May 1990 ISSN 0956-9979

VIRUS BULLETIN

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

Editor: **Edward Wilding**

Technical Editor: **Fridrik Skulason**, University of Iceland

CONTENTS

EDITORIAL

TUTORIAL

CRISIS MANAGEMENT

LETTERS

VIRUS DISSECTIONS

TOOLS & TECHNIQUES

PRODUCT EVALUATION

BOOK REVIEW

EDITORIAL

US software house *FoundationWare* recently publicised the findings of a report on computer viruses compiled by the company's president Dr. Peter Tippett. The report '*The Kinetic Replication Of Computer Viruses'* concludes that by 1992 computer viruses will affect 8 million personal computers globally and that virus infections will reach 'epidemic proportions' by 1995. According to Michael Reimer, the company's executive vice-president, the potential costs in terms of data loss, downtime and recovery could exceed 10 billion dollars over the next five years. Dr. Tippett says that computer virus hype has been highly counter-productive. "It has turned many people off, diverting the public, legislators and corporate management from understanding how serious the problem truly is" he says. He concludes that the situation will only be diverted if companies implement comprehensive system management policies in order to protect PCs from replicating software.

FoundationWare produces a range of protection and systems management packages, notably the *Certus* program* which 'prevents viruses, bombs and malicious software' and claims to be "the most comprehensive software protection and microcomputer management system available today". *Certus* was designed by Dr. Tippett in association with Mark Hosmer, and we would expect *FoundationWare's* official statements on the computer virus issue to accord with and support their product range.

However, there are early indications that the findings of the report, far from being an example of virus industry propaganda, may accurately foretell the state of things to come. Currently, the most striking example of the 'kinetic replication of computer viruses' can be found in the Peoples' Republic of China.

The BBC's *World Service* recently broadcast a report from China stating that computer viruses had indeed reached 'epidemic proportions' and that one in ten personal computers were plagued by viruses. The reporter specifically mentioned the ubiquitous Jerusalem virus and said that the *Ministry of Public Security* was engaged in a concerted education effort to try and avoid further infection on the republic's 300,000 machines. Deputy chief of the ministry's computer security department, Yang Zhihui, is understood to be in charge of a concerted research initiative to develop vaccines and disinfection programs. According to the *China Daily* newspaper, a survey taken in August 1989 established that a tenth of the country's computers were afflicted by six different viruses. Chinese computer experts blame the epidemic on imported software and software piracy which is rife in the Far East, although rumours abound that the viruses were introduced to the republic intentionally by hostile intelligence services.

The global spread of the Jerusalem virus (this edition of *VB* reports two more variants of this virus, with a third awaiting examination) since its discovery in January 1988, and events in China, seem to confirm Tippett's 'kinetic replication' theory and serve as a warning to the computer industry and users alike.

The Bulgarian Fifty

The Bulgarian series of viruses which *VB* reported last month are currently undergoing painstaking disassembly and analysis. This work has to be undertaken, because it is inevitable that more samples from this series will escape and cause 'real world' infections. The '666' virus is analysed on page 13 of this edition.

The writers of these viruses seem to be engaged in some puerile battle to prove their programming prowess. Their efforts are neither admired nor wanted. Prosecution under Bulgarian law is not possible and the only deterrent to this activity seems to be disciplinary measures undertaken by staff at the university or institute where it is occurring. This is another example which highlights the shortcomings of a country's legal system to combat computer vandals and criminals.

Popp and Morris

Proceedings for the extradition of Dr. Joseph Popp in connection with the AIDS Information Diskette extortion bid have been delayed, meanwhile sentencing of Robert T Morris (*see VB, April 1990, pp 9*) takes place on May 4th.

Jerusalem, Again!

In January, the *United States Census Bureau* accidentally distributed a floppy disk infected with the Jerusalem - B virus. The disks were distributed to 772 libraries. The disks were sent out with the County and City Data Book CD-ROM by the *Government Printing Office.*

Disk Identification Label:

```
Bureau of the Census
Elec. County _& City Data Bk., 1988
USA, Inc., 1101 King St.
Suite 601 Alexandria, Va. 22314
tel:(703)-979-9699
```

The disk is listed as C 3.134/2:C 83/2/988/floppy-2.

It is understood that all recipients have been notified and affected disks recalled or destroyed.

FAX Numbers

In the event of an emergency similar to the AIDS disk incident of last December, *Virus Bulletin* intends to disseminate preliminary information as quickly as possible. We intend to do this by facsimile. Subscribers wishing to receive immediate information should ensure that we have a contact fax number available on our database. Overseas subscribers should include both country and area codes.

* *Certus* will be the subject of a forthcoming *VB* technical evaluation. The report "*The Kinetic Replication Of Computer Viruses*" is available from FoundationWare Inc.,13110 Shaker Square, Cleveland, Ohio 44120, USA.

TUTORIAL

Virus Infected Media and Routes of Infiltration

In last month's VB, the internal routes of virus infection within a personal computer were outlined. This article documents the external sources of infection, i.e. media which can become infected by computer viruses and methods by which they are transmitted.

Infective Media

Any media which can be used to transmit executable code can be infected by a **parasitic virus**. Similarly, media which can be used to bootstrap a PC can also carry **bootstrap sector viruses**.

A PC becomes infected with a parasitic virus when the user executes (runs) an infected program. The system is infected by a bootstrap sector virus when bootstrapped from an infected medium.

Floppy Disks

Floppy disks (also called diskettes) are the most common medium for transferring data or programs between PCs. Floppy disks can become infected by parasitic viruses which can append to any executable code on the floppy, or by bootstrap sector viruses which hide in the bootstrap sector of the disk. **Bootstrap sector viruses can reside on data disks and infect from them if such disks are accidentally used to boot a machine.** Both the Brain virus and the Italian virus can infect the system and successfully transfer control to MS-DOS in this way.

Executing an infected program need not be a conscious action on the part of the user. It is quite easy to accidentally bootstrap from a non-system disk. This will cause a screen prompt saying:

```
Non-system disk or disk error, insert system disk
and press any key to continue
```

Attempting to boot from a non-system disk is a common occurrence and an easy mistake to make. It can happen if, for instance, a floppy disk is left in drive A: overnight and the machine is then switched on the next morning. A PC can also become infected by a virus if a short power-cut occurs while the machine is unattended with an infected floppy disk installed in the drive. It is likely that a user will not notice that the machine has rebooted during his absence.

It is possible to prevent bootstrapping from floppy disks in AT machines and compatibles by altering CMOS memory to 'thinking' that the machine has no floppy disk drive(s). The machine is forced to bootstrap from the hard disk. Floppy disk drives can also be disabled. Various access control products disable use of the floppy disk drives. Such actions intefere with installing software, taking backups and data transfer and do not appear to be a viable solution to the virus threat.

The commonest cause of infection is the unsuspecting use of virus infected floppy disks - the running of programs on infected machines, the use of infected system disks, and the copying and transfer of infected programs between machines.

Removable Hard Disks

Removable hard disks can be stored in a secure location away from the PC. They are often advocated for isolating sensitive data. They offer no inherent protection against computer viruses because hard disks can become infected, removable or otherwise. **Removable hard disks can be infected by both parasitic and bootstrap sector viruses.**

Magnetic Cartridges

Magnetic cartridges are normally used for storing PC backups. The machine cannot be booted from these cartridges (so there is no danger from bootstrap sector viruses), but **parasitic viruses can be transmitted via them**.

Other Storage Media

Bernoulli drives, optical disks and 1/2 inch magnetic tapes are examples of storage media used with personal computers. A simple rule is:

Any medium which can be used to bootstrap a computer is capable of transmitting a bootstrap virus, as well as a parasitic virus.

If the medium cannot be used to bootstrap the PC, it can only carry a parasitic virus.

Modems

Modem (MOdulator-DEModulator) offer PC users the means to communicate with each other, often using bulletin board systems (BBS) or electronic mail (email) using a telephone network. Modems can be used to upload and download executable images (programs) and can thus carry parasitic viruses. The Tenbyte virus (*VB, April 1990*) was an example of virus transmission via modem. The current generation of bootstrap sector viruses do not lend themselves easily to modem transmission (although such transmission is standard practice amongst the research community). **The major danger to the modem user is the unsuspected downloading of a program infected by a parasitic virus.**

Pirated Software

Copying software is illegal in most countries. Copying is a very simple and inexpensive operation and virtually free from the risk of prosecution. Needless to say, quality control (particularly guarantees that copied programs are ‘virus-free’) is not high on the list of priorities amongst software pirates. **Games are prime targets for copying and they tend to move between the computer community swiftly, thus increasing the chance of infection by parasitic viruses.** In the United Kingdom, the Jerusalem Virus was spread by (among other programs) a pirated version of the game program *‘Leisure Suited Larry In the Land Of The Lounge Lizards’.*

Bulletin Boards

Bulletin boards (BBS) often provide the means to download and upload programs which are classified as ‘public domain’ (free) or ‘shareware’ (copy at no charge, but pay if used). Reputable boards are run under the close supervision of the SYStem OPerators (SYSOPs) who regularly screen program files for known viruses and maintain logs and audits of activity. Banning the downloading of BBS software decreases the dangers of viral infection. Conversely, it may force users to expend time and money searching for equivalent utilities and will deny them access to many useful programs.

