the minimum practical length for a search pattern, while sixteen bytes should be used whenever possible in order to minimise the number of false positives (i.e. stating that a particular virus has been located when, in fact, it is not there).

How, then, does one go about trying to recognise an encrypting, self-modifying virus such as the 1260? The answer is that one must look beyond search patterns and use other known virus characteristics.

Three concepts can be defined:

The **search pattern**, also referred to as the 'hexadecimal pattern', is a sequence of bytes which are found in a virus. Several different possible search patterns can be extracted from a normal, non-encrypting virus. In a classic encrypting virus such as Cascade, which changes the majority of its own code but which retains some thirty unchanging bytes (which constitute the decryption routine), the search pattern must be chosen from those thirty bytes. In a self-modifying, encrypting virus such as 1260, no part of the virus longer than three bytes will remain constant and the extraction of a suitable search pattern is impossible.

The **virus signature** is a particular set of characteristics which a virus uses to identify itself. This can be a string such as 'sURIV', a value at a particular location or one or more of the file attributes (e.g. the seconds field in Vienna set to 62). Some viruses (a variant of Jerusalem for example) do not recognise their own signature and reinfect executables. This results in programs growing in size until they fail to load into memory.

The **virus identity**, a term which I propose, is a set of all known virus characteristics including the virus signature, structure of infected programs (e.g. JMP to a particular location), type of files infected (e.g. COM but not EXE) and so on.

To identify classic, non-modifying viruses it is sufficient to extract a suitable pattern and use a simple pattern checking program to search for it. With the advent of self-modifying viruses, this is no longer the case. A virus-specific search program must use virus identities rather than virus patterns in order to detect such a virus.

In the case of 1260 the virus identity was summarised at the end of the dissection on page 12 of last month's *Virus Bulletin*. An infected file has the 'seconds' field of the time/date stamp set to 62. It will be a COM file starting with a JMP to a location 1260 bytes before the end of the file. With this information one can recognise the virus.

Camouflage And Concealment

Will future viruses have a recognisable identity of this sort?

Any virus needs to check for its own presence using some sort of signature if it is to avoid a continual reinfection which would betray it relatively quickly through the readily observable increase in the size of infected objects. A self-encrypting self-modifying virus which took pains to leave no usable identity would be impossible to recognise using a virus-specific search pattern, but would be unable to avoid re-infection. A virus of this sort might however limit the growth of infected objects by the use of data compression which would delay the timing of its eventual discovery.

Alternatively a virus might adopt a camouflaged infection strategy, with the virus signature selected as a common pattern or attribute found in many legitimate executables. For example, the virus might ensure that the file length after infection was always an even number of bytes, and would refuse to infect any executable that was not an odd number of bytes long. The virus would not infect all files because it would incorrectly recognise its own signature in many of them, but the ones it did infect would be extremely difficult to spot. Any search using the virus signature would result in an unacceptably high number of false positives.

The 1260 virus is, I believe, the first sizeable nail in the coffin of virus-specific software. Search programs already struggle to keep up with the seemingly endless flood of new viruses - but more sophisticated viruses along the lines of the 1260, which will probably follow in the future, will soon push virus-specific recognition software past the limits of practicability.

Note the important distinction between **detection** and **recognition**. All viruses, whether like the 1260, or more sophisticated, will continue to be detectable by **non-specific anti-virus systems** based on checksums, even when they can no longer be recognised by virus-specific searching software.

TUTORIAL

This is the first in a series of short tutorial articles designed to provide background information on IBM PC computer viruses. The series is intended to support the more specific technical analysis which appears elsewhere in the Virus Bulletin. Here, the various gateways to viral infection are explained.

How Does An IBM PC Virus Infect a Computer?

The methods by which a computer virus can infect a PC are well understood and it is worthwhile documenting them, if only to demystify this process.

PC Infection and Media Infection

It is important to distinguish between an infected PC and infected media. The PC becomes infected when virus code is executed and switching the machine off clears the virus from memory. By contrast, most infected media (and this includes hard disks), will carry the virus after the power is turned off.

For instance, should a PC become infected with the Italian virus from a floppy disk, the hard disk will also become infected. If the power is switched off, the virus will disappear from PC memory, but not from the hard disk. When the power is restored and the PC bootstrapped (started) from the hard disk, the machine again becomes infected. The PC will only be virus-free if it is booted from a clean system disk

Viruses Need An Executable Path

A virus becomes dangerous only when it is executed. Virus attack points can be listed because all executable items on the PC are well known.

In addition to executable files such as .COM and .EXE programs, any file containing executable code has the potential to carry and spread a virus. This includes files with interpreted BASIC commands, spreadsheet macros and other applications.

On a PC, the attack points can be listed by analysing the bootstrap procedure when the machine is started, either by switching it on, or by using a 'warm-boot' (pressing the Ctrl, Alt and Del keys simultaneously).

