May 1991

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

The Silly Season

The commercial competition between rival software developers is intensifying as can be seen by the number of glossy colour advertisements for anti-virus programs which are currently blossoming in the trade press.

In the United States, the battle-lines seem clearly drawn between *Symantec*, *Ontrack Computers* (who have just been appointed the distributor for *Dr. Solomon's Anti-Virus Toolkit*), *Microcom* and *Central Point Software* - the latter company being a new contender to the anti-virus throne with its stunningly original trademark '*Central Point Anti-Virus*'. In the UK, a comparative backwater in the advertising stakes, traditional gentlemanly rivalries continue with regular sales-pitches from both *S&S* and *Sophos* accompanied by occasional outbursts from other developers.

The advertising battle has provided diverting entertainment from the tedium of investigating the seedier side of computing. The advertising executives and other creative 'types' have surpassed even the wildest expectations with adverts which are striking, humorous, hilarious (although probably unintentionally so), and inevitably, in a few instances, misleading. Whether it be *Sophos'* morbid colour reproduction of the gastroenteritis microbe dramatically subtitled 'KILLER VIRUS?' or *Ontrack's* offering replete with rabbits and the whimsical caption "Finding a Virus Can Be A *Hare* Raising Experience" - there will be something to impress, amuse, irritate or offend every budding computer 'virologist'.

Other illustrious examples include the 'ZAP!' advert from British company *PC Security*, showing what appears to be a family of greenfly leaping from a floppy disk but being struck down by a Lichtenstein-esque thunderbolt. This advert is a near perfect synthesis of pop-art and high comedy with its earnest intonation that 'Computer viruses are no joke. Neither is Eliminator'.

Equally amusing is the latest offering from *Symantec* in the United States - Peter Norton (is it really Peter Norton?) has undergone the ignominy of being photographed in a surgical face-mask accompanied by the caption "Warning. Peter Norton has determined that PC viruses can be hazardous to your data." Presumably the agency was unaware of Mr. Norton's previous assertion that computer viruses were a myth on a par with the alligators said to inhabit the sewers of New York City. Interestingly, the same advert states that Mr. Norton developed the *Norton Anti-Virus* - which is not true.

So rapidly-reproducing rabbits, disease-bearing microbes, high-voltage greenfly, and masked medicine-men are the current vogue. What *is* surprising is the absence to date of sexual imagery to promote these products; surely the advertising gurus can add a sexual slant to a business which regularly talks of 'dirty PCs' (pardon me constable!), clean floppies and the need for 'safe hex'. If scantily clad women can boost the sales of chocolate, could they not likewise swell the sales of disk utilities and anti-virus toolkits?

The images are intriguing, but the accompanying text is usually no less so. *Central Point Software*, for instance, would have us believe that its product "*is the only anti-virus program with File Immunization that protects your files so they can never be infected again.*" If this is true for every virus specimen then *Central Point* has achieved a remarkable feat entailing the development of a completely new operating system, the release of which we await with baited breath. The product is also said to destroy "*twice as many viruses as other programs*" (by ''other programs'' do they mean text editors, graphics packages, what exactly?). Moreover, "*it's the only program, that protects you when you're inside Windows*" and the "*only one that works across your Novell network*" - which is odd, because to quote rival developer *Symantec*: "*The Norton Anti-Virus is 100% compatible with PC networks like Novell... and of course, it's Windows compatible.*" This bickering over GUI and network compatibility, endemic to the industry, is predictable - so much so that some anti-virus advertisements avoid the issue altogether. *Ontrack*, for instance, has chosen instead to exploit the issue of virus numbers with a bold proclamation that the *Toolkit* "*checks for over 350 different viruses - 100 more than its closest competitor*". Meanwhile, *Symantec*, oblivious to this fact, has the unsporting audacity to state that *Norton Anti-Virus* "*detects over 540 viruses*" (presumably this is *not* the close competitor to which *Ontrack* refers).

The contradictions, claims and counter-claims are myriad. The advertising literature is usually not dishonest; it is, after all, simply a question of definition as to how many 'separate' viruses a package detects. *Central Point* is perfectly justified in its assertion that a program can inoculate files permanently - the company simply omits to state that this can only be achieved against a single virus or closely related subset. There is little point in objecting to adverts which, in the words of one UK civil servant, are ''economical with the truth''. Advertising must make an impact and any good advertising 'rep' will advise his client to manipulate the basic instincts of fear, sexual desire, greed and vanity. First prize will go to the software company which successfully exploits all four to promote something as mundane as a software security package.

Watch this space...

TECHNICAL NOTES

Disinfection Problems With Jerusalem

The thirty or so known variants of the Jerusalem virus make up a sizeable percentage of virus infections reported worldwide. In many cases users attempt to disinfect programs using software disinfection routines supplied with many anti-virus packages, instead of replacing them with clean copies taken from the original master software. However, this is not always found to be the simple solution intended, and the following questions are frequently asked:

➤ Why does the program not work after disinfection?

➤ Why is the program longer than before it was infected?

➤ Why is the program shorter than before it was infected?

➤ Why does my anti-virus program refuse to disinfect?

The Jerusalem virus infects COM files without problems, adding 1808 bytes at the beginning of the target file and 5 bytes at the end. Disinfected COM files should always be identical to the program before infection.

This is not the case for EXE files, which are infected in a different way. Instead of using the actual length of the file, and appending the virus code at that location, the virus uses the length according to the file header. In most cases this length is equal to the length of the file, but this is not always the case. The size according to the header can never be larger than the actual length, as this would produce an "Error in EXE file" message when the program was run, but if it is shorter the Jerusalem virus will overwrite a part of the file.

When the actual length of the program is more than 1808 bytes longer than its recorded length according to the header, the failure to infect the file properly becomes obvious. The anti-virus program should be able to determine that the program cannot be disinfected and the only solution is to replace it. However, a problem arises when the difference is *less* than 1808 bytes, as this cannot be detected. The file may *appear* to be disinfected but it may crash when executed, as up to 1808 bytes may be missing from the end.

This situation is rare, but since this phenomenon has serious implications for recovery, **it is recommended especially for EXE files infected with the Jerusalem virus that they are not disinfected using a program, but deleted and restored from clean write-protected copies of the master software**. Even if a program is disinfected successfully it might be slightly longer than before it was infected. Those extra bytes are the result of padding, performed to make the file length a multiple of 16 before infecting it. The extra bytes cannot be removed, but they do not affect the execution of the program.

Interrupt Tracing

As anti-virus programs improve and become more common, virus writers have been forced to resort to more sophisticated techniques to bypass them.

One recent method discovered in a virus currently being analysed in Austria involves interrupt tracing. The virus places the processor in single-step mode and issues an interrupt call, usually INT 13H or INT 21H. After each instruction is executed, an INT 01H is automatically issued. The INT 01H routine has a simple function - it examines the segment part of the return address on the stack.

The virus uses this servicing routine in an attempt to determine when it has reached the original interrupt processing function, **before** this was intercepted by any resident anti-virus program. If the INT 13H call is traced, for example, the virus may determine that it has reached its target when the return address is above C800H:0000H. The virus may then store the return address as its entry point for this interrupt function and call that address *directly* instead of performing a normal interrupt call which would be intercepted by the anti-virus program.

Anti-virus programs could include a simple defence against this. If the interrupt is intercepted by the anti-virus program, the interrupt handler of the anti-virus program can simply check whether the Trace flag is set and produce an appropriate warning message if this is the case.

Interrupt tracing is yet another programming technique designed to subvert TSR monitoring programs - very few, if any, of which are currently aware of this stealth characteristic.

False Positives and Shareware Inoculation

A number of anti-virus programs include a search for the ubiquitous 62 stamp in the seconds field which is used by a number of computer viruses as a self-recognition signature. The Vienna virus was the first to use this method, but it can be found in a number of Vienna variants including the self-modifying encrypted 1260 virus.

Anti-virus software programs which search for this signature (without supporting search data for verification) have recorded a very high percentage of false positive indications in clean shareware files. The reason for this is that many shareware programs released in 1989 and early 1990 adopted this signature as a means of inoculation against virus infection. Obviously, this technique provides no generic defence against virus infection as it only defends against the virus subset which uses this specific signature.

The widespread use of this marker in shareware devalues the use of the 62 seconds stamp as a search signature. Software packages which search for this signature should include additional virus-specific data in order to confirm the presence or absence of virus code in files with altered timestamps.