Analysing all programs for viral code (either known or new) is a near impossible task and it is difficult to guarantee that available bulletin board software is virus-free. BBS software is usually distributed using compression (such as ARC or ZIP) which provides some defence against viral infection. However compressed software can be unpacked, accidentally or intentionally infected, repacked and uploaded.

Trojan horses such as ARC 513 (a ‘compression’ utility which actually compresses any file to 0 bytes) have also been uploaded to bulletin boards.

Ultimately, use of BBS software must depend on how much the user trusts the individual System Operator. **Under no circumstances should software be downloaded and run from an unfamiliar BBS until further enquiries have been made.**

Demo Disks and Magazine Software

In recent years demo disks have become an increasingly important marketing tool, as have free disks offered as a premium with magazines and journals. Computer viruses, and other malicious programs, now threaten this burgeoning industry. There is no recognised association to oversee, or enforce standards over the production and distribution of these disks. The days when unsolicited software was casually installed and run have gone forever. The AIDS Information Diskette (*VB, January 1990*) served as a salutory lesson to many organisations about the need for software control and inspection.

The use of virus-specific scanning software can establish whether such software is infected by known viruses and checksumming programs can determine whether it modifies system attributes. Ultimately, however, businesses will simply not have the time to check unsolicited software. **Users wishing to install such software must first seek approval from authorised technical staff. Unsolicited disks sent by post should be reported immediately to the data processing manager.** A free DOS utilities disk (called ‘Ten of the Best’) sent out with a popular UK computing magazine in March was infected by a new variant of the New Zealand virus.

Shareware

Shareware was originally conceived in the United States. Shareware carries a copyright notice, but users are encouraged to copy it and pass it on to friends and associates. If the software is used, the user is under moral obligation to send a small fee to the author. In this way, potential users can take the software for a ‘test-drive’. Good software survives because it is genuinely useful; poorly devised, or unwanted programs meet their inevitable demise. Most shareware authors despatch the latest version number, once payment has been received. However, users first test the original (or previous) version obtained from a friend of a friend. **The software, or copy of it, may have been installed and executed on a large number of PCs before it arrives for your own evaluation, which substantially increases the chances of infection by a parasitic virus.**

Shareware is also distributed by catalogues. This diminishes the dangers because the company offering the service has a vested interest in distributing ‘clean’ software. The software could possibly be contaminated at source, and users are at the mercy of the author’s own quality control procedures.

Shareware is a cheap way of obtaining software and much of it is of excellent quality. Some of the best known anti-virus programs (Flushot+, Virus Detective, Disinfectant etc) are shareware. Writing directly to the author and requesting a copy for evaluation is perhaps the safest approach. Some shareware packages now include a checksumming program and a list of checksums for all files contained. Providing that the checksum program is itself uninfected, this should expose modifications caused by a virus.

Public Domain Software

This software is available for use, free of charge, by anybody. The same distribution risks apply as with shareware and there are added disadvantages in that there is often no known author, contact telephone number or latest version of the program available. If there is no documentation or listed contact telephone number then use of such software is not advised. (*See Should We Trust Public Domain Anti-Virus Software?, VB, January 1990*).

Home Computers

The use of home computers for company work has caused computer virus outbreaks at a number of business sites. Very often the victim companies have had anti-virus procedures installed, but have suffered as a result of overlooking this route of infection. Dangers arise where company software is used on an infected home computer. Many home computers are used by more than one family member and children are liable to load and execute games software from a variety of dubious sources. Once program and data disks leave the company premises, control over them is completely lost. **The home computer is a proven source of virus infection and potentially a threat to a company's information.**

Service Engineers

Service engineers are often a fine source of the latest games, diagnostic tools and assorted software. Working daily with numerous PCs at a variety of different customer locations, they are a walking propagation medium for copyable programs.

Distribution or installation of 'fun' programs, and the use of copied diagnostic tools respresents a major opening for spreading a virus infection, not only within a company, but between companies.

Much can be done to prevent computer viruses from infiltrating an organisation by this route. Service engineers should use write-protected diagnostic floppy disks. Alternatively, and preferably, the inspected company should invest in a set of diagnostic tools, backups of which can be taken from a 'clean' machine. The service engineer can use the write-protected backup disks to undertake examinations but is forbidden to use his own disks. **At all costs, service engineers should be told not to distribute software to employees, or install unauthorised software.**

Shrink-Wrapped Software

Until quite recently, the standard anti-virus advice seemed to be "only use shrink- wrapped software". That was before Aldus distributed a shrink-wrapped and virus infected copy of their *Freehand* DTP package. There have been incidents of shrink-wrapped virus distribution in West Germany. An incident of Apple's *Multifinder* being distributed with a virus has also been reported from the United States. The most recent incident (reported in this month's editorial) occurred when the *United States Census Bureau* contracted the duplication of a disk to a commercial disk duplication service. The infected disk was then distributed to more than seven hundred research libraries.

Shrink-wrapped software is not virus-free by definition. There are, however, certain reasons why such software is more likely to be 'clean'.

Shrink-wrap software normally refers to commercial software which is delivered to the customer in sealed plastic or cellophane shroud. This is done for legal reasons and is not an anti-virus measure. Opening the package places an onus on the customer to comply with the manufacturer's terms and conditions governing the program's use. However, this protective shroud reduces the likelihood of malicious tampering with the program once it has left the manufacturer. It also provides some degree of assurance that the program is free from viruses.

Companies selling shrink-wrapped software have invested in their product range and a damaging virus outbreak undermines both reputation and sales. They also have a sufficient profit margin to provide elaborate quality assurance of the product range, ensuring both integrity and reliability. This software is produced in a controlled environment and certainly provides the highest degree of assurance as to program integrity available.

Dangers arise from disgruntled employees intentionally introducing a virus, or other malicious device, during the duplication process prior to packaging. There is also the possibility of re-sealing a modified or infected package. No known cases of these malicious activities have been reported.

Summary

To safeguard against accidental computer virus infection, all new software should be screened prior to its introduction on an active company computer or network.

Many companies have now installed internal software inspection groups sometimes referred to as *software quality assurance sections* or SQAS. The SQAS is responsible for the screening process and all users must submit new disks for inspection. All programs which pass the vetting process are logged, enabling management to maintain a constant record of software in use. Downloading programs from bulletin boards, transferring executables by modem and other potentially hazardous operations are monitored by this group. Shareware, public domain software and other possible virus carrying media, can be inspected and certified for use. Examination which combines searching for known viruses (scanning), and looking for modifications (checksumming) will provide a very high degree of assurance as to the integrity of each program inspected.

Much can be achieved by establishing a small core of technically proficient staff, knowledgeable about program threats and able to help general users. The burden of implementing anti-virus procedures is removed from users, who are simply instructed to submit new disks for inspection.

Next month: ***Company procedures to reduce the threat.***

CRISIS MANAGEMENT

Recovery From a Virus Attack

Despite all the precautions being taken, a virus may appear on almost any microcomputer system and possibly damage programs or data. Viruses often reveal there presence - programs may dissapear or malfunction, data files may be corrupted and/or interesting side effects may occur such as a Christmas tree appearing on screen (the XA-1 virus). Several viruses do not produce any obvious symptoms. These viruses hide until some specific condition or set of conditions is met and then they trigger, sometimes destroying everything they find.

Of course, strange symptoms may have other causes - Trojan horses and hardware failure, and common software bugs can account for unusual machine activity. However, in the event of a ‘worst case’ scenario where a computer virus is present and has done some damage what steps should be taken to eradicate the problem?

Don’t Panic

It is not always necessary to undertake drastic action such as reformatting the hard disk. Stay calm and obtain more information about the exact nature of the problem in order to determine a sensible course of action.

Know Your Enemy

No work should be performed on an infected computer while a virus is active in memory. Turn the machine off and reboot from a ‘clean’ system floppy disk - preferably the original write-protected one that should have arrived with the PC. (*See VB, Feb 1990, box, pp 4*). You should now try to identify the particular virus by running a virus-specific scanning program or by searching for identification patterns.

Assuming that a virus is found, the next step is to obtain information about its effects and particularly which areas of the system are affected. Most viruses only modify the programs or executable which they infect, but some viruses corrupt data, intentionally or otherwise.

Restoration

The major question when restoring programs and data is whether to use a disinfection program or restore from backups. In some cases there is no choice. If the virus has overwritten parts of the disk with garbage, backups should be used. A disk formatted by a virus presents great difficulties, but it is sometimes possible to recover the previous contents of the disk (the best known data recovery service in the UK is *S & S Enterprises*, Tel 0494 791900, ed.).