The Bootstrap Procedure

When the computer is switched on, or a warm boot is performed (Ctrl, Alt, Del) the PC first runs the program stored in its ROM (Read Only Memory). The ROM program tests the drives (A: B: C: etc) for the first one which contains a disk. Once a disk is located, the first sector of the disk (Disk Bootstrap Sector) is read into memory. This is a short program. If the disk does not contain a recognisable bootstrap sector, the computer displays the message 'Non-system disk', or similar, and waits for the user to insert a 'system' disk.

On IBM-AT computers, the system also reads various system configuration parameters from the CMOS memory prior to performing this step.

On hard disks (usually drive C:) the disk bootstrap sector program reads in the partition bootstrap sector, which in turn reads in DOS (the operating system) and transfers control to it.

On floppy disks the bootstrap sector program reads the operating system (DOS) from disk into memory and transfers control to it.

DOS is contained in the first two files found in the root directory. These are usually called IO.SYS and MSDOS.SYS (or IBMBIO.COM and IBMDOS.COM) although different names may be used. DOS then accesses file CONFIG.SYS which stores details about the configuration of the system (device drivers, file buffer allocation etc.). Device drivers such as ANSI.SYS are loaded into memory at this stage.

DOS then loads COMMAND.COM and executes it. COMMAND.COM processes the commands entered by the user.

The AUTOEXEC.BAT file, if found, is then executed, thus completing the bootstrapping process. AUTOEXEC.BAT contains a sequence of commands which are executed automatically every time the PC is switched on.

The user then sees the system prompt (C:\> or similar) and the system awaits user commands. Commands issued are either internal DOS commands, or EXE, COM or BAT filenames. The system searches for requested files in all directories and subdirectories specified in the PATH command and executes the first found. Programs can also load executable overlay files (OVL) as and when needed. Overlay files have extensions such as OVL, OV1, OV2 and so on. Applications can, in turn, use macros which are executable code.

Routes of Infection

A computer virus must penetrate and change one or more of the above executable items in order to start iself. Let us consider each in turn in order to assess their vulnerability to attack by current and possible future viruses.

1. Programs held in ROM. There is no possibility of attack since ROM cannot be modified in any way once it is programmed by the manufacturer. CMOS memory contains no executable code and cannot contain a virus.

2. Disk Boot Sector/Partition Boot Sector on hard disks. The New Zealand virus attacks the Disk Boot Sector. Several others attack the Partition Boot Sector (Italian and Mistake).

3. Disk Boot Sector on floppy disks. Several viruses such as Brain and Italian attack the Disk Boot Sector on floppy disks.

4. DOS files IO.SYS/MSDOS.SYS. A possible attack point, although no known viruses infect either file.

5. CONFIG.SYS. This is a text file and cannot contain a virus. However, it could easily load and execute a device driver which contained a virus.

6. COMMAND.COM. The Lehigh virus attacks this file specifically.

7. AUTOEXEC.BAT. A possible attack point. This file has been manipulated by Trojan horses such as the AIDS Information Diskette.

8. Applications, .EXE, and .COM files. Several viruses attack these files. Overlay files have not so far been attacked mainly because there precise structure is not standardised. However, they could be targeted by a virus.

9. Batch (.BAT) files. Experimental viruses have attacked these files.

10. Macro files. Experimental or 'lab' viruses attacking Lotus 123 spreadsheets have been demonstrated.

To keep the system free from viruses the user must ensure that the code executed during this bootstrap sequence is virus-free. Some viruses use a number of attack points and also attempt to camouflage their exact method of infection.

Virus Type According To The Point of Attack

Computer viruses affecting the IBM PC can be divided into two categories according to the executable item which they infect: Bootstrap sector viruses and Parasitic viruses.

Bootstrap Sector Viruses

This type of virus modifies the contents of either the Disk Bootstrap Sector or the Partition Bootstrap Sector, depending on the type of virus and type of disk. The virus usually replaces the contents with its own version, the original version is transferred to another area on disk. This means that once the machine is started, the virus' 'bootstrap sector' is executed first. This normally loads the rest of the virus code into memory followed by the execution of the genuine bootstrap sector. Boostrap sector viruses normally remain memory-resident until the computer is switched off. These viruses are thus able to monitor and interfere with the action of the operating system from the very moment they are loaded into memory.

The method of infection comprises:

1. Replacing the genuine bootstrap sector with an infected version which enable the virus to gain access. The Swap virus uses a different method and overwrites the genuine bootstrap sector with a program which loads the virus.

2. Storing the genuine bootstrap sector in a previously unused sector of the disk.

3. Storing the bulk of the virus code on a number of unused sectors of the disk.

> Examples: Brain (floppy disk bootstrap sector)
> Italian (floppy disk bootstrap sector and hard disk partition bootstrap sector)
> New Zealand (floppy boostrap sector and hard disk bootstrap sector)

Parasitic Viruses

Parasitic viruses currently infect .COM and .EXE files. They usually append themselves at the beginning or end of a file, leaving the program intact. The initial JMP instruction in the infected program is modified, but program functionality is unimpaired. Infected files usually grow because the virus is attached to them. For example, a short program of 640 bytes will appear to contain 2341 bytes if it is infected by the Cascade variant 1701.