BRIEFING

Current Concerns - Source Code, BBSs and Encryption

In recent months three developments liable significantly to exacerbate the computer virus problem have been discerned. The factors currently causing the greatest concern within the research community are:

- The general circulation of virus source code.
- The distribution and collation of computer viruses via Bulletin Board Systems.
- Encrypted viruses which result in dynamic or variable code.

Regarded individually, these factors influence either the propagation of virus code or the technical methods by which it can be detected. In combination, these developments represent a significant escalation of the virus threat.

Source Code

It would appear that the long-standing assertion that a virus must *execute* before it can replicate can now be laid to rest. Researchers worldwide are now beginning to discover source code (not disassemblies and not binary) which is in turn distributed (again uncompiled) within the research community. With tongue in cheek, this process could be described as 'virus propagation' by another name!

However, the availability of source code is vexing because it hugely increases the ease with which variants of an individual computer virus can be produced.

A few years ago, the options available to the aspiring virus writer were to a) *write an original virus* b) *introduce variations into a virus already in circulation* c) *use Burger's published V1 virus from his book 'Computer Viruses: A High Tech Disease'*.

Writing an original virus is immensely more troublesome than adapting code from an existing 'template'. Similarly, the work involved in disassembling compiled virus code, analysing its operation and identifying code segments suitable for optimisation and/or alteration, is an order of magnitude greater than introducing changes into source code. It is no accident that so many of today's PC viruses trace their lineage directly from Burger's source code. With source code now in circulation, the time and effort required to modify virus code is minimised; a development which may well result in a huge increase in the number of separately identifiable virus specimens.

BBS Virus Collation and Distribution

The legitimate research community has recently discussed plans to distribute virus binary and disassemblies by BBS - both *CARO* in Hamburg (see *VB*, April 1991) and *Bates Associates* in the UK intend to introduce this service presently, with a heavy emphasis upon encrypted files, access control and user-authentication.

However, *unofficial* Bulletin Board Systems which offer virus code for download and which encourage the uploading of such code have now been established in the United States, Germany, Italy, Bulgaria, Sweden, Czechoslovakia and the UK. The code, which is often available in both binary and source code form, is offered for the purposes of "virus research", but is not stored using encryption and little if any vetting of those who log-on to the systems is undertaken by the SysOps running them.

In the United Kingdom, the appropriate authorities have been informed of this development, but the *Computer Misuse Act 1990* does not expressly render the possession, storage or wilful downloading of computer virus code a criminal offence.

BBSs are often used as a forum for the discussion of criminal or anti-social activities. In the United States there are BBS conferences covering a range of 'interests' from bomb-making to paedophilia - information can be exchanged with a virtual guarantee of anonymity for the participants while evidence can be erased by the SysOp, either directly at the keyboard or remotely using a modem. Interestingly, the anonymity afforded by Bulletin Board Systems appeals to dating agencies (particularly homosexual ones) and various other groups in which discretion is valued. Not surprisingly, this anonymous forum is now being used to facillitate the collation and dissemination of computer viruses.

The BBS 'network' usually gleans the numbers of all active BBSs using sequential number dialling software such as the freeware program *AIO*; it is safe to assume that the dial-up numbers of boards offering computer viruses for download, once established, will be posted and distributed very widely. Should this happen, a worldwide proliferation of virus samples and outbreaks will result.

Encrypted Viruses

The subject of self-modifying encryption has been covered in some detail in past editions of *VB* (March 1990 p.12, April 1990 p.10, November 1990 pp. 13-16, April 1991 pp.18-20).

A further discussion follows because of the grave implications for anti-virus software implicit to this development.

Perhaps the greatest single obstacle which impedes the development of virus-scanning programs is to render every infected file different. Some viruses achieve this by encrypting most of their code with a key which varies from infection to infection. Self-modification is usually performed either by inserting irrelevant instructions into the decryption routine (e.g. the 1260 virus) or by modifying the registers used in order to alter the binary representation of the instructions (e.g. the Proud virus).

Encryption algorithms used in computer viruses are comparatively trivial: XOR-ing with a simple key is the most frequently used method. The reason for this simplicity is that the virus writer intends to achieve variability rather than confidentiality in his code. Moreover, the decryption key must be accessible to the decryption routine, so the use of a sophisticated algorithm usually serves no purpose.

However, the introduction and wider adoption of self-modifying encryption will inevitably have an impact upon anti-virus software development.

Traditional hexadecimal patterns are already unusable for detecting certain encrypted viruses. Some anti-virus programs (IBM's *VIRSCAN* for example, see page 16) allow for the use of "wildcards" in patterns to represent the variable bytes to be found in patterns extracted from viruses such as Ontario and USSR-1594.

Other programs describe the ways in which a virus may be located in a high-level language such as C or a meta-language (such as the *VIRTRAN* Virus Transaction Language from *S&S Enterprises*) sufficiently flexible and fast to describe and identify of each such virus.

This approach will cope with perhaps 10,000 viruses, beyond which such a program would probably become too slow and cumbersome to be used in practice. By the time saturation occurs, viruses undetectable by scanners will have appeared. Sparse infection (whereby the virus self-recognition signature is so common as to be unusable for detection) and even non-self-recognition strategies will combine to render virus-specific detection impracticable due to the unacceptably high percentage of false-positive indications inherent to such a strategy.

Conclusions

Should Bulletin Board virus distribution and source code circulation continue without restraint, the number of individual virus specimens will increase dramatically (but not necessarily exponentially) over the next year. Concurrently, virus scanning looks set to become significantly more complex; these factors combined do not bode well for either computer-users or anti-virus software developers.

VIRUS BULLETIN
EDUCATION, TRAINING AND AWARENESS PRESENTATIONS

Education, training and awareness are essential to an integrated campaign to minimise the threat of computer viruses and malicious software.

Virus Bulletin has prepared a presentation designed to inform users and/or line management about this threat and the measures necessary to minimise it. The standard presentation consists of a two hour lecture supported by 35mm slides, followed by a question and answer session. Throughout the presentation, technical jargon will be kept to a minimum and key concepts will be explained in accurate but easily understood language. However, a familiarity with basic MS-DOS functions is assumed. The presentation can be tailored to comply with individual company requirements and ranges from a basic introduction to the subject (suitable for relatively inexperienced users) to a more detailed examination of technical developments and available countemeasures (suitable for MIS departments).

The aim of the basic course is to increase user awareness about computer viruses and other malicious software without inducing counterproductive 'paranoia'. The threat is explained in comprehensible terms and straightforward, proven and easily-implemented countermeasures are demonstrated. An advanced course, which will assist line management and DP staff, outlines varying procedural and software approaches to virus prevention, detection and recovery.

The presentations are offered free of charge except for reimbursement of travel and any accommodation expenses incurred. Information is available from the editor, *Virus Bulletin*, UK. Tel 0235 555139.

COUNTERMEASURES

Virus-Specific Monitoring Programs

Just like virus scanners, all virus-specific monitoring programs are only effective against **known** viruses. Consequently, frequent updates are necessary to keep them current as new viruses appear. Several different types of monitoring program exist, but they all have certain features in common, such as a database of information about the viruses they are intended to intercept. Unfortunately this database grows as the number of viruses increases, usually with a corresponding increase in the memory requirements of these programs.

Unlike generic monitoring programs such as *Flushot+* (see *Virus Monitoring Software - An Endless Battle*, *VB*, July 1990, pp. 15-16) which watch for all suspicious activity, this class of memory-resident program should only alert the user if a virus is detected with absolute certainty. Therefore, they do not generate as many false alarms as generic monitors, but new viruses will generally be able to bypass them.

There are a number of different virus-specific monitors which adopt various *modi operandi*. The major differences between these programs centre around the activities which are monitored and the operations which activate them. Some virus-specific monitors incorporate all of the following scanning routines while others use only one interception method.

Disk Scanning

On a PC no signal is generated when a new diskette is inserted into the floppy drive of the machine but on most XTs, ATs and later machines it is possible to check the Diskette Change Line Status to determine whether a new diskette has been inserted. A monitoring program could scan the boot sector of any new diskette inserted into the system, checking for any known viruses. Although the mere act of inserting an infected diskette into a drive cannot cause an infection, the diskette may be left in the drive negligently, causing a boot sector virus to infect the system upon the next boot. Obviously, this risk is reduced by scanning diskettes when they are inserted.