One problem encountered with backups is that the backup itself may be infected, particularly if the virus has been present for a considerable length of time. All programs restored from backups should be checked for infection as well. In the case of pure data files, once the infected machine has been booted from a clean, write-protected system floppy it is safe to transfer data files to diskettes. It should be noted that boot-sector viruses can infect the unused boot-sector of data diskettes and if a machine is accidentally bootstrapped from such a disk, the PC will become infected. **Never bootstrap from a non-system disk and educate users to remove disks immediately after transferring data.**

Another problem associated with backups is that they are rarely up-to-date. However, programs can be restored from write-protected master disks, backups of the master disk, or re-purchased if necessary.

If a virus only modifies executable files, a disinfection program may be all that is necessary. Using disinfection programs is easier than using backups but several problems may be encountered. In the case of a destructive, overwriting virus such as 405, disinfection is not possible - infected programs must be restored from a backup. A number of PC viruses modify the files they are about to infect, so their length becomes a multiple of 16 bytes, before appending the virus code. The disinfection program may be able to remove the virus but not the extra bytes added to the end of the program.

Some viruses, Taiwan and Jerusalem for example, are not always able to infect files but corrupt them instead- sometimes in a way which a disinfection program will fail to recognise. This situation is very rare, as the infected programs may not function properly. This will result in an early detection of the virus preempting its opportunity to spread.

It is important that all diskettes which have been in contact with an infected machine are disinfected.

Tracing the Virus

Sometimes it is possible to determine where the virus came from. If this is the case, the source should be alerted to the problem as soon as possible. Equally important, is that anyone receiving infected materials must be alerted. Any attempt to keep a virus outbreak secret will only result in the infection of more computers.

Prevention

Even though considerable effort may be expended in wiping out a virus, it will often reappear within a matter of weeks, or even days. This may result from employees bringing infected diskettes from home to work (or visa versa). **The installation of a checksumming or monitoring program is therefore strongly recommended.** (*See Anti Virus Tools, VB, April 1990, pp 14*).

Finally...

Locating a destructive computer virus before it triggers and being prepared to recover from the subsequent damage are the crucial elements in anti-virus warfare. Once a computer virus infection has been positively ascertained, time can be taken to assess the situation, seek advice and devise correct procedures to remove the problem.

LETTERS

AIDS Information Diskette Version 2.00 - Was it a Virus?

Sir,
I think that the AIDS disk does indeed contain a virus, but maybe not of the sort your writers are used to thinking of as one.

a) When you run the install program, it replicates onto your system. Since replication is all that is required for a virus according to the mathematical definition, this disk has a virus.

b) According to my understanding, after doing its damage, the rem#.exe program allows you to make a copy for redistribution to your friends. *Replication implies a virus!*

c) Since the autoexec.bat program in your system is modified and through the replication process causes another autoexec.bat to be modified in the same way on the next system, it is a virus.

d) Even if you claim ‘infection’ is required for a virus (not required by the mathematical definition), the AIDS disk ‘infects’ the autoexec.bat program with the rem sequence, and this sequence indirectly (via the rem#.exe program) causes a copy of itself to be placed in the next autoexec.bat program when you copy it to another system. Thus it is a virus.

e) Please note that the ‘diskcopy’ program is a virus in the mathematical sense. *All viruses are not bad!!!*

Fred Cohen.

Dear Editor,
I am not sure how mathematics influences the definition but it does seem far too wide to be usefully applied to the current problems of malignant computer code, particularly if its application ecompasses such programs as DISKCOPY.

I am unaware that a formal definition has been proposed for the type of program code commonly referred to as a ‘computer virus’. The term was presumably coined to indicate a similarity with biological viruses. My own working definition is:

Any of numerous kinds of computer program code, self-replicating within an active processing environment and often able to cause system malfunction.

I exclude such programs as the AIDS Information Diskette by virtue of the need for *self-replication* - ie no external agency is involved once the code is introduced into an active processing environment (by running it). I find this definition useful because it accurately defines the type of code with which we are concerned, effectively excludes code which cannot replicate without external assistance and yet does not insist that such code should be destructive. This agrees with Dr. Cohen’s final point that all viruses are not bad per se. It might be argued that a program displaying a message which said “Please Copy Me” could be described as self-replicating, but I would contend that the replication in such an instance would be via external agencies rather than the internal capability inherent in self-replication.

Jim Bates.

Dear Editor,
Further to Jim’s comments, it is important to point out that sequences of symbols are only viruses given an environment in which they replicate and/or evolve. In an environment where people act as a part of the operating system, their actions must be considered in determining whether a program is a virus, just as a biological virus which depends on a rat for its spread is still a virus, even though without the rat it would not work. No virus is universally ‘self replicating’. *Viruses only replicate in a suitable environment.*

The point is that a good mathematical definition does not leave the determination up to the viewpoint of the observer, but rather up to systematic analysis through a well defined set of rules. The formal mathematical definition of computer viruses was made in 1986 and has been widely published. This definition encompasses all of the things which Jim classifies as ‘viral’ but also encompasses many kinds of computer viruses which he may have overlooked in his consideration of the problem. For example, the formal definition deals with evolutionary viruses which are not ‘self-replicating’ because evolutions are not strict replicas. The formal definition is used to prove mathematical properties of viruses and defenses, and many companies who have ignored it have come up with easily defeated products, while those who paid attention have done a far better job of it. Jim’s definition is useful for telling a non-technical audience about a growing problem. Indeed his definition is probably much more useful for awareness than the formal definition will ever be. However, it is not helpful to solving the technical problems relating to computer viruses, whereas the formal definition is quite useful for this purpose. *If the normal operation of a computer includes people running programs and responding to prompts, then the AIDS Information Diskette is a virus.*

Fred Cohen.

Editor’s Note: *The mathematical definition was devised in Dr. Cohen’s PhD dissertation ‘Computer Viruses’ (University of Southern California) in 1986. It is published in the Elsevier journal Computers & Security (Vol 8 1989 pp 325-344) as Computational Aspects of Computer Viruses.*

KNOWN IBM PC VIRUSES (UPDATE)

Amendments and additions to the *VB Table Of Known IBM PC Viruses* as of 26 April 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.

EXAMINED VIRUSES

8 Tunes - CER: Probably from Germany. Virus infects .COM files as well as .EXE files. Length of virus code is 1971 bytes. When it triggers, one of eight different tunes is played.

```
8 Tunes        0B00 F3A4 1F33 C0CB 8CC0 0510 002E 0106; Offset 020
```

Amoeba - CER: 1392 bytes long. Awaiting disassembly.

```
Amoeba         CF9C 502E A107 0140 2EA3 0701 3D00 1072; Offset 0D1
```

Anarkia - CER: Minor variant of the Jerusalem virus, where the virus signature has been changed from ‘sURIV’ to ‘ANARKIA’. Will activate one day earlier than Jerusalem. Use search pattern for Jerusalem. Awaiting disassembly.

Durban, Saturday the 14th - CER: Adds 669 bytes to the end of infected files. On any Saturday the 14th, the first 100 logical sectors of drive C, then B, then A are overwritten with rubbish.

```
Durban         B911 00A4 E2FD B4DE CD21 80FC DF74 47C6; Offset 02F
```

Kennedy - CN: A simple .COM infecting virus, probably from Denmark. When an infected file is run, it will infect a single .COM file in the current directory, adding 333 bytes to the end of the file. The virus activates on three dates - June 6th, November 18th and November 22nd. On these dates it will display the message:

```
Kennedy er dod - laenge leve "The Dead Kennedys"
```

(*The Dead Kennedys, a San Fransisco based punk rock group, are well known political activists in support of causes such as the freedom of information. Ed.*)

```
Kennedy        E817 0072 04B4 4FEB F38B C505 0301 FFE0; Offset 0035
```

Pretoria, June 16th - CN: Overwrites the first 879 bytes of infected files with itself and stores the original 879 bytes at the end of the file. When an infected program is executed, the virus searches the entire current drive for .COM files to infect. On June 16th the execution of an infected file will cause all entries in the root directory to be changed to ‘ZAPPED’. The virus is encrypted.

```
Pretoria       AC34 A5AA 4B75 F9C3 A11F 0150 A11D 01A3
```

Prudents, 1210 - EN: 1205 bytes long. Will destroy the last 32 bytes of any file it infects. Awaiting disassembly.

```
Prudents       0E07 BE4F 04B9 2300 5651 E87E 0359 5EE8; Offset 055
```

PSQR, 1720 - CER: Another minor variant of Jerusalem. 1715/1720 bytes long. Signature is now ‘PSQR’. Awaiting disassembly.

```
PSQR           FCB8 0FFF CD21 3D01 0174 3B06 B8F1 35CD; Offset 071
```

Virdem - CN: 1336 bytes long. Virus was published in ‘Computer Viruses: A High-Tech Disease’ by Ralf Burger. One of the oldest viruses known, written in 1986. It was intended as a ‘demo’ virus, but has now been found ‘in the wild’.

```
Virdem         BE80 008D 3EBF 03B9 2000 F3A4 B800 0026; Offset 011
```

XA-1 - CN: The XA-1 virus overwrites the first 1539 bytes of infected.COM files with itself and stores the original code at the end of the file. On April 1st, a part of the virus activates - overwriting the boot sector with code which will cause the computer to ‘hang’ on next boot-up. The virus also activates on December 21st and stays active until the end of the year. It will then display a Christmas tree, and the text:

```
Und er lebt doch noch: Der Tannenbaum! Frohe Weihnachten
XA-1 BO2C 8846 FF8B 7E00 884E FE8A 4EFF 000D; Offset 01E
```

REPORTED ONLY

PrintScreen - DR

Korea - ?