Some parasitic viruses overwrite the first few hundred bytes of an affected program rendering it unusable.

When an infected program is run, the virus code is executed first. The virus then restores control to the original program which executes normally. The extra time taken for this process to occur is too brief to be noticed by the user.

Some parasitic viruses, like Cascade, spread when another program is loaded and executed. Such a virus, being memory resident, first inspects the program for infection already in place. If it is not already infected, the virus will append itself to it. If it is already infected, further infection is not necessary (although a Jerusalem variant does reinfect ad infinitum because it cannot recognise its own signature).

Other parasitic viruses do not install themselves in memory. Instead, they spread by locating the first uninfected program on disk and infect it. An example is the Vienna virus.

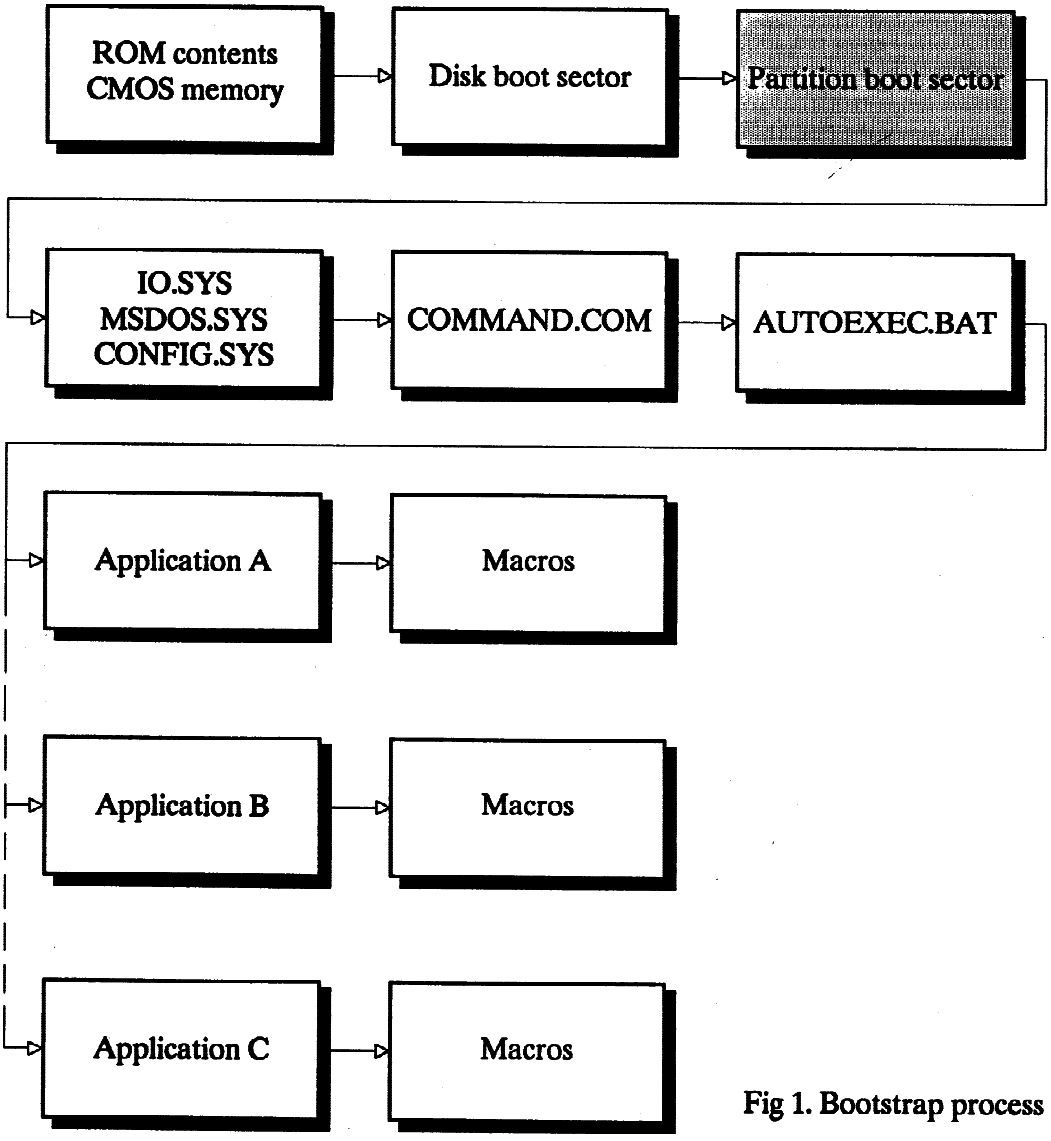

Fig 1. Bootstrap process

COUNTERMEASURES

Fridrik Skulason

PC Anti-Virus Tools - How Do They Work?

