May 1993 ISSN 0956-9979

VIRUS BULLETIN

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

CONTENTS

EDITORIAL

Better DOS than DOS?

It has been a long wait. The tension has been built dramatically by the marketing men, hyping users into a state of breathless expectation... and now the moment is here. How can you have a new computer for just £49 asks *Microsoft*? Easy - install *MS-DOS 6* on your old machine! This operating system, users are told, is the one for *you*. Why be stuck with anything else?

MS-DOS has come a long way since its birth in August 1981 - as has its creator, Bill Gates (who was born even earlier). The first version of the operating system consisted of 4000 lines of code and ran in 8 Kbytes of memory.

This was little over a decade ago, yet we now live with a 7 Mbyte monster of an operating system... which will run most of the software designed for its great-great-great-grandfather. In essence, although *MS-DOS* has been tidied up somewhat from its humble beginnings, it has not changed much.

MS-DOS 6 represents the next salvo in the increasingly bitter operating system war now being waged between *IBM* and *Microsoft*. In this release, *Microsoft* has upped the stakes by making the system much less *OS/2*-friendly. When DOS 6 was installed on one of the *VB* test machines, which has an *OS/2* partition (or rather, *had* an *OS/2* partition) the net result was to make *OS/2* inaccessible. True, the product warns the user to consult the manual before proceeding, but this 'hiccup' during installation could be seen as a blatant attack at its nearest competitor. No doubt it occurred for purely technical reasons.

For a long time, third parties supplied the parts of the operating system which *Microsoft* in its infinite wisdom had decided users did not need or want. With each release of *MS-DOS, Microsoft* examined which add-ons users were prepared to pay for and built them in to the operating system. After all, what easier means of market research than to let someone else do it for you.

Between the release of DOS 5 and DOS 6, *Microsoft* has clearly been examining which extra utilities are selling well. The three latest bolt-on goodies are disk compression, backup facilities and anti-virus software (the current boomtime industry).

It is not difficult to see why adding disk compression software and decent backup facilities is a good thing - and a good selling point. After all, the operating system is the sensible place for both.

However, the question of whether there should be any anti-virus utilities included in the operating system is much more thorny.

The first point is that for the ordinary computer user with no anti-virus software, *MS-DOS* 6 is a very good buy indeed. It will stop the Form virus, it is going to prevent the spread of Cascade - indeed, it is entirely possible that it will cause a gradual decrease in the numbers of common viruses which are in the wild.

It is all too easy for computer security practitioners to see no further than the walls of their ivory tower, and forget what is going on in the real world. The vast majority of users have absolutely no knowledge of computer viruses and *never ever* take a backup. All the arguments about good security practices and update frequency suddenly become redundant. Some protection is better than none.

Of course there are problems with the anti-virus components of *MS-DOS* - it would be foolish to say this was not the case: it will be targeted, and it is not particularly difficult to subvert. Of course there are worries about the age and frequency of the updates, but for the single user, it still represents at least some protection against viruses.

Companies may have a completely different view of built-in virus counter-measures. One would be ill-advised to base the security of any important system on the reliability of *MSAV*. Indeed, the review in this month's *Virus Bulletin* brings to light one of *MSAV's* biggest weaknesses: its age. The product performed worse than the version of *CPAV* examined in the January 1993 edition of *VB* - hardly a result to inspire customer confidence.

Apart from worries about its rather antique virus recognition capability, users have to be aware that *any* anti-virus features built in to the operating system *will* be specifically targeted. Would you trust an integrity checker of which you knew every virus writer owned a copy?

It is impossible to upgrade the *MS-DOS* operating system and suddenly be immune to all forms of computer virus. Even though it is possible to make reasonably good generic virus detection software, the very nature of *MS-DOS* makes this approach inherently insecure. By effectively handing out copies of the door locks to all would-be burglars, one is weakening the security of one's house.

The situation may improve with the arrival of the next generation operating systems. They are unlikely to be much more secure, but it is probable that it will be significantly harder to write viruses exploiting their weaknesses. Come to that, they will be much more difficult to write *anything* for them. Still, that is a very small price to pay. Isn't it?

NEWS

Popping Up

Dr Joseph Popp has been found guilty of 'attempted extorsion' by a court in Rome. Dr Popp, who was accused of being the man behind the infamous Aids diskette case, has been sentenced to a two and a half years imprisonment.

Popp was tried *in absentia*, and therefore has sixty days in which to lodge an appeal. If no such appeal is received within this time, the sentence is fixed.

Popp was originally extradited to the UK to face charges concerning this case, but was eventually ruled unfit to plead and was sent back to the USA. This is the first successful case brought against him in respect of the incident.

Jim Bates, who was involved in the attempt by the British authorities to bring Dr Popp to a UK court, said 'I am delighted that the hard work put in by both myself and the police has finally resulted in a conviction'. Sources have informed *Virus Bulletin* that part of the evidence used against Popp in the trial was an Italian translation of Bates' report on the Aids disk.

This sentence means that the authorities will immediately be looking for Popp. For a long period after the original Aids Diskette incident, his whereabouts were not known.

However, the Italian judiciary need not look far, because according to a report in an American magazine, Dr Popp is alive and well and living in Lake Jackson, Texas, where he is writing a book. Popp has declined to disclose any details of his forthcoming *œuvre* until he is ready to have it published, but it is believed to be non-fiction, written for a general audience, along the lines of 'I'm OK, You're OK'. With approximately fifty untried cases against Popp still to be heard, this seems a most unlikely statement❑

Virus Prevalence Table - March 1993

Viruses reported to *VB* during March 1993.

Virus	Incidents	(%) Reports
Form	22	33.3%
New Zealand 2	8	12.1%
Spanish Telecom	6	9.1%
Tequila	5	7.6%
Cascade	4	6.1%
Joshi	4	6.1%
BFD-451	3	4.5%
Helloween	3	4.5%
Yankee	3	4.5%
1575	2	3.0%
Michelangelo	2	3.0%
Vacsina	2	3.0%
DIR-II	1	1.5%
NoInt	1	1.5%
Total	66	100.0%

IN MEMORIAM
David Lindsay

Virus Bulletin is saddened to record the death of David Lindsay on 8th April. Lindsay, a member of this journal's editorial board since the very first edition, had been a source of help and advice for *VB* over the years.

Lindsay joined *Digital Equipment Company Ltd* in 1985 as its UK security manager, bringing with him a wealth of knowledge and experience. At this time, Lindsay was already active in several UK computer security committees. Unlike many others in the IT world, he excelled in (and even enjoyed) the intricacies and details of policies, standards and procedures.

In 1989 Lindsay transferred to a position in which his activities were focused on establishing and re-shaping *Digital's* european security policies. In this capacity, Lindsay joined the newly established European Security Programme Office.

Among his achievements while at *Digital* were his involvement in the *1990 Computer Misuse Act*, and his active membership of several computer security groups including the *BCS Security Committee* and the *IFIP/ Security Organising Committee*.

Lindsay left *Digital* in 1992 but continued to maintain close association with his colleagues and friends there. He was well respected and liked by all who had the pleasure of working with him, and will be remembered for his unselfish and above all fair contributions to computer security in the UK.

Virus Bulletin wishes to extend its sincere condolences to his widow, Celia, and his daughter, Fiona.

McAfee Makes a Deal?

Those watching the *McAfee* v. *Imageline* case will be interested to learn that the case did not come to trial as expected last month.

This indicates that the two companies reached some sort of settlement outside court. However, the exact details of the settlement have not been made public, and both sides of the case have stated that they have 'No comment' to make.

However, sources close to the case have speculated that the settlement may well be 'substantial'. This follows last month's news that *Imageline* had settled its case with *Parsons Technology*.

The exact terms of the settlement made with *Parsons* are not publicly available, but it is rumoured that no money was involved❑

Dirty Macs

Two new viruses written for the *Apple Macintosh* have been found. The first virus, named INIT-17, affects all *Macintosh* computers under both System 6 and System 7.

The infection is accomplished by altering existing program code, but due to the bugs in the routine and the way the virus alters system files, it may cause damage in some instances.

The trigger routine displays an alert message in a window entitled 'From the depths of Cyberspace' the first time a machine is rebooted after 6:06:06 pm, 31st October '93.

The second virus is named INIT-M. It is malicious and may cause severe damage to those systems affected.

The virus is only active under System 7. It replicates when application files are run and is likely to spread extensively. The infection is accomplished by altering existing program code.

Extensive damage to systems occurs on Friday 13th. Files and folders are renamed to random text strings, creation and modification dates are changed, and file creator and type information are scrambled. After the trigger routine has executed, recovery is extremely difficult.

When present on an infected system, the virus may interfere with the proper display of some application window operations. It will create a file named 'FSV Prefs' in the Preferences folder.