Form - ?

VIRUS DISSECTIONS

Three dissections feature in this month's edition. The first is an analysis of the very widespread ***New Zealand*** *computer virus.*

New Zealand (also called 'Stoned' and ' Marijuana') is, most definitely, a 'wild' virus and we have received numerous request for information and help from infected sites in the United Kingdom and Europe. However, it is most prevalent in Australasia and the Far East.

The ***4K*** *virus (also called '4096' , 'IDF' and 'Frodo') is also a 'wild' specimen and is generally regarded as the most sophisticated IBM PC virus so far devised.*

Finally, Jim Bates looks at the ***'666'*** *virus from Bulgaria. Approximately fifty computer viruses originate from Bulgaria -reportedly from the capital city of Sofia. Two distinct sets exist - the 'Bulgarian fifty' written by 'TP' and a second group written by an unidentified author(s).*

The exact story behind the 'Bulgarian fifty' is shrouded in mystery, and there have been claims that these viruses were written for experimental purposes and that the use of Hamming code (self-correcting code) makes destructive or malicious reprogramming impossible. These claims are misleading because the self-correcting code itself can be disabled.

New Zealand - Causing Chaos Worldwide

Fridrik Skulason

This virus originated in New Zealand. The author, a student in Wellington, claims he never intended the virus to run wild. His story is that he destroyed all copies of the virus except one which he kept under lock and key at his home. This copy was stolen and used to infect computers at a local computer store. This happened in early 1988, but the virus has now spread all over the world, although it is still rather rare in Europe - at least in comparison to several Asian countries where it is understood to be rampant.

The New Zealand (aka Stoned/Marijuana) is a boot sector virus capable of infecting hard disks as well as diskettes. One in eight times a computer is booted from an infected disk the virus will display the on-screen message:

```
Your PC is now Stoned!
```

The virus seems to have been designed to be non-destructive, but it is capable of causing considerable damage due to the author's lack of technical knowledge.

Operation

When a computer is booted from an infected diskette, the virus becomes memory resident. It will first create a hidden 2K block at the top of memory by decreasing the value stored at the location 0040:0013. The virus then copies itself into this block. All other boot sector viruses currently known to be in circulation use a similar method to obtain a memory block for their use, but the size varies. When DOS is loaded it will use the value stored at the location 0040:0013 to determine the amount of usable memory, but the memory block reserved by the virus will remain hidden.

The virus then hooks into INT 13H, the disk I/O interrupt. If the computer was booted from a diskette, there is a one in eight chance that the screen message above will appear. The method used by the virus to determine whether the message should be displayed is to check if the bottom three bits of the byte at 0040:006C contain 000. This byte is incremented 18.2 times every second and is often used to provide a simple random number generator. If booting from the hard disk, the message will not appear.

If the computer was booted from an infected diskette, the virus will attempt to infect the hard disk. **The Partition Boot Record** (PBR, also called **Master Boot Record** or **Disk Bootstrap Sector**) is read into memory and examined. If no hard disk is present, the read operation will fail and the virus will skip this operation. If the first four bytes of the PBR do not match the corresponding bytes of the virus, the hard disk will be infected.

The PBR is stored on track 0, head 0, sector 1 of the hard disk. In many cases the rest of track 0, head 0 is unused, a 'feature' which the virus exploits by moving the original PBR to track 0, head 0, sector 7. The virus code is then written to track 0, head 0, sector 1, ready to be executed the next time the hard disk is booted. **It should be noted that the New Zealand virus is the only PBR-infecting virus currently known.**

Finally the virus will load and execute the code found on the original boot sector.

Infection

Infection of hard disks at boot-up has already been described, but diskettes are infected in a different way. When a program performs an INT 13H call, the virus intercepts it. The function number in AH register is checked to see if it is either 2 (disk read) or 3 (disk write). If not, the virus will pass control to the original BIOS routine. Otherwise it will check whether the calling program is attempting to use drive A:. If so, the virus will check whether the motor timeout counter at 0040:0040 contains zero - probably to avoid suspicious delays whenever the disk is accessed.

If these conditions are met, the boot sector is read from the diskette and checked for an existing infection. If no infection is found, the original boot sector is moved to track 0, head 1, sector 3. On a 360K diskette this is the last sector of the root directory, so no problems will arise unless the root directory is almost full, containing more than 96 entries. The virus completes the infection by copying itself to sector 1.

Problems Encountered

The New Zealand virus seems to have been designed to be 'benign'. However, the author did not consider the existence of diskettes containing more tham 360k of memory. On a 1.2 megabyte diskette, track 0, head 1, sector 3 is not at the end of the root directory, but rather the third directory sector. Since each directory sector contains 16 entries the New Zealand virus may prove highly destructive if the root directory contains more than 32 entries. A similar problem arises in the case of 3.5 inch diskettes.

Another potential problem is that track 0, head 0, sector 7 is not always unused on all hard disks. In certain cases it contains a part of the File Allocation Table, in such instances infection of the hard disk will cause considerable damage. Such damage would be reparable, however, because DOS stores two copies of the FAT on the hard disk.

Recognition

The easiest way to spot an infected diskette is the absence of the usual DOS messages in the boot sector and the presence of the text strings:

```
Your PC is now Stoned!
```

and

```
LEGALISE MARIJUANA!
```

Text strings can searched for by using the Search Disk for Data option provided by The Norton Utilities.

The virus signature stored in the first four bytes of the PBR sector is EA 05 00 C0.

Alternatively use the Search facility of The Norton Utilities (or a virus-scanning program) to search for the following hexadecimal patterns:

```
0400 B801 020E 07BB 0002 B901 0033 D29C; Offset 043
0400 B801 020E 07BB 0002 33C9 8BD1 419C; Offset 041
```

Variants

Several variants of this virus have been reported, but only one variant can be confirmed (*but see below*). The only difference is a relocation of the original PBR to track 0, head 0, sector 2.

One variant is said to display the second message 'LEGALISE MARIJUANA!' (included in the text strings above). Another variant is said to display no message at all making it harder to detect. The existence of both reported variants has not been confirmed.

Editor's note: *The Virus Bulletin has received a further variant of the New Zealand virus. The virus was reported to have resided on a free disk distributed with a computer magazine. It contains the text message 'Your PC is now Sanded!'. This is a very simple mutation of the existing New Zealand virus which could have been completed within a matter of minutes using a utility such as Norton. The addition of this text string to a search option may be warranted. The hexadecimal patterns above will detect this variant.*

4K - A New Level of Sophistication

4K is a memory resident, .COM and .EXE infecting virus from Israel. The size of most PC viruses is 1000 - 2000 bytes, which makes 4K one of the largest for as its name implies, its length is 4096 bytes. This virus is also known by two other names, 'IDF' and 'Frodo'. 'IDF' refers to the Israeli Defence Forces, where the virus was originally found late last year. 'Frodo' refers to the on-screen message displayed by the virus, as described later. Two versions of the virus exist, the significant differences between them surround the effects after triggering. The 4K virus may occasionally cause damage to files. It manipulates the number of available clusters, which can result in files becoming cross-linked.

Most of the virus code is devoted to making the virus 'invisible' when it is active. Other recent viruses attempt this, for example 666, described on page 13. The difference is that 4K uses a more comprehensive method - intercepting more DOS function than any other virus.

From a technical viewpoint, 4K is very advanced - it uses more undocumented features of the MS-DOS operating system than any other known virus, but it requires DOS 3.x because some of the undocumented features function differently under DOS 2.x and 4.x. It is clear that the author of the virus has a very good knowledge of the internal workings of the MS-DOS operating system.

Installation

When an infected program is run, the virus will attempt to disable tracing, by manipulating INT 1H (single step), the flag register and the 8259 chip.