As the number of known PC viruses continues to rise, so does the number of anti-virus tools. Anti-virus programs may be very specific, designed to fight a single virus, but in other cases the authors aim for a comprehensive countermeasures package. Some of the anti-virus programs are commercial products, but others, including some of the best known, are distributed as shareware. A few of them offer a high degree of protection, others are useless. The purpose of this article is to provide some information on the inner workings of the current anti-virus tools in order to help you to select a suitable program.

Anti-virus software can be divided into several categories, each has certain advantages, but also some disadvantages. **It should be emphasised that the techniques described below may be of little or no use if a virus is already active when preventative anti-virus software is installed.**

Checksum Programs

A checksum program utilises checksums to detect changes to executable code. Here 'executable code' may mean ordinary applications programs, the operating system files, the boot sector or even the partition boot record.

Several different algorithms may be used to compute a checksum for any given piece of code. While it might be easy to write a virus which could evade a single checksum program (by ensuring that the modified file produces the same checksum as the original one), it is not possible for a virus to bypass every checksum in this way. A single, strong algorithm used in a checksummimg program can offer a high degree of protection against evasive viruses. (*See Checksumming Methods Used to Detect Virus Attacks, VB, September 1989*).

The main problem with checksumming software is that it can only provide a 'post-attack' warning. They are unable to prevent infection, but will alert the user after an infection has occurred. Moreover, a checksumming program cannot identify a particular virus but can only warn of unauthorised modifications to executables within the machine requiring further investigation.

Another problem is that before the checksum is originally computed, the user must ensure that the executables being checksummed are free from infection.*It is therefore recommended that a scanning program which searches for known viruses is used prior to the installation of a checksumming program (see below).*

The programs must also store the checksum in a safe place, where no virus can tamper with them.

The greatest benefit offered by checksum programs is that they are equally effective against **all** viruses (present and future) and no updates are needed when new viruses appear. They can also detect modifications caused by other malicious programs such as Trojans.

Checksum programs are only fully effective if the computer is booted from a clean disk before they are run. If a virus is active in memory when the checksum program is run, it could deceive the checksum program, by making it appear that infected programs had not been modified. Several of the latest viruses attempt to do this. (*VB, March 1990, page 10*).

Inoculation Programs

An inoculation program works by making the victim appear as if it has already been infected. Many viruses use a special signature string to mark a file or boot sector as infected. As an example, the Jerusalem virus places the string 'sURIV' at the very end of the infected file. Any program containing this string at the end will be immune from infection since it will appear (to the virus) to be infected already.

An inoculated file can be used safely in a 'high risk' environment, without any risk of it becoming infected by the particular virus it has been inoculated against. Inoculated disks can be used to transfer data safely from a computer where a boot sector virus is active, with no risk of them becoming infected.

Inoculation programs have several drawbacks.**It is impossible to inoculate against all viruses.** A number of viruses mark infected files by a sequence of bytes at

the end of the file. Therefore, it is only possible to inoculate against one such virus at a time. In other cases, inoculation is impossible because the virus does not check for a previous infection and continues to reinfect files. The 405 virus is one such example.

It is important to point out that a generic inoculation program which would operate against all viruses is impossible. Inoculation programs can only be effective against a subset of known viruses. **Inoculation offers no protection against unknown threats and is so specific that it is inappropriate as a means of general virus defence.**

Virus Specific Monitoring Programs

This type of program checks every program being executed for infection by known viruses. As this checking is done before the programs are executed, the infecting virus will not have a chance to activate. While programs of this type offer complete protection against known viruses, they are totally ineffective against unknown viruses or other malicious programs. This makes constant updating necessary.

Self-Checking Programs

It is possible to add a 'self-testing' module to an existing program. This module is executed first whenever the program is run, in order to verify that the program had not been modified since the module was added. Programs of this type may verify the file length and creation date in addition to using checksums. This method will catch any infection, provided that the virus is not able to make the program appear to be unmodified.

There is one problem with this method. When an infected program is run, the virus will be activated before the self checking module, so the computer will be under the full control of the virus.

Write Protection Programs

A number of anti-virus packages provide tools to make disks write-protected. This usually involves the interception of INT 13H followed by a check for any 'Write' or 'Format' request. Unfortunately, many of the available programs do not intercept INT 40H, so they provide no protection against viruses using this method to write to diskettes. A virus could also attempt to write to a disk by jumping directly to ROM and only a few of the currently existing programs prevent this.

Ultimately, it is only possible to provide secure write-protection by using hardware specifically designed for the purpose.