Scanning on Program Execution

A highly effective way to keep known viruses out of a computer system is to scan every program for viruses before it is executed. While this may add slightly to the loading time of the program (commonly called execution overhead), it will effectively prevent any of the viruses known to the scanner from being executed on the system. Program execution overhead is unlikely to be a noticeable problem on fast processors such as the 808286 or 808386. Moreover, because only one file is scanned upon invocation (as distinct from a non-TSR scanner which scans all executable files on disk), the relevant inconvenience of this approach is negligible.

‘COPY’ Scanning

This method extends the preventative approach of keeping viruses out of the system by scanning executable programs for viruses while they are being copied to disk. Should a virus be located the monitor will warn the user of its presence and request permission to proceed with the copying process, abort or undertake some form of disinfection or deletion of the offending file. This process impairs speed particularly noticably when many small files are copied.

Are You There?

Many viruses use a special interrupt function to determine whether they are already resident. A monitoring program may intercept the interrupt requests and either return a “Yes-I am resident” response, which will normally trick the virus into executing the original program, or raise an alarm - notifying the user of the infection.

The interrupt calls can also be used to check whether various viruses are active, but as always there are some associated problems. Only some viruses use interrupt functions for this purpose. Other viruses may, for example, use a flag in a fixed location in memory - often changing a byte in the upper half of the Interrupt Table. Finally and most significantly, non-resident viruses do not use any special interrupt functions at all, rendering this approach inapplicable to their detection. The table on the opposite page lists the ‘Are You There’ calls and responses for the IBM PC viruses which employ them.

Conclusions

Virus-specific monitoring software has grown in popularity recently, particularly in organisations using standalone PCs in which the relative invisibility of these programs and their ability to intercept viruses before they can be copied to the hard disk or executed is valued. However, as with all TSRs, these programs are prone to crash if they conflict with other memory-resident software. It is also difficult (but by no means impossible) to make virus monitoring software *Windows 3* compatible. As is the case with all virus-specific software regular updates are necessary, which can result in immense logistic problems if these programs are installed on a site-licence basis involving the upgrading of several hundred PCs each month.

The appearance of stealth viruses such as INT13 (*VB*, March 1991, pp.12-14) and Flip (which employs ‘interrupt stripping’, *VB*, September 1990, pp.18-20) proves that attempts are being made to undermine memory-resident anti-virus software. These attempts are usually targeted at generic monitors but it is probable that subversion of specific monitors will become more prevalent as these programs become more widely adopted. (See *TSR Monitors and Memory Scanners - The ‘Playground Approach’ to Virus Detection*, *VB*, March 1991, pp.18-19.) However, the ability to detect the commonest viruses known to be in the wild before they can even be copied or executed should not be underestimated.

IBM PC VIRUSES - MEMORY-RESIDENT 'ARE YOU THERE?' CALLS

Virus	INT 21 Call Parameters (hex)	Return Value if Virus is Resident
Perfume	AX=0B56	AX=4952
Dutch-555	AX=30F1	AL=00
Oropax	AX=33E0	AL=FF
Agiplan	AX=357F	DX=FFFF
Shake	AX=4203	AX=1234
Invader	AX=4243	AX=5678
MG	AX=4B04	carry clear
699	AX=4B04	AX=044B
Plastique	AX=4B40	AX=5678
Murphy-2	AX=4B4D	carry clear
Plastique-2576	AX=4B50	AX=1234
Murphy-1	AX=4B59	carry clear
Nomenklatura	AX=4BAA	carry clear
948	AX=4BAF	AL=FA
Magnitogorsk	AX=4BAF	AL=AF
Lozinsky	AX=4BDD	AX=1234
707	AX=4BFF	BL=FF
Justice	AX=4BFF	DI=55AA
516	AX=5252	BX=FFEE
1067	AX=58CC	carry clear
Klaeren	AH=76	AL=48
SVC	AH=83	DX=1990
Vriest	AH=89	AX=0123
Carioca	AH=90	AH=01
789	AX=A1D5	AX=900D
Eddie-2	AX=A55A	AX=5AA5
600 or Voronezh	AH=AB	AX=5555
Datalock	AH=BE	AX=1234
1049	AX=BE00	carry clear
Slow	AH=C0	AX=0300
Solano	AH=C0	AX=1234
905	AX=C301, DX=F1F1	DX=0E0E
Sverdlov	AX=C500	AX=6731
Yankee or MLTI	AX=C603	carry clear
Westwood	AH=CC	AX=0700
Fellowship	AX=D000	BX=1234
Diamond-A	AX=D5AA	AX=2A55
Diamond-B	AX=D5AA	AX=2A03
Dir	AX=D5AA, BP=DEAA	SI=4321
Durban	AH=DE	AH=DF
Jerusalem	AH=E0	AX=0300
Armagedon	AH=E0	AX=DADA
8-Tunes	AX=E00F	AX=4C31
Mendoza	AH=E1	AX=0300
Fu Manchu	AH=E1	AX=0400
Anarkia	AH=E4	AH=04
Spyer (INT13)	AH=E7	AH=78
Terror	AX=EC59	BP=EC59
Jerusalem-G	AH=EE	AX=0300
Frere Jacques	AH=F0	AX=0300
GP1	AH=F7	AX=0300
dBASE	AX=FB0A	AX=0AFB
Flip	AX=FE01	AX=01FE
2468	AX=FE02	AX=01FD
Black Monday	AX=FEDC	AL=DC
Sunday	AH=FF	AX=0400
PSQR/1720	AX=FF0F	AX=0101
Ontario	AX=FFFF	AX=0000

IBM PC VIRUSES (UPDATES)

Amendments and additions to the *Virus Bulletin Table of Known IBM PC Viruses* as of 18th April 1991. The full table was published in the January 1991 edition of *VB*. Hexadecimal patterns can be used to detect the presence of the virus with the 'search' routine of disk utility programs or, preferably, can be added to virus scanning programs which contain pattern libraries.

Seen Viruses

Cascade YAP - CR: A variant with a slightly modified encryption routine.

```
  Cascade YAP    0F8D B74D 01BC 8206 3124 3134 464C 75F8 ; Offset 013
```

Darklord - CER: A variant of the Terror virus, this 921 byte virus contains the string "Dark Lord, I summon thee! MANOWAR". Awaiting further analysis.

```
  999 Darklord   8EC0 488E D88B 1E03 008 3EB6 503C 326A3 ; Offset 096
```

Enigma - ER: A variant of the "Old Yankee" virus, claiming to have been written by the same author as HIV. It is 1624 bytes long, and is detected by the pattern published for Old Yankee.

Evil Empire - MR: Virus infects Master Boot Sector and relocates original boot sector to Sector 6, Head 0, Track 0. Virus displays a text message questioning United States involvement in the recent Gulf War. (*VB*, May 1991.)

```
  Evil Empire    734C 80FC 0275 4731 C08E D880 3E6C 0416 ; Offset 1E in MBS
```

GP1 - CER: This is a Dutch *Novell Netware*-oriented variant of the Jerusalem virus.

```
  GP1            03F7 2E8B 8D07 00CD 218C C805 1000 8ED0 ; Offset 124
```

HIV - CER: This virus is based on Murphy and contains a text message claiming it was written by "Cracker Jack" in Italy.

```
  HIV            2BC3 1BD1 7204 2906 0600 8BF7 33FF 0E1F ; Offset 5c6
```

Jeff - CN: Just like the Klaeren virus, Jeff can not successfully infect files longer than 4096 bytes. The virus is 812 bytes long, (not 814 as reported earlier). When it activates it may overwrite sectors on the hard disk.

```
  Jeff           B89B FF8E C0B9 3F00 33D2 32E4 8BD9 268A ; Offset 034
```

Klaeren - CER: This 974 byte virus contains a serious error, which prevents it from infecting successfully any file larger than 4096 bytes. This encrypted virus contains the text string "Klaeren Ha, Ha!" (Klaeren: the name of a professor in the school where the virus was written.)

```
  Klaeren        5351 E800 005B 81EB AF03 B9A5 0380 37 ; Offset 3b0
```

Magnitogorsk, 2560 - CER: This virus has not been fully analysed yet, but it contains a greeting to a Mr. Lozinsky, who seems to be the author of an anti-virus program.

```
  Magnitogorsk   2E8B 851F 003D FFFF 7413 BE3E 0003 F7B9 ; Offset021
```