It then alters the Memory Control Blocks (MCBs) and creates a 'gap', 6K in size just below the end of memory. The virus code is written there, overwriting the part of COMMAND.COM stored there. COMMAND.COM will therefore be reloaded and infected when the infected program terminates. The virus will intercept INT 21H, but not in the usual way - instead it will overwrite the first 5 bytes of the current INT 21 routine with a FAR JMP to itself. Most of the virus is devoted to processing various INT 21H functions. They are:

- Disk I/O Using File Control Blocks
- Function 0EH (Open file)

The virus first calls the original DOS function, and if the file is successfully opened the FCB is examined for the infection marker (year of creation > 2043). If an infection is found, the ‘file size’ field is decremented by 4096.

Function 11H (Find first matching file), Function 12H (Find next matching file)

The original DOS function is first called and if a file is found the ‘year’ field of the timestamp is examined. As in the ‘Open’ function, if the highest bit is set, the file is assumed to be infected and the ‘length of file’ field in the FCB is decremented by 4096.

Function 14H (Sequential read), Function 21H (Random record read), Function 27H (Random block read)

If a program is attempting to read from the beginning of an infected file, the virus returns the original contents of the first bytes, before they were overwritten by the virus.

Function 23H (File size)

The virus makes a copy of the FCB, opens the file and closes it again. The FCB will then contain various information, including the creation year and the true size of the file. If the file appears to be infected, the virus will once again subtract 4096 from the true length and return that value.

Disk I/O using handles/ASCIIZ

Function 3CH (Create file)

If the file being created has the ‘Readonly’, ‘System’ and ‘Hidden’ attribute bits set, control will simply be given to the original DOS function. This is probably done in order to avoid infecting IBMDOS.COM and IBMBIO.COM. (*under some versions of DOS both files have .SYS extensions, ed*). In other cases the virus will call DOS to create the file, close it and transfer control to the ‘Open file’ function.

Function 3DH (Open file)

First the extension of the file is checked, in a highly unusual manner. The ASCII codes for the three characters in the file extension are added together. If the sum is 223 or 226 the file may be subject to infection. Entering a command such as COPY X.WEF Y.BMP will result in the infection of X.WEF and Y.BMP. If the sum does not match, the virus will return to the original program.

4K maintains an internal table of open file handles and the process IDs of the ‘owner’ of the file. The term ‘process ID’ refers to the segment address of the Program Segment Prefix (PSP). If this table has any room left, the process ID and the file handle will be stored there, for future use by the ‘Close’ function.

Function 3EH (Close file)

When a file is closed, the process ID of the current process is first obtained and the table mentioned above is searched for a record containing it and the file handle. If it is found, the file is assumed to be a .COM or .EXE file and will be infected before it is closed.

Function 3FH (Read)

If a program attempts to read from an infected file, the virus will ‘disinfect’ the file as it is read, which may cause problems for checksumming programs because the file may appear unmodified.

Function 40H (Write)

When an infected file is being written to, the virus will disinfect it, only to re-infect it as it is closed.

Function 40H (Lseek)

When a program seeks to the end of an infected file, the reported value of the file pointer (position within the file) is decremented by 4096.

Function 4BH (Load/Execute)

Instead of loading and executing a program, the virus will only load it, using an undocumented sub-function of INT 21H (4B01H). The loaded file will then be checked for disinfection and any trace of infection will be removed before it is executed. However, if the file is not infected, the infection routine will be called.

Function 4EH (Find first), Function 4FH (Find next)

Just like their FCB counterparts, these functions only check whether the file is infected, in which case 4096 will be subtracted from the reported file length.

Function 5700H (Get date/time of file)

If this function is used to obtain the creation year of an infected file, the virus will subtract 100 from the value.

Function 5701H (Set date/time of file)

4K will not permit any program to set the date of a file later than December 31, 2043.

Miscellaneous Functions

Function 30H (Get DOS version)

This is perhaps the most interesting part of the virus. It will first check the current date. Before September 22nd nothing happens, but after that date a mysterious routine is executed. One version of the virus will cause indefinite results, probably causing the machine to 'hang'. This version contains some garbage bytes as well as the POP CS instruction, which is illegal on '286, '386 and '486 machines. The other version of the virus attempts to overwrite the boot sector with the '*Frodo*' code described later.

Function 37H (Get/set switchchar, device availability)

The reasons for hooking into this function remain unclear.

Infection

4K may infect programs when they are executed or when an executable file is closed. The second possibility implies that copying a non-infected program (on a machine where the virus is active) will result in the infection of the target file as well as the original.

The virus starts by checking whether or not the target file is .EXE. It reads the first two bytes and compares them with 5A 4D or 4D 5A. Virtually all .EXE files start with 4D 5A, but it is a little known fact that MS-DOS will also assume that a file starting with 5A 4D is also an .EXE file.

The virus code is written to the end of a file. In the case of a .COM file, the first three bytes are overwritten with a JMP into the virus code. EXE files are modified in a different way, by changing the information in the header, including the initial CS and IP values.

As might be expected, 4K removes the 'read-only' attribute before infecting files and restores it afterwards.

The directory entries of infected files are modified using a method similar to that adopted by the Vienna virus. Vienna modifies the 'seconds' field of the timestamp, setting it to an impossible value (62), but 4K adds a century to the year. A file created in 1990 will appear to have been created in 2090. When the user issues the DIR command, no change in the date is visible because MS-DOS only displays the last two digits of the year. This method will not work after the year 2007, however, because the last date which MS-DOS can represent is December 31st 2107. This date modification prevents reinfection and allows the virus to identify uninfected target files.

There is thus an easy way to check whether 4K is active in memory - just set the date to January 1st 2044, create a small file and issue a DIR command. If the file is reported as having a length of 4 Gigabytes and having been created in the year 100, the virus is active. The reason that DOS reports a length of 4 Gigabytes is that the virus subtracts 4096 from the file size which it assumes is infected. This produces a negative number, which DOS treats as an unsigned binary equivalent.

'Frodo' Code

4K contains a routine which attempts to modify the boot sector to display the on-screen message

```
FRODO LIVES
```

when an infected machine is booted. This text, in large letters, is surrounded by a moving pattern. This routine does not , in fact, work. The keyboard interrupt is re-directed, probably to disable the Ctrl-Alt-Del combination, forcing the user to turn the PC off and to boot from a non-infected diskette.*Editor's Note: Frodo is one of the 'hobbit' characters in J. R. R. Tolkien's fairy-tale 'The Lord of the Rings'.*

Final Notes

4K conceals itself quite efficiently but it can be detected in a number of ways. The loss of 6K of memory provides an immediate indication to the observant user that the machine is infected. Changing the date as described above will also reveal its presence.

A hexadecimal search pattern for this virus is:

```
E808 0BE8 D00A E89A 0AE8 F60A E8B4 0A53; Offset 239
```

Apart from these characteristics the virus will probably remain unnoticed until September 22nd, 1990, although it may be detected on some computers earlier, as it seems unable to co-exist with some items of network software.

Users of checksumming software, and indeed all anti-virus programs, should take not that such software can only be effective if run from a clean system disk.

If the 4K virus is active in memory it could fool checksumming programs into 'thinking' that system attributes are unmodified.

High-level security checks using anti-virus software must be run from a write-protected 'clean' system floppy disk. Anti-virus programs invoked from AUTOEXEC.BAT can be undermined by second generation viruses. However, running secure checksumming programs from the hard-disk will still provide a high degree of protection on a day-to-day basis.

For information about the write-protected system floppy disk see VB, February 1990, page 4.

666 - The Number of The Beast

Jim Bates

666 (aka V512 and Number of The Beast) is one of the '*Bulgarian 50*' viruses and although it has no trigger routine there are several features which make it worthy of analysis. A trigger routine would be difficult to add to the code because of size limitations which leads me to suspect that this virus is meant to demonstrate the writer's ingenuity and is not an example of malicious code.

666 is a parasitic virus which becomes resident in the system when the code is executed. It only infects files with the letters 'CO' as the first characters of the file extension. This obviously targets .COM files but it may also infect COBOL source files which conventionally have an extension of .COB and other files which match this specification. Other criteria for infection are that the file must have a length between 512 bytes and 65,023 bytes (inclusive) and, importantly,**the file length must be such that there is at least one free sector of space between the end of the file and the end of the last allocated cluster.**

This highlights a weak area in the MS-DOS file storage system of which virus researchers have long been aware. Disk space allocation within MS-DOS uses predetermined chunks called 'clusters'. The size of a cluster is set when the drive is first formatted and on most systems with more than around 10 Mbytes of overall disk space the cluster size is set to 4 sectors (or 2048 bytes). Since file sizes are rarely an exact multiple of clusters, there is usually unused space at the end of the file. The main advantages of using this space are (a) *DOS will not overwrite it*(b) *it is considered 'lost' to DOS operations unless special provision is made to access it.* It is thus an ideal place to 'hide' virus code so that scanning programs cannot find it. Of course, some method must still be used to ensure that such code is loaded and executed. 666 attempts to fool both resident and scanning software into 'thinking' that nothing is amiss with the host file. Obviously, there will be some files which do not have sufficient unused space within the final cluster. However, 666 checks for such files and does not attempt to infect them.