General Monitoring Programs

Originally, programs of this group were designed as anti-Trojan programs, but they are equally effective against viruses. Monitoring programs intercept various interrupts such as INT 21H and INT 13H. Every call is checked and whenever a suspicious one is detected the monitoring program takes action. It may simply cause the requested operation to fail but usually a pop-up window appears on screen. The attempted operation is described and the user is asked whether to allow it or not. A typical "suspicious operation" might be an attempt to write to the boot sector. Normally only the FORMAT command should be permitted to do this - anything else might indicate virus activity.

The main problem with monitoring programs is that the latest generation of viruses are designed to bypass them. Some of them (eg Number of the Beast, December 24th) locate the original INT 13H or INT 21H entry points and jump to that address instead of issuing an INT instruction. Monitoring programs also require a high degree of end-user knowledge so that correct actions are taken when on-screen warnings appear. (*A number of organisations which have evaluated software of this type have found that it is quickly disabled by end-users due to the number of warning messages issued. These result from the software misinterpreting legitimate program activity. This reinforces the lesson that a security product will only be useful if end-users are prepared to comply with its continued operation. Ed.*)

"Any virus can be detected and stopped, even if this means that virus tools need to be constantly updated as new viruses appear"

In theory, monitoring programs could provide a defence against all viruses, not just currently known ones. However, the appearance of evasive viruses, means that the authors of this software must predict future evasion and concealment techniques.

Virus Scanning Programs

Virus scanning programs use identification patterns (hexadecimal dumps) to identify infected programs. Some of them search for the patterns at fixed offsets from the virus' entry point, others scan a large area of the file. The first method is faster, demands absolute accuracy and is also more likely to miss new virus variants.

Some of the programs use long identification strings, 16 bytes or more, while others use short sequences of 5 or 6 bytes. Using long strings decreases the likelihood of false alarms but reduces the probability of detecting new variants of known viruses. The likelihood of detection is increased if two or more different hexadecimal patterns from the same virus are included in a scanning utility. For security purposes it is best if the hexadecimal pattern used is concealed in order that a virus writer cannot modify the exact area of the virus being searched for thus rendering the search utility useless against the modified virus.

As scanning programs rely on identification patterns, they are not effective against new viruses such as 1260, and they may fail to detect modified versions of older viruses. Future programs of this type may search for the virus' signature (as opposed to its identification pattern). The signature encompasses any modification or characteristic that is specific to a particular virus or group of viruses. (*See page 10*).

Code Analysis Programs

The idea behind this class of programs is to determine whether a program contains destructive code - a routine to format the hard disk, for example. This method has not yet been shown to be effective - the analysis programs produce a string of false alarms and also fail to detect many viruses.

Virus Removal Programs

When a virus has been detected, it must be removed. Disinfection programs are sometimes combined with detection programs such as the *ViruSafe* anti-virus program evaluated in this month's edition. However, they may also be independent. Very few disinfection programs are able to remove all known viruses - most of them are limited to 20 or 30 of them. Some of the programs which claim to remove known viruses do not restore the original program in all cases - many of them simply overwrite the virus, leaving the program unusable.

Successful disinfection is not always possible. One example is the 405 virus, which overwrites its host program, without storing the overwritten code. The Jerusalem and Taiwan viruses may infect a file incorrectly, making disinfection impossible. Finally, some viruses (Vienna, for example) sometimes destroy the programs which they infect.

Which Programs Are Best?

It is not possible to pick a single program or group of programs and say that they are the most suitable for all purposes under all circumstances. In some cases a good checksum program is all that is required. Other users may prefer monitoring programs. For diagnostic purposes, a virus scanning program is indispensable - particularly to identify infected diskettes. Some users may be more concerned about Trojans than viruses, in which case a general interrupt monitoring program might be the best choice. Unfortunately, there is no easy answer to this problem.

Finally...

It is important to bear in mind that there is no such thing as a perfect anti-virus program. A virus author could, in theory, write a virus to bypass any single anti-virus program. Each and every anti-virus program will have some weaknesses which could be exploited by a virus. *Obviously, well designed security features incorporated into an anti-virus program will reduce the likelihood of a virus successfully evading detection.*

We should also remember that there are no perfect viruses. **Any virus can be detected and stopped even if this means that anti-virus tools need to be constantly updated as new viruses appear.**

VIRUS DISSECTION

Richard Jacobs

Typo: The Touch-Typist's Nightmare

The Typo virus is believed to have originated in Israel late last year. It is a parasitic virus, infecting COM files and has an infective length of 867 bytes. The virus causes characters to be mistyped when typing faster than about nine characters per second.

The viral code is added to the end of COM files, with a jump to this code placed at the start of the file. After the virus has been executed the file runs exactly as usual. The initial JMP instruction to the virus is written over the first 3 bytes of the file. These bytes are stored within the virus and are restored before control is returned to the program, so that normal operation is assured.

When an infected COM file is executed, the virus calls a routine to set up the mistyping side effect. This routine makes a copy of the run counter, which is in the range 04H to 5BH and is decremented every time the virus runs. Two copies are then made of the low word of the main system timer. These are used, along with the run counter to decide when to mistype characters. INT 21H (Function Request), INT 16H (Keyboard BIOS) and INT 20H (Program Terminate) are then redirected while their original vectors are saved.