Most anti-virus packages for the *Macintosh* have been updated to detect these new viruses, and users are advised to obtain new copies as soon as possible❑

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 24th April 1993. Each entry consists of the virus' name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or preferably a dedicated scanner which contains a user-updatable pattern library.

Type Codes

C = Infects COM files **E** = Infects EXE files **D** = Infects DOS Boot Sector (logical sector 0 on disk)

M = Infects Master Boot Sector (Track 0, Head 0, Sector 1) **N** = Not memory-resident

R = Memory-resident after infection **P** = Companion virus **L** = Link virus

Known Viruses

10 past 3.B - CR: A 789 byte variant, which is detected by the 10 past 3 pattern.

ARCV.Lurve - CR: A 718 byte virus.

```
ARCV.Lurve      74F6 8836 AE03 E8D9 FFB4 40B9 CE02 BA05 01E8 C3FF E8CB FFC3
```

Civil War - CN: First we had Dark Avenger, then Dark Angel and now Dark Helmet... or so claims the text string inside this virus which reads 'Civil War, (c) 1992 Dark Helmet'. Otherwise, this is an unremarkable 244 byte virus.

```
Civil War       80E1 2F80 F901 5974 4A51 523E 8B9E F001 B43F B903 008D 96EB
```

CV4, Comvirus 1.0 - CN: A simple, 321 byte virus. Infected programs will display the text 'This file infected with COMVIRUS 1.0'.

```
Comvirus        C746 FBFA FAB8 0042 33C9 8BD1 CD21 B440 B905 008D 56F8 CD21
```

Danish Tiny.Wild Thing - CN: This 289 byte variant contains a text message claiming the author is 'Admiral Bailey', a member of the YAM group.

```
Wild Thing      8BD7 B902 00B4 3FCD 2181 3D07 0874 43B8 0242 33C9 33D2 CD21
```

Dreamer - CR: This 4808 byte virus has been distributed in certain quarters under the name 'Hitler', and it includes the text 'Hitler Virus by Dreamer/DY.

```
Dreamer         9C80 FC11 74B3 80FC 1274 AE3D AB42 7505 9DF8 CA02 003D 004B
```

Dutch Tiny.122 - ER: An unusual virus. It infects only EXE files, but they are infected as COM files, by overwriting the first 3 bytes with a JMP and appending the virus code. Obviously, infected programs will not work properly. A 124 byte variant with the same behaviour exists, which can be detected with the same search string.

```
Dutch Tiny.122  5253 501E 3D00 4B75 35B8 023D E8E7 FF72 2D93 0E1F B43F CD21
```

Frajer - CN: A 649 byte virus, awaiting analysis. 'Frajer' can be loosely translated from Croation slang as 'Cad'.

```
Frajer          BA00 0103 D6B4 40CD 2133 C933 D232 C0B4 42CD 21B9 0500 BA6C
```

Fumble.D - CR: A 867 byte variant of this virus, which was previously called Typo. Detected with the Fumble (Typo) pattern.

Hitchcock.1238 - CR: Similar to the variant reported earlier, but 9 bytes shorter. It is not certain whether this is an older or younger variant, but it was probably written by the same author as the original, it plays the same tune and is detected with the same pattern.

Hoa - CER: A 950 byte virus containing the following text in encrypted form: 'This playgame was writen by Nguyen The Quang, Nacentra Co...101 -Hai Ba Trung, St.1 - HoChiMinh City, Phone: 96282 .... Press any key to Continue!'. Awaiting full analysis.

```
Hoa             80FC CE75 04B8 0821 CFFC 5506 BD23 00FA E810 02FB 2EC6 0622
```

Intrep - CER: A 946 byte virus, awaiting analysis.

```
Intrep          578B FA8B 055F 3B85 7201 7402 F8C3 F9C3 E843 0072 03E9 90FD
```

Jerusalem.Glory - CER: A 1831 byte variant, which seems to have been modified significantly - perhaps in order to avoid detection.

```
Glory           7FF2 AE26 3805 E0F9 8BD7 83C2 03BB 6602 061F 0E07 B800 4B9C
```

July 13th.1199 - EN: This virus is two bytes shorter than the original, but otherwise very similar.

```
July 13.1199    2EA0 1200 3490 BE12 00B9 AF04 2E30 0446 E2FA
```

Liquid - CN: A 599 byte virus, which only works on '286 machines and above.

```
Liquid          8BD5 81C2 0102 E83F FFB8 0040 CD21 72DC BA00 00B9 0000 B802
```

Marauder.860.B - CER: This is a minor variant of the Marauder.860 virus, with the differences not visible unless the virus is decrypted. This variant is detected in the same way as the original virus.

PrintMonster - CR: A 853 byte virus, containing the string 'PrintMonster30', which interferes with printer operation.

```
PrintMonster    9C80 FC1A 7415 80FC 0075 0A3C 2072 063C 7B73 0204 022E FF1E
```

Russian Tiny - CR: A large number of small viruses of East-European (probably Russian) origin has been reported recently. Due to classification and naming problems it has been decided to move them all to the 'Russian-Tiny' pseudo-family, with any groups that can be identified classified as sub-families, as follows. The original 'Russian Tiny' is now Russian Tiny.A.131, 'CC' is Russian Tiny.B.145, a new C sub-family contains 145, 146,150 and 157 byte variants, the D sub-family contains 129, 130 and 132 byte variants (The pattern for Russian Tiny.D is found in all three variants) and finally, the 127 and 143 byte viruses are in the E and F sub-families.

```
RussTiny.C.146  80FC 4B75 5B3C CC75 0558 57A5 A5CF 5053 521E B802 3DCD 2172
RussTiny.C.150  80FC 4B75 5D3C CC75 0558 57A5 A5CF 5053 521E B802 3DCD 2172
RussTiny.D      5080 F44B 7542 5352 1EB8 023D CD21 7235 930E 1FB4 3F99 B904
RussTiny.F.143  5053 521E 80EC 4B75 47B8 023D CD21 7240 93B9 0400 0E1F 33D2
```

Shaman - CN: The name of this simple 251 byte virus is derived from the text 'DemoVirus v1.0 Copyright (c) 20.8.1991 by Shaman'.

```
Shaman          8B04 A3C5 01B4 40BA 0001 B9FB 00CD 2172 185A B440 8B0C CD21
```

Simple 1992 - CR: This 424 byte virus actually includes the text 'SIMPLE 1992 (c)', and (big surprise) it is a rather simple virus, probably written in 1992.

```
Simple 1992     BA00 EACD 218B D8B4 40BA 0001 2E8B 0E03 01CD 21B4 3ECD 2172
```

Sinep - CR: A 644 byte Russian virus, awaiting analysis.

```
Sinep           FCFA F32E A4FB 80FC 4B74 3880 FC4C 7409 80FC 3174 040A E475
```

Star One - CN: A simple, 222 byte virus. Two improved variants of it are known: Cybertech A (1076 bytes) and Cybertech B (1215 bytes). Both these variants are encrypted, and able to infect EXE files as well.

```
Star One        2D03 002E 8986 D600 B440 8D56 04B9 DE00 CD21 B800 42E8 DBFF
CyberTech A     E800 005D 83ED 0750 8DB6 1B00 89F7 B91D 04AC 34?? AAE2 FA
CyberTech B     E800 005D 83ED 0750 8DB6 1B00 89F7 B9A8 04AC 34?? AAE2 FA
```

SVC 5.0 - CER: Similar to the B variant, it has the same size and is detected with the same pattern.

Timid - CN: Two new members of the Timid family have been found, with infective lengths of 513 and 526 bytes. They are detected with the Timid.306 patterns. Both variants contain bugs, and infected programs may crash.

Trivial.44.B - CN: Yet another simple, overwriting virus.

```
Trivial 44B     023D BA9E 00CD 21B9 2C00 8D16 0001 B440 CD21 B43E CD21 B44F
```

Uruk-Hai - CR: A family of several viruses, 300, 361 and 394 bytes long. The viruses are probably of Russian origin.

```
Uruk-Hai.300    6050 3D00 4B75 65B8 0043 CD7B 80E1 3EB8 0143 CD7B B802 3DCD
Uruk-Hai.361    6050 3D00 4B75 62B8 0043 CD7B 80E1 3EB8 0143 CD7B B802 3DCD
Uruk-Hai.394    5052 5351 1E3D 004B 7503 E836 001F 595B 5A58 EBE7 B003 CFBB
```

VCL.481 - CEN: An encrypted, overwriting, 481 byte virus, which should be detectable by any program that can detect VCL-generated decryption loops.