Plague - CR: A simple 591 byte overwriting virus, based on the Leprosy virus.

```
  Plague         8A27 3226 0601 8827 4381 FB83 037E F1EB ; Offset 021
```

September 18th - CEN: This virus activates on September 18th, after 7:00 AM, overwriting the hard disk. Two variants are known, 789 and 801 bytes long, but the virus adds 1-16 extra bytes to programs before infecting them.

```
  Sept 18th      7502 32C0 3CFF 7502 B001 5051 CD26 83C4 ; Offset variable
```

South African 408 - CN: A 408 byte version of the South African virus, partially rewritten to foil scanners, but with no new effects.

```
  South Afr.408  1E8B ECC7 4610 0001 E800 0058 2D5A 0090 ; Offset 04f
```

Vriest - CN: This virus adds 1280 bytes in front of the COM files it infects. When it activates it will display "Something's coming up ...", produce a high-pitched sound for a few seconds, and finally display "Vriest of g greets Vic ear Moeli~".

```
  Vriest         80F2 E5B4 02CD 21EB F0B0 B6E6 4333 DBE4 ; Offset 159
```

Reported Only

834 - CR: Infects COM files other than COMMAND.COM and modifies the Master Boot Sector so the computer hangs when booted.

Arf, Rigor Mortis - CN: A 1000 byte variant of Vienna, reported to have been written by a person or group named Thor.

Captain Trips - CER: A variant of Jerusalem, 1808/1813 bytes long.

Lazy - CR: A 720 byte virus which installs itself in unreserved memory causing a system crash if many large programs are run.

Kemerovo B - ?: A US modification of the Kemerovo virus.

Striker 1 - CN: A 461 byte virus which does nothing but replicate.

VIRUS ANALYSES

Jim Bates

1. Evil Empire - Hypocrisy Writ Large

This virus was received on 9th April 1991 from software developer and researcher Ray Glath in the United States with a note stating that it was isolated at a University in Canada.

Like the Saddam virus, Evil Empire appears to have been written during the run-up to the Gulf War, following the Iraqi invasion and occupation of Kuwait in August last year. There have also been persistent rumours from California of a ‘Scud’ virus - presumably this only hits one in every three PCs!

General Structure

This specimen is a simple boot sector virus which infects the Master Boot Sector of the first fixed drive of a machine and all densities of floppy disks (in any drive). The virus code occupies somewhat less than a single sector, but data comprising an encrypted message takes up a further sector. The virus is similar to most other boot sector viruses in that it installs its own memory resident INT 13H intercept routine which both protects the virus code on disk and ensures that every opportunity is taken to infect other disks.

There is a trigger routine which is invoked at random and this produces a political message which in this instance also displays breathtaking hypocrisy.

Virus Installation

The virus is loaded at boot time from either floppy or hard disk, and immediately collects the INT 13H vector addresses from low memory and stores them within the virus code. The available memory pointer maintained by the machine BIOS is then accessed and decremented to remove 2 Kbytes from normal use. Processing then moves the virus code up into the memory area above the new limit and passes control to it.

Once in high memory, the virus issues a reset request to the disk BIOS routine and then examines a flag within its own code which indicates whether the virus was loaded from fixed or floppy disk. If the flag shows that the virus was loaded from a floppy disk, processing branches to a routine which checks whether the first physical fixed drive is infected - and if not, infects it. In either case, the second sector (containing the encrypted message) is then loaded into high memory.

The final phase loads the original boot sector (from either fixed or floppy disk) into the normal boot area and passes control to it, enabling the boot process to continue normally.

Infection

This virus replicates by intercepting INT 13H disk function requests according to the following criteria:

Floppy Disk Read requests are intercepted and the virus examines the boot sector of the target floppy. If the target is infected, the trigger conditions are tested before processing continues normally. If the target disk is **not** infected, the floppy boot sector is collected and the disk is infected by writing the first sector of the virus to the boot sector (Track 0, Head 0, Sector 1). The original boot sector of the floppy is written to Track 0, Head 1, Sector 2 and the second sector of the virus code is written to Track 0, Head 1, Sector 3.

Hard disk access requests are dealt with differently - only those dealing with the Master Boot Sector are subverted. Read requests are redirected to collect the original boot sector, while Write requests are changed to a disk reset instruction.

In the same way as the New Zealand (Stoned) virus, the method that this virus uses to store its own code and the copy of the original boot sector on both hard and floppy disks will cause data loss on certain types of machine and disk. For example on floppy disks other than 360 Kbytes, the second sector of the virus code and the original boot sector are stored in sectors that DOS reserves for directory information. Destroying data in these sectors prevents DOS from accessing file information referenced within them.

The Trigger Routine

The trigger routine (see Figure 1) in this virus occurs on a random basis and consists solely of displaying a message on screen which raises questions of “ethics” and “purity” concerning American involvement in the Gulf War.

```
I'm becoming a little confused as to where the
''Evil Empire'' is these days. If we paid
attention, if we cared, we would realise just how
unethical this impending war with Iraq is, and
how impure the American motives are for wanting
to force it. It is ironic that when Iran held
American hostages for a few lives the Americans
were willing to drag negotiation on for months;
yet when oil is held hostage, they are willing to
sacrifice hundreds of thousands of lives, and
refuse to negotiate .......
```

Figure 1. Politics, pacifism and programming: screen message from Evil Empire, the first explicitly anti-war computer virus. Viruses offer a powerful platform for political protest worldwide. The phrase ‘‘Evil Empire’’ is taken from a speech by former U.S. President Ronald Reagan in which he vilified the Soviet Union. Self expression, freedom of speech and thought ... but at whose expense?

The message leaves me breathless since it happily talks about “unethical” conduct and “impure” motives and yet chooses to use unethical, impure and downright criminal methods to propagate itself! This is surely the high-tech equivalent of writing a note exorting the reader to “Stop Crime!” and then wrapping it around a brick before hurling it through a window.

Detection and Removal

Apart from the message, this virus is unremarkable. It is classified as an ordinary Master Boot Sector virus, infecting both fixed and floppy disks. Its presence on some machines may cause data loss as a result of FAT corruption. Removal of the virus code may be effected by booting from a known clean system floppy and then copying Sector 6, Head 0, Track 0 to the Master Boot Sector of the fixed disk. Repair of corrupt sectors may be possible depending upon the machine in question. No encryption is used and so a simple recognition string can be extracted as follows:

```
734C 80FC 0275 4731 C08E D880 3E6C 0416 ;
Offset 1EH in the MBS
```

2. MACHOSOFT - Spaghetti Code

One of the older parasitic viruses - MACHOSOFT - was selected for examination recently when it was reported by a user in the UK as causing problems in booting his machine.

This virus was first noted by researchers in October 1989 and reports and analyses have been circulating since then. Unfortunately, these reports were a little vague and since one of them described sections of the virus code as “very professional”, it seemed that a more detailed analysis was called for.

MACHOSOFT is a minor variant of the SYSLOCK virus altering the location of its “corruption storage file” and the nature of the trigger routine. Most of what follows in this analysis applies to both viruses.

Operation

This is a parasitic virus which infects both COM and EXE files (including COMMAND.COM) and far from being “professional”, the code shows evidence of having been written by someone relatively new to programming.

The virus does not become resident, operating solely on a “one shot” infection method each time an infected file is run. When infecting COM files, the virus replaces the first 35 bytes of the file with a combination of code and data which first runs a simple decryption toggle routine and then calculates a new segment value before completing a Far return through a stack address transfer. This is unusual, since a simple JMP instruction is all that is required to get to the appended code! However, the technique allows the remainder of the code to be simplified as beginning at offset 0 in the new segment.

As a result of this initialisation, the virus code must save various register values before proceeding. With EXE files this initialisation routine is not necessary, since appropriate modifications are made inside the EXE header to enable processing to start at the correct segment and offset. The starting point for EXE infections is slightly different and this enables the virus to maintain a flag which indicates whether it is functioning from within a COM file or an EXE file.