INT 20H is set to perform an INT 21H with function 4CH (Terminate with Return Code). INT 21H performs as usual, unless function 00H (Terminate) or function 4CH are called. Unless the COM file finishes with a JMP 0000H instruction it must call one of these functions to terminate. When this happens, the mistyping routine is copied into memory, INT 20H and INT 21H are restored to normal operation and INT 16H is set to the start of the mistyping routine. The program then terminates using DOS function 31H (Terminate Stay Resident), leaving the mistyping routine in memory. If the mistyping routine is already installed in memory it is not re-installed.

All subsequent INT 16H calls are directed to this memory resident routine. All INT 16H calls other than "read next key" are carried out as normal. When a request is made to read in a key, the original INT 16H call is used. The key is either returned as normal, or the next key to the right on the keyboard is returned. The decision on whether or not to return the correct key is based on a combination of the speed of typing and on the run number mentioned earlier. This combination results in an apparently random occurence of incorrect keys returned. This side effect is **not** case sensitive and all printable characters (including numbers, punctuation marks and other symbols) are affected.

Typing fewer than 9 characters per second will never return wrong keys, and above this speed two consecutive keys can never both be incorrect. *This side effect will impede touch-typists and secretaries with a typing speed of approximately 70 words per minute or above. Typo is similar to a number of other viruses in that it is designed to cause irritation and loss of confidence in the user.*

The virus copies the stored initial bytes of the file into a temporary location, before checking the date. If it is an odd day of the month then all uninfected COM files in the current directory are infected. Otherwise the initial bytes of the file, along with INT 20H and INT 21H are restored and the file executes normally. INT 16H is left redirected to the TSR section of the virus.

If the day of the month is odd all COM files in the current directory are examined in turn. First of all the attributes of the COM file are set to Read/Write. The file is then opened and the first three bytes are read into a safe location within the virus. These bytes are used to check whether or not the file is already infected. They will be overwritten if the file has not been infected and so must be stored elsewhere. The first byte is checked to see if it is a JMP instruction. If it is not, then the file is not already infected and the infection routine is run. Otherwise the next two bytes are examined. These bytes are used to compute the address of the start of the virus code, if the file is infected. The file pointer is moved to a location given by this address and two pairs of test bytes are compared; if they do not match the infection routine is run. The next COM file is then checked and infected. After all COM files in the current directory have been checked, the first three bytes of the COM file being executed are restored and the program runs normally.

The infection routine of the virus is straightforward. It finds the end of the file to be infected and copies itself to this position. It then overwrites the first three bytes of the file with a jump instruction to this viral code.

Finally the date and time of the previous write to the file are restored and the file attributes are returned to their original settings, to hide the fact that the virus has been added.

```
Typo: 5351 521E 0656 0E1F E800 005E 83EE 24FF
Offset 01D, 867 Bytes
```

PRODUCT EVALUATION

Dr. Keith Jackson

ViruSafe: Virus Protection System

ViruSafe is a software package that claims to: *recognise all viruses that infect the boot sector and partition table of disks; recognise all viruses that infect program files; remove most known viruses; and identify viruses which are trying to become memory resident.* In short, ViruSafe utilises all of the three categories of programs which are commonly used against viruses. These three types involve scanning for virus signatures, looking for changes in checksums calculated across programs, and watching for virus activity using memory resident software.

The copy of ViruSafe used for this review was provided on a 3.5 inch floppy disk. I had first attempted to use a 5.25 inch floppy disk, but this proved impossible due to the copy protection scheme in use.

Note: ViruSafe requires at least version 3.00 of MS-DOS (see technical details below). Anyone who has retained MS-DOS v2.11, perhaps for its compactness, cannot use ViruSafe without upgrading to a later version of MS-DOS.

As a routine precaution I check any incoming disk with two scanning programs *(Sweep* from Sophos Ltd., and *Scan* from McAfee Associates) to find out if the disk contains any viruses. I did this with the ViruSafe disk, and no viruses were reported. As ViruSafe itself contains a scanning program, the signature patterns used by this scanning program to identify viruses must be concealed, probably using encryption.

At only 42 pages long, on paper smaller than A5, the ViruSafe manual is not the longest ever written. However, it is easy to comprehend. It contains a section for beginners entitled *"What is a computer virus?"*, through installation instructions, to a detailed explanation of how ViruSafe operates. The manual explains that ViruSafe cannot be used simultaneously on more than one computer. However, it studiously avoids using the words 'copy-protected', which would make matters a lot clearer. My 3.5 inch ViruSafe floppy disk came with a covering letter explaining that ViruSafe was copy-protected, but I do not know if this is standard issue to all purchasers.