VCL.Dome - EN: A 546 byte, overwriting virus. Infected programs may display the text 'Divide Overflow'.

```
VCL.Dome        B41A 8D56 80CD 21B4 4EB9 1000 BA34 02CD 2172 2780 7E95 1075
```

Youth.Hannibal - CR: This variant is closely related to the Futhark variant, but it contains the text '(c) Hannibal Lechter'.

```
Hannibal        80FC 1274 BB80 FC4E 74B9 80FC 4F74 B42E 803E 8301 0074 03E9
```

Zaphod - CN: A 399 bytes virus which does not seem to do anything remarkable. Awaiting full analysis.

```
Zaphod          03F0 B905 008A 253A 2475 0746 47E2 F6EB 7290 5EB8 0042 8B9C
```

INSIGHT

Bates: Blues and Roots

One of the joys of being a virus researcher is that one is not necessarily tied to an office in the centre of town. Jim Bates, the author of the *VIS Anti-virus Utilities* is living proof of this. 'It's easy to find me', Bates had explained earlier, 'go over the first cattle grid, and we're just behind the old Hall'.

At first glance it is difficult to believe that a high-tech industry is run from these rather bucolic surroundings. However, Bates is a well known figure in the anti-virus community and, as a sax-toting, plain-speaking, jazz-playing researcher, is a colourful character.

Getting In

Like so many others in the industry, Bates became involved right at the beginning of the virus problem. 'A chap in Leicester sent me a copy of Brain and I decided to take it apart. It took me three days to do it. I wrote a report about it and sent it to a few magazines. Next, I received a copy of Italian, and then Jerusalem. Each time I wrote a short report. As I did so, more and more people started to send me viruses. It started off as a spare-time occupation.'

Bates did not immediately produce a commercial package. 'I feel uncomfortable about charging for anti-virus software. It's a bit like seeing somebody who is drowning in the canal, and asking for a tenner before you save them. The first thing I wrote was a simple scanner for pattern recognition. The next thing I wrote was called *SCAN-X*. It was an unusual product, because it was designed to work in an infected environment. After a few weeks, I got a call from a journalist who wanted to include it in a review. To my surprise it came top in terms of speed and accuracy, and I started to get a lot of calls asking for copies of it.'

'I started up the *Virus Information Service* (*VIS*) - the idea was people paid a subscription and I kept them up to date about particular viruses and software to detect them. From there the whole thing just snowballed. I still get calls that *SCAN-X* has reported an error or detected a virus - and it must be three or four years old by now!'

Trust Me, I'm an 'Expert'

Bates believes that the lack of independence within the industry is a serious problem. 'When I was setting up *VIS*, I asked around various government agencies and companies to see if they would fund some independent research into computer viruses. I still feel that this industry badly needs some genuine independent input - something which does not involve anyone with a commercial interest in the virus problem. While there are a lot of very good researchers who keep their commercial interest to one side, there are others who don't, and as far as the users are concerned, we all get tarred with the same brush.'

This 'self-interest', Bates believes, is largely responsible for the lack of education of computer users. 'When an anti-virus vendor puts out an alert about a particular virus, it's a fairly common reaction for the user to say "Well, don't you write the viruses?" or "Of course, you want to panic people." It's very difficult to put out an alert, if people know that you are selling an anti-virus product.'

Reviewing Reviews...

One of the biggest problems with anti-virus software is that users have no way of reviewing it themselves - they can only trust reviews published by others. 'If I was ever in a situation where I didn't need to sell anti-virus software, I'd like to set up an independent anti-virus software review centre' says Bates. 'There is some very good software out there, but there is also some incredibly bad software. If I was to review software, I would get the reply, "Well you're bound to say that, it's competing against your product".'

'The problem is, in order to review anti-virus software, you can't just look at it and say "This is a pretty interface, and it does this, this, this and this." The only way you can check whether anti-virus software is any good is to run in against actual virus infection conditions. Throw away the library of God-knows-how-many-thousand viruses everyone claims to have. Bring in a range of viruses chosen because of their different capabilities. Each one of those could be introduced to a machine under a range of different conditions to see how the software coped with it. It's an enormous hole in the industry which badly needs filling.'

Products and Problems

Bates' product has not fared too well at the hands of the reviewer in recent months. Does he have anything to say? 'The main criticism is one of speed, which is being addressed. The new version [version 4] was undertaken by a programming team rather than by me, since I seem to be spending so much time in non-virus related areas. I'm hoping in the not-too-distant future to be taking control of the development again.'

Bates is characteristically not afraid to admit his own mistakes: 'My concern is that my reputation for high-speed and accuracy has perhaps been compromised. I think I paid too much attention to the bells and whistles and not enough to the meat of the thing. The only reason for the delay in this

Bates: 'My users come first.'

is because of a range of new developments. Rather than make the changes one at a time, I want to do the whole thing with a bang'.

Aiding and Abetting

Bates typically undersells his contribution to the fight against computer crime. For example he was instrumental in the fight to bring Dr Popp to justice. 'When the Aids disk situation broke, it happened that I was feeding *PC Business World* articles and information about viruses in general. They started to receive letters about a disk which their readers believed had been sent out by them. They sent me a copy of the disk by *Red Star*, and it fairly soon became apparent that this thing had an unusual installation routine. The first thing I did was to write a program that would remove the installation.'

'The number of phone calls that we received went up and up, and it soon became clear that this was a major incident. The Aids disk contained 146 Kbytes of code written in a high level language - not an easy thing to take apart. However, I got lucky, and eventually cracked it, and wrote a cure program. I didn't charge for the cure program, because I felt that if I charged for the solution I would be no better than the guy that wrote it. Since then, I've read that I got paid thousands of pounds for the work I did. If only!'

Law and Order

Ever since this case, Bates has been a stalwart supporter of the British Police force, assisting in numerous raids and cases all over Britain. Typically, Bates does not mention that the vast majority of this work has been done for no personal gain whatsoever. 'The police themselves don't maintain their own expert staff. If they are going in somewhere with a VAX they will bring in a VAX expert. If they are dealing with PC's then they will call me and ask if I can liaise with the appropriate force.'

This police work takes a great deal of time. Does it encroach on his other duties? 'Not at all' he laughs, 'It's absolutely fascinating work, and I have tremendous respect for the officers doing it in very difficult conditions.'

Bates' views on virus writers are well known, and he is not afraid to be outspoken. 'My passion is the virus writers. They have no conception of what it is they are destroying. The distance computers have come in my lifetime is incredible - the amount of power computers have now compared to what they had only twenty years ago is almost frightening. It's fairly obvious that the destruction of trust that the virus writers have caused is massive. Hackers are one thing - they're like thieves, they have to do it themselves. But a virus writer - it's like introducing poison into a water supply. It is such mindless vandalism. I don't understand it.'

The Way Ahead

Asking an anti-virus researcher for a prediction of the future is an instant way to stop him from talking. Bates was suitably circumspect about what the next few years would bring. 'I'm very wary of predictions. It depends on which day you catch me. Some days I feel very depressed and I think that things will get steadily worse and worse. On the other hand, I feel that we are in an industry where the better parts respond to pressure. At the moment, they have us on the run, but my feeling is that things are starting to swing.'

'I think the future has to be generic. With a virus-aware integrity checking package, if it tells you something has changed then you know it is because of a virus. I did see a message on a BBS somewhere, from a supposed virus researcher, which said ''looking for virus-like activity is futile. Viruses do the same things other programs do.'' This is not true - viruses replicate. I firmly believe generic integrity checking is the way forward - something which checks the integrity of the system as well as the files.'

Final Thoughts

In a market driven by hype, Bates' claims are different from many of his competitors': 'If somebody said to me ''how many viruses does your package find?'', my response would be "the one your machine is infected with." - and that is all it needs to find. I don't care how many viruses people have in collections - I'm not interested in their collections. I'm interested in the user's machine. At the end of the day he is the guy that I am trying to help. I'm not trying to help reviewers, or magazines or anti-virus houses; it's not even done primarily to help the police. My users come first.'

VIRUS ANALYSIS 1

Eugene Kaspersky and Vadim Bogdanov

The Volga Virus Family

The vast majority of all virus trigger routines simply involve either displaying a silly message, overwriting the hard drive or both. Indeed, while virus authors seem to spend a great of time thinking of new ways to infect a system, little thought ever seems to be given to the trigger routine, which is the virus' *raison d'etre*. Unfortunately the Volga virus family is an exception to this rule.

The family consists of several variants which are all related to the New Zealand II virus. They were discovered in the Volgograd State University in Russia, and are internally dated from July 1991 to the end of April 1992. All of the members of the Volga family occupy one disk sector, and take up one or two Kilobytes of memory when resident.

Operation

There is nothing particularly novel about the way the Volga viruses replicate. When a machine is booted from an infected hard or floppy drive, the virus installs itself into high addresses of system memory, then checks the hard disk Master Boot Record to see whether it is infected.