> *“The over-riding impression is one of a newcomer to computing, who has spent many hours building a ‘seat of the pants’ program on a piecemeal basis, probably with a ‘Beginners’ Book of Programming’ in one hand and a comforter in the other.”*

Once processing gets under way in earnest, the code calls a routine to examine the environment, looking for a switch consisting of “VIRUS=OFF” (in SYSLOCK this is “SYSLOCK=@”). If this is found, processing transfers to the host program without further action. Otherwise, the extension type flag is tested and processing branches accordingly.

For COM files, the DOS version is checked. If this is 2.xx, a further random test is instigated which produces a high probability that processing will return immediately to the host program. For some reason this test is not processed when the virus is run from an EXE file.

From this point, for both EXE and COM files, the code is tediously familiar. First a very primitive search routine locates files for potential infection and selects one of them at random.

An indication of the style of this code appears in the file finding routine, where each individual letter of the file extension is tested to locate COM or EXE files.

Once a suitable file is found, it is checked for infection in one of two completely separate ways, depending on the extension. Infected COM files contain the following sequence of hex bytes at offset 10H:

```
  39H, 28H, 46H, 03H, 03H, 01H
```

EXE files are checked in a different way by testing the EXE header checksum word for a value of 7CB6H. Files which fail the relevant test are ‘passed’ for infection. These criteria apply for both SYSLOCK and MACHOSOFT.

Trigger Routine

The trigger routine in this virus is unusual in that it searches the default drive, sector by sector, looking for the word “Microsoft”. In the MACHOSOFT virus, this is changed to “MACHOSOFT” whereas in SYSLOCK it becomes “MACROSOFT”.

The progressive nature of this search is maintained by the virus in a Hidden, System file called “IBMNETIO.SYS” within the root directory of the default drive (in SYSLOCK the file is called “KEYB.PCM” and is created in a randomly chosen subdirectory). Two bytes in this file contain the logical sector address of the last sector corrupted. Each time the trigger routine is executed (which may not be every time an infected file is run), this file is opened and the corruption routine continues from where it left off.

Once the end of the disk has been reached, the count is reset and the search begins again from the first sector. The trigger routine itself only attempts to corrupt 32 sectors at a time and again displays a clumsy approach to coding the search.

Conclusions

Far from being professional, this is poor “spaghetti” programming. Within the program there is an interesting section of code which the virus does not use. This is a standard textbook routine which is used to display the contents of a specific register. This was probably used by the author to test the results of various routines and has simply not been removed.

The over-riding impression is one of a newcomer to computing, who has spent many hours building a “seat of the pants” program on a piecemeal basis, probably with a “Beginners’ Book of Programming” in one hand and a comforter in the other. However, the virus functions and has been found in the wild, facts which make its inclusion in anti-virus software essential.

A reliable search pattern for MACHOSOFT was published in *VB*, January 1991 and is repeated here:

```
  5051  56BE  5900  B926  0890  D1E9  8AE1  8AC1
```

AUTHENTICATION

Yisrael Radai

Virus Bulletin has not received permission to reproduce this article on CD from the author. Readers can obtain a paper copy of the original issue directly from *VB*.

STRAINS & FAMILIES

Fridrik Skulason

Overwriting Viruses

Most viruses attempt to preserve the functionality of infected programs. This article will take a look at those which do not, but instead overwrite any program which they infect.

Although viruses such as Jerusalem may occasionally infect a program in a way which causes damage to the host program, this is not intentional. The virus is designed to perform its function when an infected program is run and to finish by restoring the host program to its original state and transferring control to it. The purpose of this is obvious - by executing the original program, the virus attempts to make operation appear normal. A virus of which the user is unaware has a much better chance of spreading than one which causes alarm the first time an infected program is run.

Overwriting viruses do not attempt to hide their existence in any way - when an infected program is run and the virus has finished its task, it simply terminates. The failure of the infected program to perform as expected usually gives rise to alarm, and while the user may inadvertently infect more programs while trying to ascertain the cause of the problem, the destructive nature of overwriting viruses actually reduces their chances of spreading. While there have been occasional outbreaks of overwriting viruses, they have been isolated and easy to control. However this does not imply that they are harmless or should be ignored.

Because of the way in which overwriting viruses infect, it is impossible to recover infected files by using a disinfection program. Infected files must be deleted and restored from write-protected copies of the master software. This process, which is far more secure and reliable than using disinfection programs, is also more time-consuming.

At the moment five families of overwriting PC viruses are known. This is only a small percentage of the total number of viruses, which is understandable given the technical limitations inherent to this class. The majority of virus writers shun this method and seem to dismiss such crude tactics. The major reason overwriting viruses exist at all is due to their simplicity - it is far easier to write an overwriting virus than any other type.

Burger's Virus

The oldest overwriting virus was published in the infamous book by Ralf Burger - *Computer Viruses: A High Tech Disease.* It is 560 bytes long, although the actual virus code is shorter, only 540 or 541 bytes depending on which assembler is used to assemble the code. When an infected program is run, it will search for COM files to infect. If none are found, it will try to rename EXE files to COM and infect them. Should this fail, the virus will overwrite sectors with random garbage. The virus assumes that all programs start with a 3-byte jump instruction, and when it is finished it will jump to the target address of this instruction. If this address is within the first 560 bytes (which were overwritten by the virus) or if the program does not start with a JMP - for example if it is an EXE file, the system will probably crash when the program is run. The virus will infect COMMAND.COM, but the infected program will not work at all. In his book Burger writes:

> *"Experienced readers may note that this already short virus program (500 bytes) can be made shorter by removing remarks, extra segment calls/jumps, etc. This is perhaps a task for a long winter evening."*

It should not come as a surprise that several variants of this virus are known - all of which seem to have been created by typing in Burger's listing and modifying it. Most of these variants are 560 bytes long, but the length of the actual virus code varies. This is because the number of bytes written to infected files is stored as a constant in the program, instead of being determined by the offset of the last instruction.

The variants known to *VB* at the time of writing include:

Burger 560-A: This variant has been circulating around the virus research community as 'BURGER.COM'. It can be identified by looking at byte 7, which contains A2H, not A3H as in the original variant. This variant has been modified more than many others, in particular in the data area near the end.

Burger 560-B: This variant is also known as 'Burger-528', as the virus code is 528 bytes long, although the virus overwrites the first 560 bytes of infected files. It can be identified by looking at byte 35, which contains BEH. Some mistakes have been corrected in this variant, including removing an incorrect 'MOV AL,1' instruction.

Burger 560-C: The variant which is closest to the original virus appears to have been typed in verbatim from the book. The virus code is 541 bytes in length, so an alternative name for the variant is 'Burger-541', but it is also incorrectly known as 'Burger-509'. This variant can be distinguished from the others by looking at byte 142, which contains 90H.

Burger 560-D: Just like the 'B' variant, this one is 528 bytes long, although an uninitialised buffer at the end of the virus contains "garbage" instead of being filled with zero bytes. They can be differentiated by looking at byte 34, which in this variant contains BEH. This variant has been incorrectly reported as having a length of 496 bytes.

Burger 560-E: This variant is also known as 'Burger-537'. It is intermediate between the 'B' and 'C' variants, containing BEH at offset 35, but also containing the incorrect 'MOV AL,1' instruction at offset 46. This variant has been incorrectly reported as having a length of 505 bytes.

Burger 560-F: Practically identical to the 'C' variant, this appears to have been typed in from the book with only one minor change - the removal of the incorrect 'MOV AL,1' instruction at offset 46. This has shortened the virus by two bytes, making its code 539 bytes long.

Burger 560-G: This variant is also known as 'CIA' or 'Burger-542'. It is only slightly different from the variant in the book - some unnecessary segment overrides have been removed and a few other instructions have been modified slightly. It can be identified by looking at byte 241, which contains 90H, not B4H as in the 'C' variant.

All the variants above start with three NOP instructions (90H), which are used to identify a file as already infected. This is not true of the next variant, which starts with the bytes 96H, 00H. The code to check for an existing infection is also different, having changed from

```
CMP BX,9090H
```

to

```
CMP BX,9600H
```

However, the author of this variant made a mistake, as the byte order should have been reversed. The result is that the virus is not able to identify existing infections, and will overwrite files repeatedly. This variant was one of the first viruses reported and received a separate name, '405'. Considering its relationship to the other variants of the Burger family, it should rather be named 'Burger-405'.

The last known variant of the virus appeared in Taiwan. It has been partially rewritten and made significantly shorter - 382 bytes in length. The virus was first reported as '382 recovery virus', but 'Burger-382' is probably a more suitable name.