The back page of the manual contains the epic phrase: *"The panacea for the computer virus epidemic has been found, and it is called ViruSafe*". The *Oxford Dictionary* defines *panacea* as "*Universal remedy; nostrum*". Interestingly, the same dictionary defines nostrum as *"quack remedy, patent medicine or pet scheme*".

ViruSafe contains a README file (my version was dated 10th Nov 89), which explains that it can handle TSR viruses, partition table viruses, and DOS boot sector viruses (i.e. all of the types of virus that are currently known to affect the MS-DOS operating system). The utility called UNVIRUS provides a list of 79 viruses currently known to ViruSafe, along with a description of each virus, its size, and type (program virus or boot virus). From the 79 viruses, ViruSafe claims to be able to remove 55.

Menu Options

A utility called VSMENU is provided to facilitate simple operation of ViruSafe. This is excellent, and has four main menu options, which correspond to the four main programs within the ViruSafe package. These menu options offer to check/remove viruses, check program integrity, immunise memory and display a list of known viruses. The function of VSMENU is to tie together these disparate sections of ViruSafe. For instance, the program UNVIRUS can be executed directly (*see above*), and in this mode of operation will immediately show a list of all viruses known to ViruSafe on the screen. This option can similarly be invoked through the appropriate menu choice from VSMENU. Direct execution is aimed at skilled users, menu operation is provided for those just learning to use ViruSafe. First-time users will find menu operation a fairly painless introduction, and they can then move on to more advanced usage when they feel capable.

I found VSMENU simpler to use than the documentation. I have seen a previous version of ViruSafe which did not contain the VSMENU program, everything seemed far more confusing, and it was difficult to see how and why everything worked. VSMENU makes operation far easier.

Does ViruSafe Detect Viruses Correctly?

I tested ViruSafe against all the viruses listed in the technical notes section below. It encountered slight notation problems about which version of Datacrime and Jerusalem had been removed, but given the number of possible permutations, this is hardly surprising. The only mistake made by ViruSafe was to identify South African virus 1 and 2, while erroneously stating that they were Icelandic 1 and Icelandic 2 respectively. When ViruSafe has found a virus while searching a disk, it makes a whooping noise (reminiscent of a police siren), when one or more viruses are found. It is very difficult to ignore this warning.

Can ViruSafe Also Remove All Of These Viruses?

Of the viruses listed below, only 405 cannot be removed, and the ViruSafe documentation makes it clear that this is not one of the viruses which are capable of being removed (the virus destroys part of the original program). A clear on-screen

message warns that 405 has not been removed, and advises that the specified file should be immediately deleted.

Given the nomenclature problems which plague the investigation of viruses, we should not be surprised at the minor problems described above in detecting and removing viruses (*see "Nomenclature for Malicious Programs", Virus Bulletin, March 1990*).

When ViruSafe tests for viruses in memory, while executing from a floppy disk, it insists that the disk is not write-protected. I don't know what causes this, but no matter what the reason, it is bad practice. **The only way to be absolutely sure that a disk is not infected by a virus is to leave the write protect tab in place.** Investigating viruses with floppy disks that are not write protected is fool-hardy.

I invoked the option to produce a "List of marked files" from the program integrity menu. This drew a pretty border, wrote a suitable heading, and then did absolutely nothing. After some investigation it transpired that this was because I had not correctly marked any files, but nothing warned me of this. I was just shown a blank screen, which is not very helpful. After marking some files, all was well. It is probably best to check program integrity whenever DOS is booted, which requires direct execution of the checksumming program used to verify file integrity. Therefore the menus provided by VSMENU are not as useful for checking integrity as they are when searching for viruses.

Copy-Protection

I did not install ViruSafe on my hard disk. When I had tried out an earlier version of ViruSafe (on a 5.25 inch floppy disk), I tried to install it without being told that it was copy protected. ViruSafe was being installed from a 5.25 inch drive (either drive B or drive C on my computer), but the installation software kept on insisting that it had to read drive A (a 3.5 inch drive). I tried to copy ViruSafe to a 3.5 inch disk, but installation then failed because the copy protection mechanism intervened. I eventually gave up, and had to obtain a 3.5 inch version to write this review. However, I found much later that ViruSafe had made a hidden directory on my hard disk, with space characters in the name to make it hard to erase, which contained two hidden files. *The Norton Utilities* was the only program which I possess capable of erasing this directory and its contents. Standard MS-DOS utilities (such as delete, or remove directory) won't operate with anything that has spaces embedded in its name. Nothing in the manual states that this will happen. Most users would not even notice the presence of these hidden files, which are not revealed by using the normal MS-DOS utility (DIR).

This was a consequence of the copy-protection employed by ViruSafe. I never give a program that has caused such problems a second chance to foul up my hard disk, so all the above tests were run from the floppy disk copy of ViruSafe.