The virus has been padded with a three byte 'signature' consisting of the characters '666' (hence its alternative names) to make its length 512 bytes (exactly one sector). It is positioned on the disk to overwrite the first sector of the host file. The original contents of that first sector are written beyond the end of the file in the aforementioned free space.

Installation

The code determines the DOS version of the current operating system and then collects the INT 13H vector from page zero of memory. The DOS version is then checked and if the minor part (after the decimal point) of the version number is .30 then a little known and undocumented DOS function is called which swaps out the original (ie: pre-system load) INT 13H entry point (this usually points to the ROM INT 13H routine). The swapped out address is pushed onto the stack and the function is called again to swap the vectors back. This restores normal operation but leaves the required address on the stack. This address is popped from the stack to take the place of the INT 13H vector collected from page zero. If the minor DOS version is not .30 then the swapping function is not used and processing continues with the page zero DOS INT 13H vector.

The swapping process recovers an entry point into the disk I/O services which is not usually monitored by anti-virus software. However, it is possible to hook monitoring software into this function and also to monitor the swap function in order to intercept this virus installation routine.

Installation then continues by storing the relevant INT 13H vector within the virus' code segment and then collecting the INT 21H vector - again by direct access to page zero of RAM. The offset portion of the INT 21H vector is checked for a value of 121H and if found, the indicated segment is checked for the virus' presence.

If 666 is resident, processing branches to the portion of the code which is processed after the virus has been made resident. If the virus is not resident, the code locates the address of the first Disk I/O buffer which DOS uses. These buffers are exactly 512 bytes long and usually have 16 bytes as a header so there is enough space for this virus code to be installed.

Once the virus has been moved into the buffer, its address is removed from the Disk I/O chain and the INT 21H vector is modified to point to the interception routine within the newly re-located virus code.

The code then overwrites a small section of the transient portion of the command interpreter. This is to ensure that when the program terminates, COMMAND.COM will be reloaded and infected.

666 actually attempts to infect COMMAND.COM the very first time it is executed.

The code then goes on to check whether the current program is a command process (ie: COMMAND.COM itself) or is running as a child of DOS. If it is a command process, processing terminates back to DOS and since the virus code is now resident, the reloading of COMMAND.COM will ensure that it is infected. When an infected program is run for the first time on a clean system (ie: COMMAND.COM is not infected), the host program itself will not be executed - but if run a second time, it will be! Subsequent program executions, after the 'parent/child' check, causes the data in the original first sector to load into the appropriate area (overwriting the recently loaded copy of the virus).

The method of accessing beyond the end of the file involves direct manipulation of the DOS System File Table (SFT). This technique, used in several places throughout the code, allows the opening of a file for READ access (**thus not alerting resident anti-virus monitoring software**), changing the SFT to allow WRITE access. The file length and date/time fields are also modifiable. However, during program execution, only the file length field is adjusted; by adding 512 so that the original data can be read from the disk.

Interrupt Interception

Only the INT 21H vector is intercepted during installation and the virus allows all function requests except 3FH (READ), 3EH (CLOSE) and 4B00H (LOAD & EXECUTE).

Read Function

The READ intercept attempts to 'hide' the the virus code in a manner reminiscent of the Brain virus. When a READ request is received, the current position of the file access pointer is noted and then the READ performs correctly. Next the file access pointer is checked to see if the read request was for a portion of the file within the first sector. If it wasn't, processing is returned to the caller, but if it was, the file time stamp field is checked for 1FH (31 decimal) in the seconds bits. This is equivalent to a setting of 62 seconds and is one of the markers used by 666 to indicate an infected file (the other being the presence of the virus code itself).

If the file is marked as infected, the SFT is accessed again to modify the file size field and the original first sector is read from the last cluster. Then the file size field is restored to its former value before processing returns to the caller. **Thus the virus effectively conceals itself by supplying the caller with the correct data rather than the virus code. This means that simple scanning programs will not detect the virus code on an infected system! This emphasises the absolute necessity of ensuring that the system is 'clean' before searching for virus code on disk.**

LOAD & EXECUTE and CLOSE Functions

The intercept routine for these functions is substantially the same except that when closing a file, the file handle is first duplicated and subsequent operations are carried out on the duplicate handle. Loading a file for execution results in the file being opened (in READ mode) for virus operations. In both cases the file handle being used is closed before the original request is allowed to continue normally (using the original file handle). The interception checks the file for existing infection and if it is not infected, checks its suitability for infection by determining unused file space in the final cluster.

Infection

After the routine opens (or duplicates) a file handle for the target, the file position pointer is set to zero (Beginning of File) and the SFT access privilege field is changed to allow WRITE access.

Then the vector for INT 13H is changed to that collected during the Installation phase. Remember that the virus code will usually have collected this vector during the initialisation of the system (ie: via an infected COMMAND.COM) or from the undocumented vector swap interrupt. *It is unlikely that this vector will be monitored by anti-virus software.* 666 does not use INT 13H directly, but INT 21H functions associated with file I/O use it and could thereby alert monitoring software.

The infection check routine also re-vectors INT 24H (Fatal Error Handler) to point to an IRET instruction within the code. This disables error reporting.

Once these two interrupt vectors are modified, the code checks the time stamp field for the 62 second marker. If this marker is not found the file is tested for extension 'CO', otherwise the extension check is by-passed. There seems no valid reason for this alternative checking, maybe the programmer had other options in mind and overlooked this sloppy coding. Whatever the reason, the assumption is that a file with the 62 second marker set will be a .CO? file.

From this point on, if a check fails, the handle is closed and processing is return to continue the original INT 21H function call.

The next check ensures that the target file is between 512 and 65,023 bytes long.

A further test looks at the SYSTEM attribute setting of the target file and, if set, rejects it.

The final check before examining the file for existing infection involves testing the file length against the number of sectors per cluster and calculating unused space available in the final cluster.

Once the target file has passed all these checks, the first 512 bytes of the file are read into a buffer. The virus uses the high part of the Interrupt Vector Table as a buffer, thus overwriting all Interrupt vectors above 7FH.*This is a flaw within the virus since an increasing number of machines, most network software and several high-level languages use these interrupts and the destruction of the vectors will cause system failure and immediate alarm.*

Once the start of the file has been read, it is checked against the existing virus code for infection. If infected, the time stamp field is set with the 62 second marker and the file is closed. If the file is not infected, the contents of the buffer are appended to the end of the file and the virus is written to the first sector. The file size field is modified during this process but restored afterwards to leave the appended code outside the size setting. The date/time field remains largely unaltered (except for the 62 seconds) as a result of direct access to the flags field of the SFT. This ensures that the visible directory entry for the file remains unchanged.

Within the infection routine there are two calls to Function 40H of Interrupt 21H.**These are immediately obvious to monitoring software and therefore provide a useful detection point.**

Problems With Scanning Programs

The use of the LOAD & EXECUTE function for file infection is standard amongst virus writers but the use of the CLOSE function is of much more concern. Dark Avenger also subverts this function.

Problems centre around simple scanning programs which examine a file by opening it using the DOS file I/O functions. Once scanning has been completed the file is closed and - if a virus of this type is present - becomes infected. Thus the scanning program becomes the agent for infecting every suitable file on the disk at one go! The answer is to restrict the use of scanners to a known clean system, ie keep the scanning program on a write-protected system floppy disk and only use it from there. The only scanners which should be used on an automatic basis (invoked by AUTOEXEC.BAT) are those which collect file information on an absolute sector basis and thereby do not use the DOS file I/O facilities. If the method used by the scanner is not known, assume that it does use DOS file I/O and take appropriate precautions.

Some Final Thoughts...

My impression on disassembling 666 is that its author considers himself a cut above other virus writers. The code makes use of undocumented and obscure features of DOS and almost shouts "*See how clever I am*" and yet ignores such giveaways as the overwriting of the Interrupt Vector Table and the plain use of the WRITE function 40H of INT 21H.

Other programming flaws include the method of checking for the virus' existence in memory. This will cause multiple installation of the virus code if any subsequent program revectors INT 21H. Complications arise if an infected program is copied on an uninfected system. In this case, since most COPY routines work on a byte by byte basis (and not cluster by cluster) the infected file will lose the contents of the unused space at the end of the file and while the copied file will still contain the virus, it will no longer perform its original function. Assuming that a virus should not announce its presence by destroying its host, this is a failure in the code.

The final flaw in this virus, of interest to technicians, is that infection can be stopped simply by ensuring that all files occupy an exact number of clusters. Thus no free space is left and the virus cannot replicate.