If the hard disk is not infected, the virus uses a standard boot sector virus infection routine. The original contents of the MBS is encrypted and stored in an unused sector of the hard drive, and the virus code is inserted in its place. The encryption algorithms vary between different members of Volga virus family. Once the virus is resident, it hooks INT 13h, and infects any suitable floppy disks placed in the disk drives. None of this is particularly noteworthy, however the Volga family of viruses is interesting because of an unusual (and extremely annoying) trigger routine.

Destructive Trigger

All the viruses in the Volga family have the unfortunate side-effect that once a PC is infected, it is very difficult to recover the information stored on the drive. Even after the virus has been removed from the machine, a further clean-up procedure is required to restore normal functionality.

The virus author uses the fact that the fixed disk controller stores a error correction code (usually four or six bytes in length) at the end of every sector. The disk controller uses this information for error checking and error correction of the data stored within that sector.

If the extra information stored at the end of a sector is not what the disk controller expects, then an error code is returned, and the read request fails. However, IBM was prepared for this eventuality and implemented a call which allows software to read the entire contents of a sector, including this extra information.

When one of the Volga viruses is resident, it intercepts calls to INT 13h and substitutes the two calls

`INT 13h, AH=02h`	read disk sector(s)
`INT 13h, AH=03h`	write disk sector(s)

with

`INT 13h, AH=0Ah`	read long hard disk sector(s)
`INT 13h, AH=0Bh`	write long hard disk sector(s)

These substituted calls use exactly the same registers and return the same values, so no additional programming needs to be done to ensure that the read long calls function correctly. However, this is a process fraught with potential pitfalls. The IBM BIOS Interface Technical Reference Manual states that services 0Ah and 0Bh are 'reserved for diagnostics', and that these calls should be used with care.

"the time taken to recover data from the hard drive classes it as one of the most irritating viruses in the wild"

The Trigger in Action

Therefore when an INT 13h write request is issued, the virus intercepts the call and changes it into a 'write long sector' call. As discussed above, this means that the sector is no longer readable by standard calls to the BIOS.

However, when the virus is memory-resident, all read requests (INT 13h, AH=02h) are altered to 'read long sector' requests (INT 13h, AH=0Ah). This 'read long sector' call will read not only sectors which have been altered by the virus, but sectors which have been written by DOS in the standard format. As long as the virus is memory-resident the computer will appear to operate normally.

The catch is that if the hard disk is accessed without the virus memory-resident (either after clean booting or after the machine has been disinfected) the standard DOS functions will not be capable of reading the rewritten long sectors. This occurs because the standard INT 13h call cannot read these altered sectors correctly.

Cleaning Up

Even though it is relatively easy to disinfect machines infected with these viruses, recovering the data stored on affected hard drives is a tricky task, best carried out by a program written specially for that purpose. This program has to read all sectors on the hard drive, and, if it encounters an error, attempt to use the 'read long sector' function call. If this call is successful, the sector should be rewritten using the standard write sector call.

As the only way to test if a sector is affected is to read data from it, this procedure can take a lot of time to complete - from several minutes to an hour, depending on hard disk size and speed. This makes the Volga family of viruses one of the most difficult from which to recover.

Although the trigger routine should not cause data loss, the time taken to recover data from the hard drive classes it as one of the most irritating viruses in the wild. One can only hope that the last virus in the Volga series marks its author's last attempt at virus writing.

VOLGA

Aliases: VolGU

Type: Resident, Master Boot Sector.

Self-Recognition:

Disk: Text string at the beginning of MBS. Varies for different variants.

System: Varies for different variants.

Hex Pattern: Positioned at offset 0 of sector 0

```
Volga-A: BE00 7C33 FFFA 8ED7 8BE6 FB9A 3000 C007
Volga-B: BE00 7C33 FFFA 8ED7 8BE6 FBEA 3A00 C007
Volga-C: BE00 7C33 FFFA 8ED7 8BE6 FBEA 3000 C007
Volga-D: BE00 7C33 FFFA 8ED7 8BE6 FBEA 3000 C007
Volga-E: BE00 7C33 FFFA 8ED7 8BE6 FBEA 2901 C007
Volga-F: BE00 7C33 FFFA 8ED7 8BE6 FBEA 3301 C007
```

Intercepts: INT 13h for infection and damage

Trigger: Rewrites sectors on the hard disk drive using the INT 13h 'write long sector' request, making sectors unavailable when the virus is not memory-resident.

Removal: Specific and Generic removal is possible under clean system conditions.

VIRUS ANALYSIS 2

Jim Bates

Pitch - A new Virus High Note

In spite of the increasing complexity of viruses arriving on my desk these days, there is still the occasional trivial and primitive specimen which makes me grit my teeth at the sheer irresponsibility of the originator. This cause of this month's ire is a 593 byte virus which infects COM files in various directories on the host machine.

Although the whole of the virus code does become memory-resident, the infection cycle is a one-shot mechanism which is only invoked when an infected file is executed. The virus contains the usual crop of mistakes and under certain circumstances will irreparably damage infected files. However, the trigger routine is not intentionally destructive, as it simply causes a high-pitched whine to be emitted from the computer's speaker.

Installation

When an infected file is executed, the virus code is run first and begins by allocating two memory blocks for its own use. Processing then passes to a routine which attempts to find files with a COM extension in the current directory of the active drive.

Once a suitable file is found, it is infected and a counter is decremented. When the counter reaches zero or there are no more matching files available, processing returns to the calling routine. The starting value of this counter is not initialised at this stage and it is therefore not possible to predict how many files will be infected.

A secondary infection routine is then called which attempts to get to the root directory of drive C. If this is successful the find and infect routine is called again (without resetting the infection counter).

Once the requisite number of *.COM files in the root directory (including COMMAND.COM if it is there) have been infected, the routine shifts its attention to the first subdirectory and infects any COM files there. In this case the counter is set to 3 before the infection search begins and a check is made to ensure that at least one file is infected before the routine is exited.

If this check fails (i.e. no suitable files were found), then the next subdirectory off the root is tried, and so on. Once all available files in the root and primary subdirectories of the

C: drive are infected, the machine will hang. Subsequent attempts to execute an infected COM file will also have the same effect.

Once these infection routines have completed, an 'Are you there?' call is issued to determine whether the virus is memory-resident. If it is, processing passes to the host repair routine which replaces the original block of 593 bytes at the head of the file and passes control to it.

If the virus code is not resident, an additional 42 bytes of memory are allocated from system resources and the two interrupt interception routines are copied into it. The addresses for these routines are then hooked into the system and processing finally passes to the host repair routine and thence to the host program.

Resident Operation

The first interception routine simply installs an INT 47h routine which serves to answer the virus' 'Are you there?' call. INT 47h is not used by DOS and on most systems will remain unallocated; however there is at least one application package which uses it (a network oriented database engine from *Gupta Technologies*), and machines running this package will malfunction in an unpredictable manner in the presence of this virus.

The second interception routine takes over the timer tick routine at INT 1Ch. As in the previous case, this interrupt is not used by DOS but again there are several packages which use it on an occasional basis and malfunctions will certainly occur in these cases.

The interception maintains a counter which is initialised to a value that represents a time delay of approximately ten minutes. Once this delay has elapsed, the routine accesses the sound control ports and causes the speaker to emit an annoying high-pitched note (slightly above the highest note on a piano). This will then continue until the machine is switched off.

Neither of the interception routines attempts to maintain connection with any previous routines at the specified interrupt locations.

Infection Processing

This virus only infects COM files and makes no check of their size or the content of the header. A block of 593 bytes is copied from the beginning of the file and appended to the end. The virus code is then written over this initial block so that it executes first. Repairing the host file is a reversal of this process. COM files greater than 64,942 bytes will be irreparably damaged.

Once infected, the seconds field of the time stamp of the infected file is set to the ubiquitous 62 seconds. The virus has no stealth capability and infected files will appear 593 bytes larger than their original size (except in the case of large files mentioned above).

Conclusions

The mismanagement of memory resources by this virus makes it unlikely to spread very far. Unpredictable system crashes will occur at random intervals depending upon any other memory management software that may be operative. In addition, its rather obvious trigger further limits the likely spread of the sample.

This is just another poor attempt at virus programming. The range of mistakes in the code suggests that the author has very little experience in assembly language. Fortunately this misbegotten creation will cause no problems for existing anti-virus software and is best consigned to the dustbin of history.