The 403 Virus

Unlike Burger's virus which does not remain resident in memory, this virus will hook INT 21H and remain resident using the standard INT 27H call. If the user later runs a COM file, the virus will intercept the call and overwrite the file.

Leprosy and Plague

The Leprosy virus overwrites the first 666 bytes of the files it infects. When an infected program is run the virus attempts to infect any COM or EXE file it finds, and then displays one of the two following messages:

```
Program too big to fit in memory
```

```
NEWS FLASH!! Your system has been infected with
the incurable decay of LEPROSY 1.00, a virus
invented by PCM2 in June of 1990. Good luck!
```

Part of the virus, including the text, is encrypted. The virus was originally reported in a program named 486COMP, which was uploaded to several BBSs in the California area.

Later a new variant appeared. It had been slightly modified, although the virus length remained unchanged, but the text message now went like this:

```
ATTENTION! Your computer has been afflicted with
the incurable decay that is the fate wrought by
Leprosy Strain B, a virus employing Cybernetic
Mutation Technology(tm) and invented by PCM2 08/
90.
```

The third variant in the family is slightly shorter, only 591 bytes, and named 'Plague' because of its text message:

```
Autopsy indicates the cause of death was THE
PLAGUE Dedicated to the dudes at SHHS
VIVE LE SHE-MAN!
```

Leprosy would not be a very interesting virus, were it not for the fact that it is written in C. This is not obvious from looking at the binary code, and was only revealed when the author released the source code and documentation.

Deicide

It is obvious that the Deicide virus was written by an inexperienced programmer, which is actually confirmed by a text message inside the virus:

```
This experimental virus was written by Glenn
Benton to see if I can make a virus while
learning machinecode for 2,5 months.
(C) 10-23-1990 by Glenn. I keep on going making
virusses.
```

When an infected program is run, it will attempt to infect another program in the current directory. If no uninfected COM file is found, the virus displays the following message and wipes out the first 80 sectors on drive C:

```
DEICIDE!
Glenn (666) says : BYE BYE HARDDISK!!
Next time be carufull with illegal stuff.
```

Minimal-45

This Bulgarian virus not only has the distinction of being the smallest overwriting virus - it is also the smallest virus known. When executed it will attempt to locate a COM file to infect and overwrite the first 45 bytes with the virus code. Its infection method is crude but functional.

Conclusion

It is probable that overwriting viruses will continue to appear for the foreseeable future, if only because of the widespread dissemination of Burger's book. However, the general trend is towards the development of virus code invisible both to the computer user and to diagnostic tools. The very fact that overwriting viruses destroy their host programs and reveal their presence means that the current examples are likely to become extinct, while the programming method in general will be surpassed by more subtle techniques.

PRODUCT REVIEW (1)

Mark Hamilton

IBM's VIRSCAN

This is the first time that *Virus Bulletin* has reviewed a virus scanner which is not available to users in the United Kingdom. As far as *VIRSCAN* is concerned, this has not always been the case, but since March - when IBM's *PS2 User Group* signed a non-exclusive marketing agreement with *S&S International* - its own product has only been available to computer users outside the United Kingdom.

Many major computer companies have opted to ignore the computer virus threat - IBM is not one of them. The company does not herald its achievements in the press or on public platforms, nor does it trumpet them in its advertising, but the assistance and cooperation it provides to the research community are exemplary. IBM has established a dedicated research unit called the *High Integrity Research Laboratory* at its *T. J. Watson* research facility in New York State which plays a leading role in the investigation into computer viruses and the development of techniques to combat the threat. Other lesser players would do well to follow IBM's lead.

I have decided to adhere to the guidelines and testing protocol published in last month's edition (*VB*, April 1991, pp. 5-8). This protocol makes possible an evaluation of *VIRSCAN* based on a comparison with the scanners tested in the April edition. All products were tested under identical conditions.

Documentation

So low profile is this product that all you receive is a disk in a disk-mailer. There is no printed documentation, no smart binder and no slip-case. The documentation for *VIRSCAN* is in the form of a text file on the distribution diskettes - both 3.5" and 5.25" were provided for review. When printed out, this comes to 38 sides of A4 of which 14 are devoted to a table of virus characteristics and a table of aliases.

IBM recommends that *VIRSCAN* is invoked only after the PC to be checked has been booted from a write-protected system disk and provides instruction on how such a disk should be created.

The documentation also explains the action (and interaction) of the scanner's 27 command line options. IBM provides a cross-reference table of aliases to the names it recognises. These are listed alphabetically by alias. It would have been useful to include a table listed alphabetically by IBM name in addition to this, with the appropriate variants listed against each one. However, the information which IBM provides is more than that supplied by most other manufacturers.

Operation

VIRSCAN uses a signature file in plain ASCII format (that is to say, it is typeable). To ensure that the signature file is correct, it includes a checksum that is compared against one calculated internally as the signature file is read in. If the two do not agree, the scanner will not proceed. This means that you cannot alter or add-to the main signature file (VIRSIG.LST). You may however create an additional signature file, called ADDENDA.LST, and incorporate new patterns therein. Patterns published in *Virus Bulletin* may be used. VIRSCAN offers two facilities not found in many scanners. Firstly, it recognises a pair of question marks contained within a signature as a "wild card". This means that if the signature contains the sequence: 61AB??CD, it will match 61AB7CCD, 61AB05CD etc., but not 61AB05CE.

Secondly, the scanner can search for virus 'mutants', by which IBM means that if a signature of a particular virus almost matches a sequence of bytes within a file, the file is assumed to be infected with a variant of that virus. If this option is enabled the incidence of false-positive identifications increases, so results should be viewed with caution. *VIRSCAN* can disable memory scans - conventional memory and upper memory can be disabled separately, as can DOS Boot Sector and Master Boot Sector scans. It can also undertake dedicated floppy drive scans. This option I welcomed as it meant that I could scan a pile of floppy disks with just one invocation of the program. I wish more developers of virus scanning software could incorporate such an option.

Scanning Speed

Unless instructed otherwise, *VIRSCAN* scans every byte of BIN, COM, EXE, INI, OV? and SYS files as well as boot sectors and the first megabyte of memory. The fastest sensible search mode is invoked by using the command line options to exclude scanning memory and searching for 'mutants'. In this 'Turbo' mode, *VIRSCAN* scanned a 20 megabyte hard drive containing 887 files (316 executables) in 3 minutes 16 seconds and a 360 kilobyte floppy disk containing 10 files (3 executables) in 51 seconds. The speed tests were run from the floppy drive against a clean hard disk and a clean test diskette. The results are roughly comparable to those of SCAN (*McAfee Associates*), SWEEP (*Sophos Ltd.*) and VISCAN (*Bates Associates*) but are far from being the fastest recorded (see *VB*, April 1991, p.15)

Detection Rate

VIRSCAN claims to be able to detect 224 viral strains and variants and its pattern file contains search strings for viruses which have only just been published in *Virus Bulletin*. These include Jeff and Super Hacker [1] (which IBM calls "Talentless Jerk" - an apt name for many virus writers). However, against the *VB* comparative review test-set of 306 parasitic and seven boot sector infections, *VIRSCAN* did not fare too well. In its

"Turbo'' mode, it reported infections in 235 files out of 306, while in its "Secure'' mode (i.e. searching for 'mutants' as well), 239 infected files were reported. However, these results are short of the 90%+ detection rates achieved by some of the other programs in last month's comparative review. This possibly reflects IBM's decision to exclude some of the more esoteric virus specimens from its scan data.

IBM's VIRSCAN

Product and Version	VIRSCAN 2.00.01	
Developer	IBM	
Issue Date	7th March 1991	
Availability	Worldwide (except UK)	
Number of Viruses in documentation	224	
Memory Checks		
- Conventional	Yes	
- Upper	Yes	
Network Aware	Yes	
Single File Check	Yes	
Definition Format	IBM. VB patterns may be added by the user.	
Virus Removal	None	
Access to VB Test-Set	No	
User Upgradeable Pattern Library	Yes	
Resident Scan/Monitor	No	
Scanning Speeds		
Test 1 (i) Hard Disk - 'Turbo'		3 min 16 sec
Test 1 (ii) Hard Disk - 'Secure'		4 min 3 sec
Test 2 (i) Floppy - 'Turbo'		51 sec
Test 2 (ii) Floppy - 'Secure'		1 min 9 secs
Scanner Accuracy		
Parasitic Viruses - 'Turbo'		235 out of 306
Parasitic Viruses - 'Secure'		239 out of 306
Boot Sector Viruses - 'Turbo'		7 out of 7
Boot Sector Viruses - 'Secure'		7 out of 7
Accuracy Percentages		
'Turbo' Mode		77.31%
'Secure' Mode		78.59%

For information about the entries in this table refer to the evaluation protocol published in *VB*, pp.6-7, April 1991.