666 is reported to be a 'lab' virus, intended to assist in the development of anti-virus software. This may be the case but the fact remains that it uses dangerous techniques which could be modified with malicious intent to exacerbate the virus threat.**There are enough genuine viruses around without the irresponsible development and distribution of new ones.**

The following hexadecimal search pattern can be used to locate this virus:

```
5A52 0E07 0E07 0E1F 1EB0 5050 B43F CBCD 2172;
Offset 0A3
```

Editor's Note: The Number Of The Beast is referred to in Revelations, Chapter 13, The New Testament. According to The New Oxford Annotated Bible the number 666 is the sum of the letters of a man's name, foretold as a false prophet and Anti-Christ.

TOOLS & TECHNIQUES

Backup!

Backing up data has always been standard practice on mainframe systems but has only recently been regarded as serious precaution for PC users. Regular backups can protect users against a range of threats including power failure, crashes, human error and malicious programs. Without backups there is no way to restore unique data. **Backups are the single most important precaution that can be taken against computer viruses and are vitally important in case of attack by a destructive virus. Unduplicated data stored on hard disk will be irretrievably lost in the event of a virus such as Datacrime triggering.**

As part of the backup procedure, the master disks for all software (including the operating system) should be write-protected and stored safely. All backups (and other disks) should be accurately labelled (date and time created) and recorded in an inventory. Critical data (e.g. marketing databases, company accounts) should be stored both on-site (in a fire proof safe) and in a secure, remote location (off-site).

Backup procedures must be periodically tested by performing complete restorations of the system. There is little point in taking backups unless the data can actually be restored. It is also important that backup procedures are carefully monitored and adhered to.

Fastback Plus - Backup Made Easy

Fastback Plus is a backup program for MS-DOS computer systems. The Fastback package has been available for some time and has gained a good reputation. This article describes briefly the features offered by Fastback Plus and discusses in more detail the new features offered by Version 2.

It is vital that a good backup program offers maximum flexibility in the manner in which files to be backed up are specified. Fastback Plus scores heavily on this point, as files can be chosen by any desired combination of named directories, named files, and files within specified date parameters. The type of backup can also be varied. The user can choose: a complete backup; only those files that have changed since the last backup of any type; or the files that have changed since the last complete backup. Backups can be made to/from any defined DOS device including floppy disks, hard disks, tape or cartridge drives, and external hard disks. This is far superior to the crude facilities offered by the BACKUP and RESTORE programs included with the MS-DOS operating system.

Version 2 of Fastback Plus offers improved data compression; the manual claims that backups not only compress into a smaller space, but they are written to disk at a faster rate. I tested this by backing up the hard disk of my portable computer (*see Technical Details below*). The last backup taken with version 1 of Fastback Plus required eleven 3.5 inch floppy disks (720Kbyte) to backup 6.5 Mbytes. Using Version 2, I backed up 6.76 Mbytes on to 8 disks of the same capacity - a significant improvement. This was completed in 5 mins 17 secs, a rate of 1.28 Mbytes per minute. Impressive.

Version 2 backups can be protected from unauthorised use by a password. This facility should be used with caution as once a password has been defined for a backup set, restoration is impossible without the correct password. The addition of a password means that Fastback Plus backups cannot be restored by any version of Fastback Plus earlier than 2.10. Don’t forget that if the password is not known, for whatever reason, there is no way to restore the data.

One final feature worth mentioning is that Fastback Plus can now restore to a PC a backup set that has been generated on a Macintosh. To achieve this, you obviously need a copy of Fastback for the Macintosh. This is a useful way to transfer files from a Macintosh to a PC, a feat that is often difficult to achieve. There are problems in compressing Macintosh filenames within the MS-DOS convention of 8 character filename plus 3 character extension, but these can be circumvented.

Fastback Plus now controls the proliferation of history files which previously had to be done manually. A history file is an ASCII file containing full details of all files contained in a backup set. This file is kept on the disk being backed up, and can be inspected without resorting to searching the actual backup disks. When a backup is made to disks that previously contained a Fastback backup, the relevant history file(s) on the hard disk are automatically deleted.

I liked the original version of Fastback Plus. Version 2 is even better. Fastback Plus changes taking backups from a time consuming chore into an easily executed task. This can only be to the good, and should encourage more people to invest in the best anti-virus measure of them all - **regular backups**.

K.J.

Technical Details

Product: Fastback Plus

Developer: Fifth Generation Systems Inc., 10049 N.Reiger Road, Baton Rouge, LA 70809, USA. Tel 504 291 7221

Vendor: Riva Ltd., 3 Bentley Industrial Centre, Bentley, Farnham, GU10 5NJ, UK. Tel 0420 22666, Fax 0240 23700

Availability: IBM PC/XT/AT, PS/2, or compatible with 330K of RAM, MS-DOS version 2.01 or higher, and a hard disk.

Version Evaluated: 2.10

Serial Number: 113-0409565, supplied on both 3.5 inch and 5.25 inch floppy disks.

Price: £152.50

Hardware Used: Toshiba 3100SX, battery powered laptop portable with a 16MHz 80386 processor, one 3.5 inch (720K) drive, and 40Mbyte hard disk running under MS-DOS v4.01.

PRODUCT EVALUATION

Dr. Keith Jackson

Vi-Spy: A Virus Diagnostic Utility

Vi-Spy is an MS-DOS utility that checks system memory, the boot sector of a disk, and/or files stored on disk for the presence of a virus. If a virus is detected, Vi-Spy can erase the locations that contain the virus. Vi-Spy is sold as a software package in the normal way, and has an optional "Subscription Service" which provides free telephone support, and automatic updates for an annual fee (*see Technical Details below for pricing information*).

Vi-Spy is provided on both 3.5 inch (720K) and 5.25 inch (360K) floppy disks, so most types of user are catered for. This is necessary as the documentation that comes with Vi-Spy decrees that Vi-Spy should not be installed on a hard disk. **This is sound advice. If Vi-Spy is always executed from a write protected floppy disk, then there is no possibility of the executable program itself becoming infected with a virus.** Vi-Spy is not copy protected, so it should be possible to make as many backup copies of the Vi-Spy floppy disk as necessary, but for some strange reason, the manual insists that backup copies should not be made. This is in spite of the fact that Vi-Spy does not seem to be copy protected (as far as I can tell a copy works in exactly the same way as the original), so why not advise users to take backup copies and store them in a safe place? Curious. The manual does state that if the original floppy disk ever develops a fault, it will be replaced free of charge.

The manual that accompanies Vi-Spy is 50 pages long, A5 size, and bound as a small booklet. It explains very clearly what a virus is, how a virus infection can spread, how a virus is introduced into a computer system, and how to avoid being affected by viruses. The problem of viruses being known by various names is discussed at length, and wherever possible all of the known alternative names for each virus are provided. This discussion is one of the best parts of the manual. Fourteen pages of the manual are devoted to a section entitled "Virus trouble-shooting", which explains in detail how to use anti-virus software, and gives examples of good practice. In short, I think that the Vi-Spy manual is one of the clearest I have come across and is written in terms which can be followed by someone totally unfamiliar with viruses. Its major fault is the lack of an index, a problem which could easily be cured.

Writing a clear manual must have been greatly helped by the inherent simplicity of Vi-Spy itself (a compliment not a criticism). If a virus is found, the only thing that Vi-Spy will actively do is erase it. No attempt is made to retrieve files that have been infected, clear out an infected boot sector, or otherwise regain space on a disk which has been occupied by a virus. Vi-Spy does not mess about, it just overwrites the virus, and makes sure that any future attempt at resurrecting the virus is guaranteed to fail by performing multiple overwrites (seven in total) of the virus affected locations. Multiple overwrites are required as it is technically possible (but difficult) to retrieve data from a file that has been overwritten just once.

"Vi-Spy does not mess about, it just overwrites the virus ..."

The Vi-Spy disk contains a README file which is an overview of how the operating system manages memory usage, an explanation of how Vi-Spy checks its own executable file for alteration(s), and an explanation of how to use Vi-Spy on a file server. The README file also contains an up to date list of all the viruses that can be detected by Vi-Spy. Since the manual was printed, September 1989 according to the first page, Vi-Spy has increased its store of known viruses from 22 to 46. This is still only about half the current total of viruses described in the latest issue of *Virus Bulletin*, but all of the most prevalent viruses are present, and no doubt the developers of Vi-Spy will have increased the total number of viruses yet again before this review gets into print.

One nice feature of Vi-Spy is that a user does not need to specify a drive, the default setting is to search all available drives. Any floppy disk drive that does not actually have a disk in it is ignored, Vi-Spy catches the error produced by the operating system, and does not let the familiar 'A(bort), R(etry), I(gnore)' error message from MS-DOS intrude.