Within a matter of months, ViruSafe has gone from having knowledge of only 6 viruses [April 1st, Friday the 13th, Brain, Marijuana (New Zealand), Ping-Pong (Italian), and Alabama] up to a total of 79 viruses, coupled with a reasonable description of each virus. The authors of the program have certainly done a good job of keeping up with recent developments.

I think that ViruSafe is well thought out, has an excellent front-end menu system that makes initial operation quite painless, and is capable of detecting and removing a large range of viruses. I would recommend its use but for the fact that it is copy-protected. I've written at length about copy-protection in the past, and firmly advise people to avoid copy-protected software at all costs. In this case it is particularly galling, as I really do like the way that ViruSafe works. **Copy-protection prevents taking proper backups which are the first (and most vital) line of defence against viruses.** Life is too short to put up with the inanities that can be introduced by copy-protection schemes.

Technical Details

Product: ViruSafe

Developer: Eliashim MicroComputers Ltd., P.O.Box 8691, Haifa 31-086, Israel, Tel: +972 (4) 523601, Fax +972 (4) 528613.

Vendor: Eliashim Microcomputers Inc., 520, W.Highway 436 Suite 1180-30, Altamonte Springs, FL 32714, U.S.A., Tel: +407 (682) 1587, Fax +407 (869) 1409.

Availability: IBM PC/XT/AT, PS/2, or compatible with 256K of memory running MS-DOS version 3.00 or above.

Version evaluated: 3.00

Serial number: VS303187, 3.5 inch copy protected disk.

Price: US$60.00

Hardware used: ITT XTRA (a PC compatible) with a 4.77MHz 8088 processor, one 3.5 inch (720K) drive, two 5.25 inch (360K) drives, and a 30 Mbyte Western Digital Hardcard, running under MS-DOS v3.30.

Viruses used for testing purposes: for a complete explanation of the nomenclature used, please refer to the list of PC viruses contained in *Virus Bulletin*:

Brain, Italian, Vienna, Jerusalem, 1701, 1704, Datacrime 1, Vienna 1, Cascade 1, 2, Datacrime II, 405, Fu Manchu, Jerusalem 1, 2, Traceback, Suriv 1.01, Suriv 2.01, Suriv 3.00, South African 1, 2.

END-NOTES & NEWS

UK based *Microcom Software Division* have acquired US software company HJC, developer of the Virex anti-virus package for the Macintosh (*see Product Review, VB, December 1989*). Virex has been updated to version 2.51 and can recognise the two recent Trojan horses (Mosaic and FontFinder) which *VB* reported last month. The package will also detect the latest nVIR-B virus clone 'FUCK', also known as nVIR-F. The latter name has caused widespread confusion because it implies the existance of nVIR-C, D and E which, so far, have not appeared.

A CERT advisory of March 19 reported several **persistent intruder attempts on the US DARPA Internet.** According to a report in the *New York Times* the intruder is using a host emulation program, or similar, to trawl passwords. CERT deny having located a program on the network but admit that several systems were broken into on both March 15 and 16. They directed system administrators to disable the UNIX *Trivial File Transfer Protocol* which can be used to steal password files and to watch for vendor supplied default passwords, holes in *sendmail* (Berkely BSD 5.61 patches all known holes used by the intruder) and *fingerd* (these holes were exploited by the Morris Internet worm). Recent VMS system attacks exploited system default passwords that had not been changed since installation.

VALERT-L posted a warning on March 29 about a **newly discovered IBM PC virus** (provisionally called XA-1) which activates on April 1st. On that date, the virus drops sabotage code into the partition table of the hard disk and into the boot sector of floppy disks. Virus hangs the system on next boot-up. Virus only infects .COM files. Infective length is 1539 bytes. Second payload is active between December 21 and end of year. Draws a Christmas tree on screen with message "Und er lebt doch noch: Der Tannenbaum! Frohe Weihnachten..." (Translation: And still it is alive: The Christmas Tree! Merry Christmas...). Possible reference to CHRISTMA EXEC (see page 9). Virus signature which is found at the beginning of .COM files is:

```
EB 07 56 0A 03 B9 00 (This is the virus signature not a search pattern)
```

Computer Virus Conference, Amsterdam 20 April; London 23 April, Geneva 24 April. Dr. Fred Cohen presents the results of seven years of research. Details from IBC Technical Services, UK. Tel O1 236 4080.

Second Annual Seminar on PC Security (7 June) **and Computer Viruses** (8 June). Speakers include John McAfee (CVIA) and Fred Cohen. IBC Technical Services, UK. Tel 01 236 4080.

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including delivery:

US$ for USA (first class airmail) $350, Rest of the World (first class airmail) £195

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Abingdon, OX14 3YS, England

Tel (0235) 555139, International Tel (+44) 235 555139
Fax (0235) 559935, International Fax (+44) 235 559935

US subscriptions only:

June Jordan, Virus Bulletin, 590 Danbury Road, Ridgefield, CT 06877, USA
Tel 203 431 8720, Fax 203 431 8165