Comments and Conclusions

VIRSCAN is unique in that IBM offers an "Enterprise-wide" licence. This means that for a single payment of US$42.00, the licensee may copy the program as often as he wishes for use within his organisation.

I am sure that if someone took the time and effort to enter the patterns published in *Virus Bulletin* for the viruses that *VIRSCAN* does not recognise, he would end up with a scanner which would rival those at the top in detection capabilities. It is a great pity that IBM has decided that UK-based users are to be denied this opportunity.

Given its detection rate of 78 percent, *VIRSCAN* should not be relied upon as a company's sole virus scanner. However it is *VIRSCAN's* user-updateable features and command line flexibility (which appears to be unparalleled) which make it such a useful and well-designed item of diagnostic software.

Test Conditions

For a full description of the test conditions, you should refer to page 8 of the April 1991 edition of *Virus Bulletin.*

The hard disk speed test was conducted on a Compaq Deskpro 386/16 that has two 42 Mb hard drives each of which is partitioned into two 21 Mb partitions. The test drive contains 887 files (of which 316 were .COM or .EXE executables) occupying 20.5 Mb. The floppy test was conducted using a 360 Kb 5.25 inch floppy disk (Microsoft C V5.1 Setup Disk) which contains 10 files (of which three are executable). No virus infected files were present during speed tests.

The virus test suite comprises 306 parasitic infections using the following viruses: 1049, 1067, 1260, 1600, Eddie-2, 2144, 2480, 405, 417, 440, 492, 4K, 5120, 516, 600, 696, 707, 717, 800, 8 Tunes, 905, 948, Agiplan, Aids, Aids II, Alabama, Ambulance, Amoeba, Amstrad, Anthrax, Anti-Pascal, Armagedon, Attention, Bebe, Best Wishes, Blood, Black Monday, Burger, Cancer, Carioca, Cascade, Casper, Christmas in Japan, Christmas Tree, Cookie, Dark Avenger, Datacrime, Datalock, Dbase, DBF Blank, December 24th, Destructor, Diamond, V2000, Dir, Diskjeb, Dot Killer, Durban, Dyslexia, Eddie, Evil, Fellowship, Fish-6, Flash, Flip, Fu Manchu, Ghostballs, Jerusalem variants, Guppy, Hallochen, Hymn, Icelandic, Internal, Itavir, Jocker, Joker-1, July 13, Kamikaze, Kemerovo, Kennedy, Keypress, Lehigh, Leprosy, Liberty, Love Child, Lozinsky, Machosoft, MG, MG 3, MGTU, Mix1, Mix1-2, MLTI, Monxla, Murphy, Nina, Nomenklatura, Nothing, Number of the Beast variants A to F, Ontario, Oropax, Parity, Perfume, Phoenix, Piter, Pixel variants, Plastique, Polimer, Polish 217, Proud, Prudents, Rat, Russian Mirror, Saddam, Scott's Valley, Shake, Slow, South African, Spanish, Spanish Telecom, Spyer, Stupid, Subliminal, Sunday, Suomi, Suriv variants, SVC, Sverdlov, Svir, Sylvia, Syslock, Taiwan, Tenbyte, Terror, Tiny, Tiny Family 1 variants, Tiny Family 2 variants, Traceback, TUQ, Turbo 488, Turbo Kukac, Typo, V-1, V2000, V2P2, V2P6, Vacsina variants, Vcomm, VFSI, Victor, Vienna variants, Violator, Virdem variants, Virus 90, Virus B, Virus 101, Voronezh, VP, W13-A, W13-B, Westwood, Whale, Wisconsin, XA-1, XA- 2, Yankee variants, Old Yankee variants, and, Zero Bug.

The following Boot Sector viruses were also used: Aircop, Brain, Disk Killer, Italian, Joshi, Korea, and, New Zealand 2.

[1] *VB* originally reported the Super Hacker virus as 1077.

PRODUCT REVIEW (2)

Dr. Keith Jackson

VET Anti-Viral Software

VET is the first Australian anti-virus program which *VB* has reviewed.

The documentation provided with *VET* comes in the form of a 72 page A5 booklet. Its stated aims are to provide a better understanding of how PCs work, how viruses attack, how they can be removed, how to use *VET* effectively and how to minimise the risk of damage caused by a virus. The level of detail included by the author can be gauged from the fact that the chapter entitled 'Background information' contains sections which provide very low level descriptions of PCs, disk storage, the MS-DOS operating system, interrupts, terminate and stay resident (TSR) programs and how to format various types of disk. The parts which describe the various stages of booting an MS-DOS computer are very clearly written, among the best I've seen. Viruses are described in general terms, leading up to the latter part of the booklet, where individual viruses are described in detail.

The author of *VET* is a lecturer in Australia, and his communicative skills shine through. The content of the documentation is generally excellent, and it is obvious that he knows what he is talking about. Having said this, the production quality of the documentation leaves much to be desired. Although glossy verbiage is avoided, it is somewhat difficult to locate specific sections. This problem would be eased by including an index (how often have I complained about that in *VB* product reviews!) and organising the contents in a more logical manner. For instance, describing the installation procedure in a section tucked away on page 36 is a mite perverse; it should be at the beginning.

Software Components

Software included in *VET* checks memory for viruses, scans boot sector(s) and files for known viruses, removes viruses, can recognise and/or replace infected boot sectors, monitors MS-DOS for 'illegal' function calls and calculates checksums for any file. These facilities are provided by a suite of programs rather than in one all-encompassing package.

Scanning Speed

I used *VET* to scan the hard disk on my Toshiba portable (see Technical Details below). In its default mode *VET* reported that it had scanned 290 files out of a possible 1297, which took it only 46 seconds. Impressive. At this speed, *VET* is scanning only executable files, however most scanners have similar constraints (certainly both *SWEEP* and *SCAN*, see below, can act in this fashion). When *VET* was instructed to scan all files on the hard disk, it inspected 1297 files in 2 minutes 25 seconds. This is still fast enough to put most scanning programs in the shade. For comparison purposes, *SWEEP* (v2.24) from *Sophos Ltd*. took 4 minutes 54 seconds to search the same hard disk, while looking for 317 virus patterns, and 11 virus identities. *SCAN* (4.5v66) from *McAfee Associates* took 4 minutes and 20 seconds to scan the same hard disk. 79 subdirectories, 1296 files in total, were scanned.

Last month's *VB* contained a comparison of 12 virus scanning programs, and as no absolute measure of scanning speed was included (measured in Kbytes scanned per minute - or some such similar unit), the only way to formulate a reasonable comparison of how quickly *VET* can scan for viruses is to estimate the relative search speeds of other scanning programs that were included in the comparative review. Using the figures quoted above for *SCAN* and *SWEEP*, and estimating a proportionality factor between last month's results and my testing, in scanning speed tests *VET* would have come out top (or thereabouts) of all the packages tested.

Installation and Self-Check Routines

During installation, *VET* checks its own floppy disk to ensure that no corruption has taken place. During this process, the entire contents of the disk are checked, so if the *VET* files have been copied across to another floppy disk for testing purposes (as I did!), then *VET* installation will terminate very rapidly. A direct image of the *VET* disk must be taken using the MS-DOS DISKCOPY command to ensure that the master disk is intact. Even though I have written much in the past decrying copy protected software, I have no objection to checks such as this. Backup copies can still be made (albeit only by using DISKCOPY), and the user is never left in the position of having only one usable copy of the *VET* files. The installation program also requests the user's name and the type of computer on which it is to be used and these are displayed every time that *VET* is used. I view such tactics as a reasonable way to deter software pirates. Indeed as the boot sector may vary from one computer to the next, they are almost essential for *VET's* boot sector restoration facilities.