PITCH

Aliases:	593
Type:	Resident Parasitic COM infector (including COMMAND.COM).
Infection:	All COM files.
Self-Recognition:	
Files	Time stamp is 62 seconds.
System	88h in AL, INT 47h returns 44h in AL shows virus is resident.
Hex Pattern:	
	`8916 1403 8B16 1803 81C2 0001`
	`0316 1403 8916 1A03 B43F 8B0E`
Intercepts:	INT 1Ch for trigger routine. INT 47h for 'Are you there?' call.
Trigger:	Ten minutes after system infection occurs, speaker emits a continuous high pitched tone.
Removal:	Specific disinfection is possible in most cases. Under clean system conditions, identify and replace infected files.

VIRUS ANALYSIS 3

Roger Riordan
CYBEC Pty.

The Loren Virus - Viral Nitroglycerine?

An investigator specialising in computer fraud was recently called in to a local school when the hard disks on a number of PCs malfunctioned. What he found was a previously unreported virus, Loren. The virus is a fairly normal parasitic virus, but has a couple of twists. The most significant of these is that it traps the CP/M compatible Find First and Find Next functions (INT 21h functions 11h and 12h) and infects every executable file returned by either function.

I would not have thought many programs would still use these calls, but was surprised to find they are used by the DOS command DIR. As a result, whenever the DIR command is issued on an infected PC, every eligible file in the directory is infected. To make matters worse, the virus contains a counter, which is zeroed when the virus goes resident, and then incremented each time a file is infected. When the counter reaches 20, the trigger routine is executed.

The trigger routine attempts to reformat cylinder 0, head 0, using a technique which will bypass most, if not all, active monitors. If this fails, it tries to do the same to drive A and then drive B. If it is successful the following message is displayed:

```
Your disk is formatted by the LOREN virus.
Written by Nguyen Huu Giap.
Le Hong Phong School *** 8-3-1992
```

It may be difficult to recover affected disks. The usual panacea, FDISK /MBR, will not work, popular utilities may refuse to recognise the drive, and even a low level format may fail. A PC shop called in during the original outbreak still has one drive it has been unable to recover!

General Information

The Loren virus infects all files opened for execution, and all COM and EXE files reported by INT 21h functions 11h and 12h. The virus increases the length of infected files by 1387 bytes. The virus has limited stealth capabilities, as it contains code to fake the file size of infected files.

The virus code is added to the end of the file, and the EXE header (or the start of a COM file) is patched in the normal way, so that the virus is executed before the original program. The virus will always try to infect the file specified by the COMSPEC= statement in the environment.

Infected programs continue to run, and can be disinfected by removing the virus and putting the original values back in the header. However the required information is encrypted, and must be decrypted before it can be replaced.

The virus installs a special handler for INT 01h, which it uses for self recognition. INT 01h is the Single Step interrupt used by debuggers, and was presumably chosen to make analysis more difficult. However this is not likely to cause significant problems for anyone examining the code.

Installation

Most of the virus is encrypted, using a fixed routine with a variable key. When an infected file is run the virus decrypts a small block which contains the recovery information, and then issues an INT 01h. If the virus is already memory-resident the INT 01h handler will intercept this call, restore the original file, and run it.

If the INT 01h is not present, the virus reduces the size of the current memory block by 60h paragraphs, decodes the main body of the virus and copies itself into this 'hole' in memory. It does not check that the current memory block is actually the last one. It then zeroes the infection counter and hooks INT 01h and INT 21h. The INT 01h handler is used for self recognition, and the INT 21h handler looks for files to infect. The interrupt handler also contains a routine for a new INT 21h call, B5h, which is used by the virus whenever a suitable candidate for infection is found.

The virus assumes that the environment starts with the statement 'COMSPEC=', and attempts to infect the specified program. Finally the virus issues another INT 01h. The newly installed handler intercepts this, restores the original file and executes it.

Interrupt Handlers

The handler for INT 01h simply pushes one of two addresses (depending on whether the file is a COM or an EXE file) onto the stack and returns to it. There is no attempt to check that the call was issued by the virus.

If an INT 21h call is issued with AH=11h or 12h, the virus allows the call to proceed, but examines the returned values. If the call returned the filename of any EXE or COM file, the virus converts the directory entry to a path name and uses INT 21h function B5h to call the infection procedure. This sets a flag to indicate whether the file is infected. If it is, the length field in the directory entry is adjusted to reflect the original file length and the doctored entry is returned to the calling program.

If AX=4B00h the handler simply calls the infection procedure, and then passes the call on to the original handler.

The Infection Routine

If AH=B5h, the infection procedure is called. This installs a temporary INT 24h handler, saves the file attributes and clears them, opens the specified file, saves the date & time and reads the start of the file. It then checks if the file begins with 'MZ' or 'ZM'. If so, it assumes the file is an EXE file and reads the CRC from the header, subtracts the initial values for CS and IP. If this value is equal to 01B3h, the virus assumes the file is already infected.

If the file is not already infected, the contents of the EXE header are saved and replaced with new values and the header is rewritten. A similar procedure is used to rewrite the start of COM files. Infected COM files are recognised by the bytes 'RC' immediately following the initial jump. If this is not found, the first five bytes are saved and patched and the start is rewritten.

In either case the virus is then encrypted in two parts, using keys derived from the clock, and written to the end of the file. The infection counter is incremented, the date, time and attributes are restored and, if infection was due to a Load and Execute request, the file is executed.

Trigger Mechanism

The infection counter is zeroed when the virus is installed. Every time a file is infected this counter is incremented. When it becomes greater than 20 (decimal), the trigger routine is executed and the counter is reset.

When the warhead is triggered, the virus attempts to format head zero, cylinder zero, using INT 13h, function 5. It first tries drive C. If this fails, it tries drive A, and finally drive B. If it succeeds, the virus displays the message shown above. The format is performed by setting up the appropriate registers and then issuing a far call to the address stored at a particular location in the DOS area. I have not found it documented anywhere, but on all the PCs I have checked, it contains the address of the INT 13h handler which was present when DOS loaded. Thus, many active monitors will be unable to intercept the command.

The track is formatted with non-standard data, and FDISK will not recover the drive - it will be necessary to do a low level format. Some utilities permit a single track to be rewritten and if this can be done successfully there should not be any loss of data. It may be necessary to return some IDE drives to suppliers.

Symptoms

At first glance it may seem surprising that the trick of infecting on DIR has not been tried more often. However this causes a lot of extra disk activity, which leads to a noticeable degradation in performance. I was not prepared to test this on my hard disk (for obvious reasons!), but when I checked it on a floppy with 29 files, I found that instead of the normal 3 seconds, DIR took 42 seconds the first time, and 25 seconds on subsequent passes.

Conclusions

This virus does not introduce any significant new techniques, but is very destructive, as it has a very sensitive and damaging trigger. It is very infectious, but is probably too obviously destructive to spread very widely. However the fact that it can be set off by running an infected file, and then doing a single DIR, demonstrates the limitations of integrity checking software. Active monitors are unlikely to be able to intercept the trigger routine, though most should be able to detect the infection process.

It should also be noted that any scanner, or integrity checker, which did not detect the virus in memory, and used INT 21h functions 11h and 12h to search for files, would trigger the warhead. The viral code incorporates very little error checking and may interfere with other programs. It will almost certainly interfere with any program using INT 01h.

LOREN

Aliases:	None known
Type:	Memory-resident, parasitic file infector.
Infection:	COM and EXE files.
Self Recognition:	
Files	EXE File CRC = CS + IP + 1B3. COM File Bytes 3, 4 = 'RC'.
System	INT 01h handler present.
Hex Pattern:	
	`502E 8B86 D005 2E89 86DB 0558` `C3E8 0000 5D81 ED49 05E8 9400`
Intercepts:	INT 01h Used for self recognition. INT 21h Functions 11h, 12h, 4Bh, and B5h (private function) for stealth and infection.
Trigger:	Cylinder zero, head zero, formatted with non-standard data.
Removal:	Exact recovery is possible, but specialised software is required.

FEATURE

Jim Bates

The Other Virus War

There are two distinct schools of thought, in the virus world, about the best way to prevent the spread of viruses: in one camp are the advocates of scanning, in all its forms, and in the other are those who press for generic virus detection.

I had thought that the battle was over, but recent comments in various computer publications seem to indicate that though the discussion is dead, it simply will not lie down. Let me map out the field and present as best I can the relative pros and cons of each choice.

The Battle Lines

Broadly speaking, there are two ways of protecting a PC against virus attack. The best known, and easiest to understand, is scanning for known virus code and warning the user about it. The user is then expected to take appropriate action to prevent infection, corruption or destruction of his data. Less easy to understand and more difficult to implement (both as code and in computing practice) is the generic access control approach. Basically this consists of verifying a clean operating environment and maintaining its integrity throughout subsequent normal operations.

There is no doubt that the rapidly increasing number of different viruses is causing serious problems for the scanners and despite the best efforts of the more aggressive members of the scanning fraternity, there is a distinct shift towards access control and automatically maintained system integrity. There appear to be fairly simple reasons for the insistence on virus specific scanning, but before I look at those, perhaps I should sketch in a few of the relevant details.