I tested Vi-Spy against all of the viruses listed in the

Technical Details section below. With just two exceptions it detected all of them correctly. The first problem that I found was that Vi-Spy insisted that the Italian virus test disk contained the “TYPO BOOT (Fumble)” (their name) virus. Vi-Spy knew that the virus was a boot sector virus, but found the wrong one. Vi-Spy also reported the South African virus as the Icelandic virus, but as this is the second package that I have tested recently which detects the South African virus as the Icelandic virus (*see the review of Virusafe in the April 1990 issue of Virus Bulletin*), I must now begin to be suspicious of my test set of viruses on this particular point. Therefore the only problem definitely attributable to Vi-Spy was in wrongly identifying the Italian virus.

On every single occasion that a disk containing a virus was presented to Vi-Spy it detected the presence of a virus. As seems quite common (see reviews of virus detection products in *Virus Bulletin* over the past six months), there can be problems with identifying a virus correctly. In this case the Italian virus is consistently identified as another virus. This may (again) be more a comment on nomenclature difficulties than anything else, but to be safe when using a program which searches for virus signatures, I would always recommend using more than one such program (from different authors), and comparing the results whenever a virus is found.

Probably the main criterion with which to judge an anti-virus program which searches for virus signatures is speed. How quickly can the program search through all of the files on a hard disk? Execution speed is important for the simple reason that if inspection takes an inordinately long time, then it will simply fall into disuse. Humans are easily bored.

I measured Vi-Spy’s searching speed in comparison with two similar programs: SWEEP (version 2.08) from Sophos Ltd., and SCAN (version 1.75V50) from McAfee Associates. All three programs were instructed to inspect the complete hard disk of my Toshiba portable (see *Technical Details* below for processor and clock speed information) which contained 18.2 Mbytes in 1467 files. SWEEP searched for 78 virus signatures and needed 4 mins 11 secs to complete this test, SCAN looked for 52 virus signatures and required only 1 min 46 secs, while Vi-Spy searched for 46 virus signatures in just 1 min 33 sec. In common with SWEEP, Vi-Spy searches system files (.SYS) and overlay files (.OVL) as well as executable files (.COM and .EXE). Vi-Spy is easily the fastest of the three programs tested.

The manual makes it clear that Vi-Spy is written entirely in assembler, and an optimising assembler (OPTASM) at that, so Vi- Spy’s fast searching speed is perhaps not too surprising. SWEEP is written in C, for reasons of portability. I do not know what language SCAN is written in.

In conclusion, I found Vi-Spy simple to understand (it just detects viruses and destroys them by overwriting), easy to use, and very fleet of foot in searching for virus signatures on a disk.

Technical Details

Product: Vi-Spy

Developer and Vendor: RG Software Systems Inc., 2300 Computer Ave, Suite A-7, WillowGrove, PA 19090, U.S.A., Tel:215 659 5300

Availability: IBM PC/XT/AT, PS/2, or compatible with 128K of memory running MS-DOS v2.00 through v4.01.

Version Evaluated: 2.0, dated 1st quarter 1990

Serial Number: None visible

Price: US$ 250, periodic and emergency update service US$ 150 per year.

Hardware Used: Toshiba 3100SX (a battery powered, laptop, PC) with a 16MHz 80386 processor, one 3.5 inch (720K) drive, and a 40 Mbyte hard disk, running under MS-DOS v4.01. Also 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. Both computers used during testing.

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 and 2, Datacrime II, 405, Fu Manchu, Jerusalem 1 and 2, Traceback, Suriv 1.01, Suriv 2.01, Suriv 3.00, South African 1 and 2

BOOK REVIEW

Computer Viruses And Anti-Virus Warfare by Jan Hruska - Ellis Horwood - 128 pp.

Dr. Jan Hruska has been involved with the development of anti-virus software since the appearance of the Brain virus in the United Kingdom. He is well known in the UK for his lively and technically ambitious presentations on PC viruses at computer security conferences and seminars. His book,*Computer Viruses And Anti-Virus Warfare*, is written in an equally lively manner and may best be described as an 'introduction and overview' of the subject.

Hruska concentrates on viruses affecting IBM PCs and compatibles - the area with which he is familiar. The introduction and overview of threats explains viruses but also contains examples of Trojan horses, worms and logic bombs. The book appeared in February '90 and thus contains details of the AIDS Information Diskette which is listed as a Trojan (*see page 7*) and the WANK worm (*VB, April 90*).

A concise documentation of the bootstrap process is included to show files and features which are vulnerable to virus attack. The range of media and proven carriers of infection are discussed and the author offers his advice on such contentious issues as shareware, public domain software, bulletin boards and support engineers. Although never explicitly saying so, it is obvious that the author is distrustful of all four categories. This section is very helpful in identifying all the possible methods of virus transmission. It is marred by an overly brief section on safeguarding PC networks from virus infection and file-server security. Books on LAN security are readily available from specialist publishers, but there is still a dearth of detailed information about containing and eradicating network virus infections - a subject for a book in itself.

A section on virus structure follows which will enlighten the bemused newcomer to the bizarre world of computer viruses. Parasitic and bootstrap sector viruses are explained and memory resident and non-memory resident infections, encryption, interrupt interception and other essential principles are outlined.

Chapter 4 contains the *Virus Bulletin Table of Known IBM PC Viruses* (published with our permission) as it appeared on the *VB* database in January of this year. The problems surrounding virus names and aliases are noted and there is a discussion of the all important hexadecimal pattern and its use for virus detection. The book had gone to press by the time that the notorious 1260 virus appeared and thus there is no discussion of encrypting, self- modifying viruses and the techniques necessary to locate them. This was discussed by Dr. Peter Lammer (Dr. Hruska's colleague) in last month's edition of*VB*.

A speculative survey of the types of groups or individuals who write computer viruses is included. This was written with the assistance of a psychiatrist who has diagnosed a number of 'technopaths'. Criminal, deviant, terrorist and intellectual motivations are analysed and various groups identified.

For the technically inclined, there is a short but involved section on virus disassembly using DEBUG followed by an intriguing discussion on forensic evidence including methods to determine the assembler, programmer's nationality, style and so on. Mutations, 'improvements' and other results of tinkering are included but there is no reference to 'second generation' viruses. This is the book's major shortcoming - the advent of viruses such as 666, 4K and 1260 has already caused a certain revision of defensive methods and tools.

Company procedures and methods to minimise the threat are detailed as well as the preparation of contingency plans and post-attack recovery. Anti-virus software methods are compared. The author concludes that the only long-term strategy for virus prevention will necessitate the use of checksumming software based on cryptographic checksums which will detect any virus, present or future. With the current state of technology, this seems a reasonable statement, although there is still no consensus about the future direction which anti-virus software must take.

The source codes of a virus-specific detection program (in 'C' and Assembler) and a virus-non-specific checksumming program in 'C' appear towards the end of the book. They are also available on disk from the publishers.

The book is carefully indexed and includes thorough listings of anti-virus software developers, relevant books and journals.

The strength of *Computer Viruses And Anti-Virus Warfare* is that it is logical in the way it addresses the subject, clear in its explanations and devoid of the sloppy mistakes which have undermined similar works. The book is also refreshingly straightforward and contains no incomprehensible algebra, medical analogies or theories. It serves both as a sound, factual introduction and as a reference work.

Computer Viruses And Anti-Virus Warfare

ISBN: 0-13-171067-2

Author: Jan Hruska

Price: £17.95

Available from bookshops or direct from: Ellis Horwood Ltd, c/o International Book Distributors Ltd, 66 Wood Lane End, Hemel Hempstead, Herts HP2 4RG, UK. Tel 0442 231555.

END-NOTES & NEWS

Sophos Ltd have made a **training video** in partnership with *Airtech Security Ltd* about the virus threat. The video is designed to help teach all PC users (including temporary staff and new employees) about computer viruses. Part 1 of the video demonstrates possible effects of viruses and shows users how to react should they encounter the problem. Part 2 describes prevention, detection and containment and is aimed at technical staff. Details from Karen Richardson, Sophos, UK. Tel 0235 559933.

Jim Bates has produced the first of a series of **technical information sheets** and software as part of a virus information service which is available on subscription. The service is aimed at PC technicians and provides detailed information about individual viruses. Details from *Bates Associates*, UK. Tel 0533 883490.

Software Under Siege: Viruses & Worms is a new book from *Elsevier Advanced Technology*. The book is a non-technical guide providing case histories, symptoms and measures for prevention. Details from Elsevier, UK. Tel 0865 512242.

The *US National Computer Security Association* has expanded its **BBS virus information** service. Downloadable files include anti-virus programs, diagnostic tools, product reviews, virus information etc. The BBS can be called at 300, 1200 or 2400 baud using no parity, 8 data bits, 1 stop bit on number USA 202 364 1304. There is no charge for calling the BBS and no uploading to the board is permitted.

Computer Viruses and Computer Software Vaccines for Software Protection is a new bibliography from the US *Department of Commerce's National Technical Information Service* (NTIS). The report was compiled between January 1988 and October 1989. Details from NTIS, 5285 Port Royal Road, Springfield, VA 22161, USA.

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

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, of from any use or operation of any methods, products, instructions or ideas contained in the material herein.