CRC Verification

A program is included with *VET* (VERIDISK) which calculates CRCs (Cyclic Redundancy Checks) for all parts of a disk and then combines these into a single checksum which can be tested to see whether anything on the disk has changed. The combination process is not explained in the *VET* documentation but the documentation does describe another program available from the authors of *VET* which calculates the same checksum and then hides the result at the end of the boot sector of the disk. VERIDISK can test that the disk has not been changed in any way. This is presumably how the aforementioned *VET* installation program knew that I had copied the files from the original disk.

One of the README files provided with *VET* states that the version of *VET* used for this *VB* review was prepared under extreme pressure to meet the deadline set by the Michaelangelo virus (which triggers on March 6th), and warns that it may well have bugs if a user decides to 'try doing anything adventurous' (their phrase). I dislike software being rushed out in this fashion; it should be properly tested or kept in the cupboard.

Scanner Accuracy

I tested *VET's* scanning accuracy using the *VB* set of test viruses listed in Technical Details below. The version of *VET* used for this review only claims to detect 37 unique viruses, but the results I obtained are even worse than this figure. For the first time in one of my *VB* reviews, it is easier to name the viruses that *VET did* detect, rather than the ones that it failed to detect. Out of the 183 virus samples used, *VET* only detected 18 instances of virus infection; 4K, Brain, Cascade(2), Dark Avenger, Fellowship, Flash, Italian, Prudents, Slow, Sunday(2), Taiwan(2), Yankee(4). Amazingly, only four viruses were detected in more than a single test sample; these were the Cascade, Sunday, Taiwan and Yankee. Curiously, (and I think uniquely), *VET* claims to detect more boot sector viruses (20 in all) than file viruses (17), and using more boot sector viruses in the test-set would no doubt improve the detection rate. The fact remains that even if *VET* detected all of the viruses to which it lays claim, this would still be far short of what competitor packages are currently achieving.

The documentation says that the viruses known to *VET* include all those 'on the loose' in Australia. I doubt that viruses respect international boundaries, and I know that many more viruses are extant in other parts of the world.

Memory-Resident Monitor

VETSTOP is a TSR program provided in the *VET* package that watches out for 'illegal' DOS calls while virus-infected programs are executing. However the list of viruses known to VETSTOP is even smaller than the list of viruses known to the scanning part of the *VET* package - just 13 in total. Given this major constraint, I'm unsure how useful VETSTOP really is. The documentation states that it is useful in schools, where one particular virus may be known to be prevalent. However, if a particular virus does not feature in its very short list of known viruses then VETSTOP's usefulness is questionable.

Conclusions

The scanning part of *VET* is very quick, but this is partially accounted for by the very small number of viruses for which it searches. In last month's comparative review of scanner programs, the lowest detection rate was 36% of all test samples used. Even using the smaller *VB* review test-set of 114 unique viruses, *VET's* scanning accuracy is much worse than this (approximately 10%), and as more esoteric viruses are included (which have purposely been *excluded* from the standard review test-set), the detection rate would inevitably get much worse. With such a low percentage result for scanning accuracy, I cannot in all honesty recommend *VET* as a useful scanning program, and as the program provided to trap 'illegal' DOS calls has an even shorter list of viruses of which it has knowledge, neither can this be recommended except in this most limited of circumstances.

I am saddened by these results. The author of *VET* obviously has much to offer in terms of super-fast scanning software, and a natty line in well written, very readable, documentation, but without extending the list of known viruses quite extensively, *VET* will not come close to competitive packages.

Technical Details

Product: *VET*

Developer/Vendor: *CYBEC Pty.Ltd.*, PO Box 82, Hampton, Vic.3188, Australia, Tel: Not stated.

Availability: IBM PC/XT/AT, PS/2, or compatible running MS-DOS.

Version Evaluated: 6.4

Serial Number: None visible

Price: Ranges from 90 Australian dollars for a single PC, down to 9 Australian dollars per PC when used in schools. Quantity discounts are available.

Hardware Used: A Toshiba 3100SX laptop portable with a 16 MHz 80386SX processor, one 3.5 inch (1.44M) floppy disk drive, and a 40 Mbyte hard disk, running under MS-DOS v4.01. Also an ITT XTRA with a 4.77 MHz 8088 processor, one 3.5 inch (720K) floppy disk drive, two 5.25 inch floppy disk drives, and a 32Mbyte Western Digital hardcard, running under MS-DOS v3.30.

Virus Test-Set: This suite of 114 unique viruses (according to the virus naming convention employed by *VB*), spread across 183 individual virus samples, is the standard *VB* test-set. It comprises two boot sector viruses (Brain and Italian), and 112 parasitic viruses. There is more than one example of many of the viruses, ranging up to 12 different variants in the case of the Tiny virus. The viruses used for testing are listed below. Where more than one variant of a virus is available, the number of examples of each virus is shown in brackets.

1049, 1260, 12 TRICKS, 1600, 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 800, 8 TUNES, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), Anti- Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Cascade (2), Casper, Dark Avenger, Datacrime, Datacrime II (2), December 24th, Destructor, Diamond (2), Dir, Diskjeb, Dot Killer, Durban, Eddie 2, Fellowship, Fish 6 (2), Flash, Flip (2), Fu Manchu (2), Hymn (2), Icelandic (3), Internal, Itavir, Jerusalem (2), Jocker, Jo-Jo, July 13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (2), LoveChild, Lozinsky, MIX1 (2), MLTI, Monxla, Murphy (2), Nina, Number of the Beast (5), Oropax, Parity, Perfume, Piter, Polish 217, Pretoria, Prudents, Rat, Shake, Slow, Subliminal, Sunday (2), Suomi, Suriv 1.01, Suriv 2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Taiwan (2), Terror, Tiny (12), Traceback (2), TUQ, Turbo 488, Typo, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Whale, Yankee (7), Zero Bug.

END-NOTES & NEWS

Correction

In the April 1991 edition of *VB*, it was incorrectly stated that *VISCAN* from *Bates Associates* was not available on permanently write-protected diskette. The software **is** distributed in this form by UK distributor *Total Control*.

Virus Bulletin Conference, Hotel de France, Jersey, September 12-13th 1991. For information, registration, flight and hotel information etc., contact Petra Duffield, *VB Conference*, UK. Tel 0235 531889. Please note that this number differs from that published below for the *Virus Bulletin* office.

The *National Computer Security Association* in the United States has released the second edition of its very comprehensive report entitled ***Virus Scanners: An Evaluation***. The 117 page report features 15 prominent virus scanners and describes each program's features and performance against a virus test-set comprising 921 infected files. Interesting to note that scanner accuracy and speed results reported by the *NCSA* are strikingly similar to those reported by *VB* in last month’s comparative review, despite the fact that the test-set was dissimilar. The report costs US $78.00 and is available from *NCSA*, Suite 309, 4401-A Connecticut Avenue NW, Washington DC 20008, USA. Tel 202 364 8252.

Central Point Software (developer of *PC Tools*) has released ***Central Point Anti-Virus***, a menu driven virus ‘prevention, detection and removal’ package which includes a TSR monitor called VSAFE, a virus-specific monitor called VWATCH, and various disinfection and ‘immunisation’ facilities. Registration includes access to a “24 hour Virus Hotline”, on-line support and quarterly updates. The standard package costs £115 and virus protection updates will be available for £19.50. UK distributors for the package are *Softsel* (Tel 081 568 8866) and *P & P* (Tel 0706 217744).

The “**Total Office Anti-Virus Solution**” is being offered by UK company *SA Software*. The company is offering a combined package comprising its *PC Immunise II* checksumming program along with *Dr. Solomon’s Anti-Virus Toolkit*. Tel 081 998 2351.

S&S UK is holding a **Seminar on Data Recovery** (May 8-9th 1991) and a **Virus Seminar for Managers** (May 29th 1991). Tel 0442 877877.

Sophos Ltd. is holding a seminar on **Anti-Virus Strategy for Software Producers**, London, 17th July 1991. Seminar addresses sterile program development and testing, disk duplication, risks from programming subcontracting etc. Tel 0235 559933.

BOOT Plus (*Commcrypt, Inc.*,10000 Virginia Manor Road, Suite 300, Beltsville, MD 20705, USA) is a **program to facilitate copying the Master Boot Sector** thus easing recovery in the event of a boot sector virus infection. The program is designed to be quicker and simpler to use for this purpose than disk utilities such as *Norton* or *PC Tools*.

VIRUS BULLETIN

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

USA (first class airmail) US$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, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.