The Objective

The whole problem hinges on 'unknown' software. PC users are an adventurous breed, and there are many occasions when 'unknown' software may be introduced into a system, without regard to its origins or possible content. This can range from simply transferring data between machines (without regard for the possibility of boot sector infection), to running programs like games or utilities without checking them.

Proponents of the scanning method will insist that the only way to verify the cleanliness of such disks or software is to scan them. This ignores the fact that scanning software will always be out of date and can only identify virus code known to the vendor at the time of the last update.

Those in favour of generic detection will equally insist that unknown software should be tested on a special PC equipped with a comprehensive range of monitoring programs. Each method has its advantages and disadvantages and these should be clearly understood both from a technical and a practical point of view. The goal, however, is the same: to provide secure computing for the users at minimum cost in both time and money.

Scanning

Since the virus threat first materialised, scanners have developed beyond the simple pattern recognition routines with which they began. It is now possible to analyse the structure of program code in a way that can identify multiply encrypting viruses with almost 100% accuracy and no false positives. Some scanners can even execute part of the code under tight control to complete decryption routines and thereafter examine the decrypted code.

"scanning software will always be out of date and can only identify virus code known to the vendor"

As viruses have become more complex, so have the scanners, and it is still generally accepted that we have yet to see the first truly unidentifiable virus. This sounds great, but there is a penalty: the more complex a scanner becomes, the more time it takes to execute. This penalty is also increased by the sheer weight of numbers involved.

So while scanners got off to a flying start, their limitations are now beginning to show and as time passes, their general usefulness will diminish even further. It used to be quite acceptable to scan your whole machine each day before beginning work. However, this is now becoming counter-productive as a general protection measure, because of the time that it can take to scan the entire drive.

It should also be remembered that updating scanners properly requires that each virus should be accurately disassembled and analysed before its recognition profile can be incorporated into the next update. This is not to say that scanners will ever die out completely. There will always be a need for accurate identification of virus code, if only to check that a particular virus has not introduced insidious system corruption during its period of control.

Another serious problem when using scanners is updating them. Consider a Technical Support Manager who is responsible for several thousand machines. If the scanning package he uses is updated monthly, for every thousand machines he controls, he must update 50 per working day just to keep up, and even then, a good proportion of them will be weeks behind the update. It does not solve the problem to reduce the update frequency because then the level of protection decreases, since new viruses are arriving daily in ever increasing numbers.

In summary, the pros and cons of a scanner are:

✓ It is very easy to write a simple scanner.
✓ It provides a proactive way of stopping a virus entering the system.
✓ Users understand what a scanner is and what it does.
✗ It requires frequent updating.
✗ It is ineffective against unknown viruses.
✗ The number of 'difficult to detect' viruses is growing.

Generic Checking

Properly written integrity checking programs do not suffer from update problems in the same way that scanners do. In spite of much ill-informed criticism of integrity checkers; when tailored specifically to virus techniques they are undoubtedly a very effective way of maintaining a general watch on the functioning of PCs.

Essentially they must first be run on a known clean system to build an integrity database of information about the condition and contents of each file. Subsequent invocations will then check each program to ensure that it has not changed since it was first introduced to the system. This is not quite as easy as it sounds but it *can* be highly effective, very fast and, if well researched, extremely difficult to circumvent.

Virus-aware integrity checkers are much misunderstood, even by many self-appointed experts. Although a simple 'change detector' can be beset by false positive problems, it is possible to analyse the nature of the changes made to a file and distinguish between added or updated files and ones which have been attacked by a virus.

Another type of integrity checking program concentrates on monitoring the state of the system services. If a program attempts to hook into the system in an unorthodox way, the monitoring software will immediately begin an additional series of checks to determine the intruder's motives.

For example, many programs will hook the main DOS interrupt service 21h, some programs may hook the disk BIOS interrupt 13h and a few programs might write data or code back to executable code. However, a program which does all three of these things would be extremely rare and should be considered highly suspicious.

Contrary to a popular belief, it is possible to write generic virus detection software which does not deluge the user with a barrage of false positive results. The actions of a computer virus are pretty specific: after all, how many pieces of code actually patch additional sections into an executable file?

The main limitation with such checking programs is that validating unknown software can only be accomplished reliably on a machine deliberately set up to risk becoming infected. This does not take into account the possibilities of sparse infectors, such as a virus which only infects on certain days of the week.

However, the advantages gained by using generic anti-virus techniques are considerable because well-written generic software can provide protection without requiring regular updates and can also repair files infected by hitherto unknown viruses. Changes will only become necessary if a new technique arrives that was not anticipated in the original protection design.

In summary, the pros and cons of generic techniques are:

✓ They can detect unknown viruses.
✓ They require no updates.
✓ The rapid increase in virus numbers is not a major drawback.
✗ They can cause problems when new software is installed.
✗ They are currently prone to false positives.

Memory-Resident Protection

The question of resident versus non-resident virus protection is just an additional skirmish point in the overall argument. Quite obviously, if protection can be incorporated into the system, there is far less reliance upon the user to complete a particular series of actions (ie scanning or integrity checking) on a regular basis.

There are two major considerations when examining memory-resident software - the integrity of the system services that resident software needs to use and the amount of memory which may be needed. This latter requirement immediately exacerbates the position for scanners.

It would be ideal if a resident program could be designed to scan any program presented for execution in an attempt to identify virus code before commencing execution. However, increasing virus numbers will naturally increase the search database and even if only an index is maintained in memory, the memory requirements must increase. Even if things were

arranged so that the database was only accessed on the disk, the penalty of increased execution time would become unacceptably intrusive to the user and the memory requirements prohibitive.

Arrangements could certainly be made to use extended or expanded memory, but this is not always available and is fraught with technical difficulties, particularly under *Windows,* or any other multi-tasking system.

The generic approach lends itself more readily to resident operation since it only needs to verify that the target program file is unchanged and even extremely sophisticated code can be packed into very tight spaces. The integrity database would naturally be accessed only on disk, and only once during the checking process.

"the ideal solution would be an automatic approach in which virus infected code would not be allowed to execute"

The biggest problem though, is system integrity. When a virus is memory-resident, one can no longer trust the system services on a PC to return correct information.

Let us assume that we have a simple, resident checking program that has in its database all the necessary information to verify whether a certain program file has changed. The machine is now infected with a stealth virus so that our specimen file becomes infected.

When we next attempt to execute the file in question, the simple integrity checker intercepts the request and uses DOS services to check the file. The stealth routines may be intercepting DOS file access requests and substituting clean code in the returned information. Our simple resident checker would see clean code and report no problem.

A more intelligent checker would collect the same information via two or more methods of access - only if the results matched would processing be allowed to continue. File and disk integrity checking can only detect changes *after* infection and thus provide a fail-safe if an infected file is somehow brought into the system.

What About The User?

If we look at the problem from the user's point of view, the ideal solution would be an automatic approach in which virus-infected code would simply not be allowed to execute. This is still only a pipe-dream, but developments in several packages are encouraging. The main approach works like this - a generic system and file integrity checking program is made resident in memory and continually monitors operations looking for activity known to emanate from virus code.

Unknown software presented to the system is recognised as such and sent for analysis before being allowed system access. This uses the strength of the scanner in checking unknown software before its details are passed to the generic database. The end result is a system which adds an acceptably small overhead to normal operations and yet provides a blanket protection for the whole operation.

So why has the user had to wait so long for this and why is there still so much emphasis on scanning alone? To answer the second question first, scanners are easy to write - scanning is an easily understood process and testing a scanner is apparently something that anyone can do. Just get yourself a collection of viruses, scan them with the product of your choice and log the results.

The reason why there has been such a long wait for an effective generic system protection program is that such code is much more complex to design, requires a much higher degree of technical skill and system knowledge to write and needs a far wider understanding of the range of techniques that virus code uses. We are not there yet, but most of the reputable anti-virus vendors are beavering away at the necessary research - it is just a matter of time.

Conclusions

A final observation may help to set the whole problem into proper perspective. While scanning is advocated as the main defence against virus code, we are locked into the spiralling rise of virus numbers.

A new virus is written and distributed. After discovery and analysis by a virus researcher, its details are added to the scanning database and updates are issued. Then another new virus is written and distributed, and after discovery... etc, *ad nauseam*. With generic protection, only new virus techniques will require the protection software to be updated and each technique detected will mean writing new undetectable viruses will become harder.

This means that there is a place for all three techniques discussed here in a well implemented anti-virus package. By combining the best features of all these techniques, it should be possible to maintain a reasonable level of security on a computer system without too much work on the part of the users. A multi-pronged attack is best, and users should beware of anyone who advocates only one of these methods as providing adequate protection.

PRODUCT REVIEW 1

Dr Keith Jackson

MS-DOS 6 - Worth the Wait?

VB claims timeliness as one of its virtues, and as *Microsoft* has just released version 6.0 of the *MS-DOS* operating system, which includes built-in anti-virus features, this month seems the appropriate time to take a look at this upgrade. Note that the version of *MS-DOS* looked at in this review is an *upgrade* - it assumes that some previous version of *MS-DOS* is already installed.

The review copy of *MS-DOS* was provided on three 1.44 Mbyte, 3.5 inch, floppy disks. I am not sure what other formats are available, as the manual does not seem to discuss this point. No doubt *Microsoft* has some means of obliging those users who have PCs without 1.44 Mbyte disks, but prospective purchasers should beware.

Given that *MS-DOS* now incorporates some security features (see below), I was pleased to see that all the floppy disks were provided in permanently write-protected form.

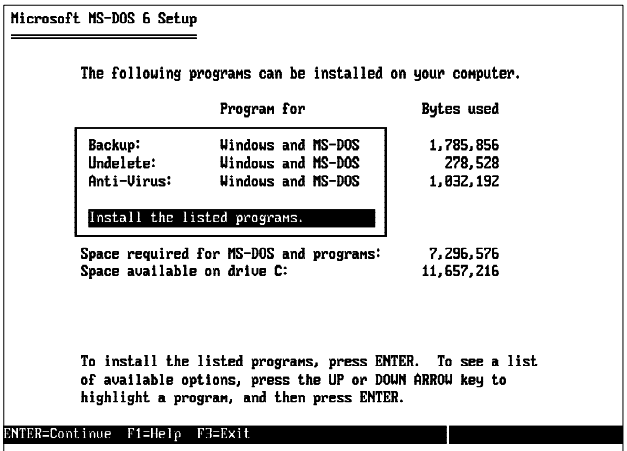
Documentation

The documentation that comes with the *MS-DOS 6* upgrade comprises a single A5 manual. At 321 pages long, it is well written, easy to understand, and contains a thorough 18 page index. It is noticeable that the space taken up by an explanation of the vast majority of the *MS-DOS* commands (the ones that were available in previous versions) occupies only 10 pages. It is really necessary to have an old version of the *MS-DOS* documentation to hand, unless the somewhat terse help facility is adequate for your needs.

Installation

Even though the documentation says little more than 'insert the first floppy disk, execute the program called SETUP, and answer the questions on screen', installation proved to be very straightforward indeed. The whole process took about 20 minutes on my *Toshiba 3100SX* laptop computer.

Oddly, complications arise if your computer currently has an *OS/2* partition, and you wish to upgrade to version 6.0 of *MS-DOS*. In this case the documentation contains several *pages* of explanation of what to do in various circumstances. The recent rift between *IBM* and *Microsoft* has obviously widened to the extent that *Microsoft* seems to be doing all it can to ensure that two operating systems will not co-exist happily, a point of view I find very childish.



It is lucky that *MS-DOS 6* has built-in disk compression as the days of small operating systems are long gone. Wave goodbye to seven Mbytes of disk...

Installation of this version of *MS-DOS* suitably modifies the CONFIG.SYS and AUTOEXEC.BAT files, and when the computer is rebooted, it also permits the user to select whether or not to execute these files. If a '?' is placed before the '=' sign in a 'DEVICE' line within the CONFIG.SYS file, then the user is prompted for confirmation that this device driver should be installed at boot time.

During installation, SETUP offers to include several new features: *Backup*, *Undelete* and *Anti-Virus*. The screen shows how much hard disk space will be occupied by these features, and permits a choice between DOS-only installation, *Windows*-only installation, or installation for both DOS and *Windows*. It is very noticeable that the default setting is to install these features for *Windows*-only - adding weight to the current speculation that *MS-DOS 7* will be completely interlinked with *Windows*.

The hard disk space required for version 6 of *MS-DOS* is 5.4 Mbytes if DOS-only versions of these programs are installed, 6.1 Mbytes if *Windows*-only versions are installed, and 7.3 Mbytes if both versions are installed. The *Windows* specific programs are installed in a specially created *Windows* group called '*Microsoft Tools*'.

Anti-Virus Software

The anti-virus features provided with v6 of *MS-DOS* are a lightly disguised (ie badged) version of *Central Point's Anti-Virus (CPAV)* program. I reviewed this software as a constituent part of the *PC Tools* package only 4 months ago (see *VB* Jan. 93), and with the exception of a name change, I am hard pushed to see many differences between the *Microsoft Anti-Virus (MSAV)* program and *Central Point's* original offering. This similarity even extends to the inclusion of various bugs in *MSAV* which were also present in

previously reviewed versions of *CPAV*. As a stand-alone program, *Central Point Anti-Virus* was first reviewed by *VB* in June 1991, and again in May 1992.

In a previous review I commented that the *CPAV* documentation was very thorough, but *Microsoft* has reduced this to just a few pages in the DOS manual plus some on-line help. Even so, *MSAV* is probably quite usable by all except the most naïve user. Esoteric features such as immunisation are not included with *MSAV*, but the main features of scanning and file integrity verification are.

Speed And Accuracy

MSAV is capable of scanning a hard disk under either *Windows* or DOS. The computer used to produce the following test results was a *Toshiba 3100SX* laptop containing 827 files spread across 24.8 Mbytes; the hard disk used *Microsoft Doublespace* data compression (see below). The time taken by *MSAV* to scan this hard disk varied enormously depending on what options were selected, ranging from 5 minutes 34 seconds under *Windows* when all files were subject to scanning and integrity verification, down to 1 minute 14 seconds under DOS when the default options were selected. None of the above figures includes an overhead of 18 seconds to scan memory before the scan of the hard disk commences.

For comparison purposes, *Dr Solomon's Anti-Virus Toolkit* scanned the same hard disk in 53 seconds, and *Sweep* for *Sophos* took 60 seconds in quick scan mode, and 6 minutes when doing a complete scan. The *Windows* version of the *MSAV* software includes an excellent point and shoot system that provides a short explanation of the salient points of each virus about which *MSAV* knows.

The accuracy of virus detection was reasonable, but rather surprisingly slightly worse than that reported in the January issue of *VB* for the *CPAV* program. The previous review reported that *CPAV* detected all but 4 of the 215 viruses it was tested against: it missed Kamikaze, Rat and 2 copies of the Amstrad virus. Using the same test-set, *MSAV* failed to find 10 viruses; the four viruses quoted above and 1260, Anthrax, Casper, V2P6 and two copies of PcVrsDs. This makes *MSAV* look rather out of date when compared to its older half-brother - perhaps *Microsoft* is not very concerned about upgrading the anti-virus software in good time.

As mentioned earlier, some bugs pointed out in previous *VB* reviews are still present. De-installation still leaves checksum files scattered throughout every directory of the hard disk, which is very annoying. The file integrity 'checksums' are still not calculated across the entire file, and only seem to refer to the file's date, time and size. Alterations can therefore be made to a file's content which would not be detected by the file integrity checks. Last but not least, when tested against 1024 samples of the Mutation Engine, *MSAV* consistently locks up after detecting 255 samples.

Upgrades

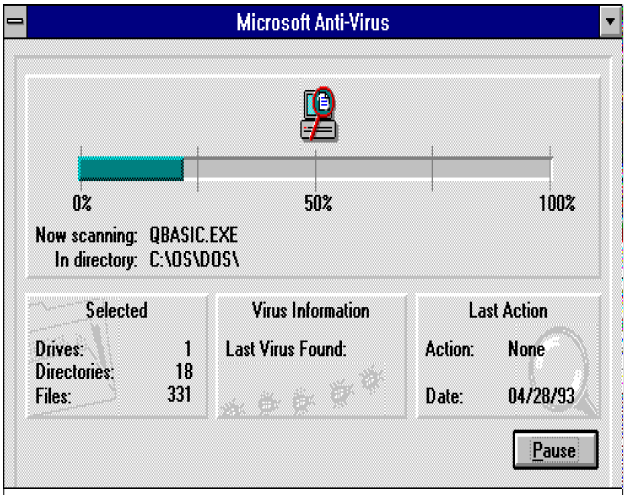
Upgrades of *MSAV* are available for $14.95 in the United States, beside which the UK rate of £14.95 seems somewhat inflated. Upgrades are offered in most major Western countries.

I do not know what corresponding version of *CPAV* this will upgrade *MSAV* to; such details are not to be found in the *Microsoft* documentation. Anyone using *MSAV* seriously should enquire about this - for *CPAV* to keep *MSAV* users permanently behind in the anti-virus stakes, just to further sales of *CPAV* would be extremely irresponsible, bordering on negligence. Upgrades of only the virus patterns can be obtained from a *Microsoft* BBS in the USA.

Backup

Microsoft makes great play of the fact that version 6 of *MS-DOS* now comes with *Microsoft Backup*, 'a program that makes it easy to back up your data'. Perhaps my memory is failing me, but I always thought that all previous versions of *MS-DOS* came with utility programs called BACKUP and RESTORE, which (at the time) *Microsoft* assured us were quite adequate for backing up hard disks.

I am glad to see that *Microsoft* has at least reacted to all the adverse criticism that has been heaped upon these two ponderous utilities and included decent backup facilities within *MS-DOS*. As with the anti-virus software, this new program is a badged version of *Central Point* software.



Microsoft Anti-Virus is basically a badged version of *Central Point's* product. Capable of running under both DOS and *Windows*, *MSAV* is now an extra bolt-on goody. But does it do the job?

The DOS and *Windows* versions both seem to work very well, are easy to use, and I find it hard to find much to say about this particular feature - apart from wondering what the poor souls who do not have a disk drive capable of using 1.44 Mbyte, 3.5 inch, floppy disks are supposed to do, as *Microsoft Backup* refuses to use 720 Kbyte 3.5 inch disks.

Compression Software

MS-DOS now includes a feature called *Doublespace* which applies compression techniques to all data stored on a hard disk, thereby providing an increase in the available storage space. I found *Doublespace* extremely easy to set up (just type '*Doublespace*', answer a couple of questions, and wait twenty minutes), and throughout the testing for this review I found no fault whatsoever with its operation.

Doublespace provided an extra 12 Mbytes of storage on my hard disk, at a stated data compression ratio of 1.6 to 1. The extra storage is somewhat reduced from what would normally be expected on a 40 Mbyte hard disk, as I have several large hidden files which *Doublespace* left on an uncompressed partition of the hard disk.

MS-DOS commands such as CHKDSK and DIR have been modified so that they are now aware of *Doublespace* compression, and they report information about what compression ratios have actually been achieved. I am a long term *Stacker* user [*But I can quit any time I want... Ed.*] and I was very impressed by *Doublespace*. It operates transparently to the user, and stays hidden away where data compression ought to be - buried within the operating system.

The Rest

The program MEMMAKER is provided, which reorganises the software which is installed by the CONFIG.SYS and AUTOEXEC.BAT files so that as much as possible is tucked away in high memory, thereby providing more available conventional memory. The features appear very similar to those offered by the OPTIMIZE program supplied with *QEMM* from *Quarterdeck International.* MEMMAKER increased my available memory from 528 Kbytes to 597 Kbytes, a very worthwhile gain.

UNDELETE offers features which provide three distinct levels of file 'resurrection', ranging from only undeleting if the file's remnant parts have not been overwritten, up to keeping copies of deleted files in a specially assigned area of the hard disk, and restoring them on request.

The INTERLINK program provides client/server features over a serial link between two computers, in a similar fashion to the program *PC-Anywhere* which has been around for some years now.

Power conservation features are included for laptop computers that conform to the Advanced Power Management (APM) specification (whatever that is). My *Toshiba* laptop did not object to this, but installing *MS-DOS 6* did disable the power-down resume feature normally available on it.

Conclusions

I hope that the marketing men at *Microsoft* feel that they have done a good job, because the technical advancement offered by *Microsoft* itself is not far from zero. With a few exceptions, the main improvements seem to come from software sold by other (rival) companies, which have been badge-engineered by *Microsoft.*

I have thought hard and long about the add-ins to *MS-DOS*, and I cannot for the life of me see what *Central Point* gets out of this deal. The computer press has reported that they are being paid no royalties (yet?), and I have little doubt that for all its faults, the *MSAV* anti-virus program will become extremely widespread. Maybe *Central Point* has decided that *MSAV* will be the only game in town in a few years time, and therefore wants to have a piece of the action.

Many anti-virus vendors are going to be hit very hard by the inclusion of anti-virus features within *MS-DOS*. Why pay for something that comes free with the operating system? The obvious answer is if the paid-for product is technically superior, or offers more features. Do users really care? I think not. Place your bets as to who will be most affected, but I am in little doubt that a vast shake-up is imminent.

Life looks far grimmer for *Stac Electronics* (the developers of the *Stacker* software). *Doublespace* is easier to set up than *Stacker* (which I have used without hitch for well over a year), works transparently, is hidden within the operating system, and gives adequate compression. *Stac Electronics* must be fighting for its very existence, for if *Doublespace* catches on, *Stacker's* sales will surely wither away.

Technical Details

Product: *MS-DOS* version 6.
Developer and Vendor: *Microsoft Corporation*, Redmont, Washington, USA. Local support arrangements apply in most countries around the world. BBS for virus signature upgrades, Tel: +1 (503) 531-8000.
Availability: Not explicitly stated.
Version evaluated: v6.00
Serial number: None visible.
Price: Special introductory offer, £49.95
Hardware used: (a) *Toshiba 3100SX*, a 16MHz 386 laptop, with 5 Mbytes of RAM, one 3.5 inch (1.44M) floppy disk drive, and a 120 Mbyte hard disk, (running under *MS-DOS v6*.0!).
(b) 4.77MHz 8088, with one 3.5 inch (720K) floppy disk drive, two 5.25 inch (360K) floppy disk drives, and a 32 Mbyte hard card, running under *MS-DOS v6*.

PRODUCT REVIEW 2

Mark Hamilton

VET - The Wizard of Oz

Cybec is an Australian company led by Roger Riordan. Its anti-virus package, *VET*, was conceived in 1989 to combat an outbreak of New Zealand II at a university and has never looked back since.

Last time *Virus Bulletin* reviewed *VET* (May 1991, p.18) it was noted that its scanning performance was rather poor, and that 'without extending the list of known viruses quite extensively, *VET* will not come close to competitive packages.' It has been two years since Dr Keith Jackson came to this rather acidic conclusion: has *VET* improved?

Reviewer's Guidelines?

Whenever I receive a copy of a newly-announced *Borland* or *Microsoft* product, I am usually sent a Reviewer's Guide: this is often written in a fairly condescending tone (aimed at me) and two-thirds of its content is marketing hype and a 'positioning' statement. This, I invariably ignore.

The remaining one-third contains a 'script' which, if I were to follow it, is designed to demonstrate this shining example of the programmer's art in its best possible light and it is upon this that a number of my colleagues seem to base their reviews. Being a somewhat cynical journalist, I often wonder whether the script was carefully crafted to avoid the numerous bugs that lurk elsewhere in the package waiting to catch-out the unwary user - quite often, it seems, this is exactly so.

Cybec sent along a copy of its Reviewer's Guide and I was somewhat relieved to note that there was no condescension (Australian journalists probably would not stand for it), and it provided hard facts and 'signposts' - things I needed to look out for during the review. The guide was well written and (for once) quite helpful in evaluating the product.

Documentation

Like most - but not all - products in its class, *VET* is delivered on both 5.25 and 3.5-inch floppy diskettes. Also in the box is a 127 page, saddle-stitched, A5 sized manual, a set of release notes and a copy of *Cybec's* irreverent in-house newsletter, Cyclops.

The manual warrants special mention as a shining example of just how software manuals should be written and presented. The first 58 pages are devoted to installing and running the software and the author has included numerous examples of the various screen displays and the prompts a user will face. There follows a chapter which deals with frequently asked questions. Most of these are informative, but the author clearly could not resist the following:

Q: I ran INDEX (.DOC) and was told it was a 'BAD COMMAND OR FILE NAME'.

A: How odd!

Next are several chapters which explain, in simple layman's terms, how a PC works, the distinction of the various virus types and how these differ from Trojan horses, logic bombs and worms. In addition there are technical explanations of the product itself and some of the more common viruses it detects and cures.

There are a number of very amusing cartoons which lighten the tone of the manual, but do not detract from its serious message. Humour is a great way to educate, and *Cybec* seems to have got it just right. All in all, this is one of the best software manuals I have come across - full marks to *Cybec* for their hard work.

Installation

Installing the product is a simple enough affair. During the installation procedure, *VET* needs a blank, preformatted disk for the install program to store configuration information. The routine then CRC checks the installation disk before continuing, to ensure that there everything is as it should be.

It was at this point that I encountered a problem. All my anti-virus software is installed on Drive D - sorry to be awkward, I just prefer it that way. Although *VET* will install in the directory of my choice, the documentation implies that this has to be done on drive C. *Cybec* has noted this shortcoming and assures me that the documentation will be altered to make this clearer.

The install program adds information about the host PC and adds checksums for the installed files to the end of the VET.COM file so that it exactly fills a cluster. The *VET* program file is then encrypted and a new set of checksums is calculated and stored. When *VET* is run, the set of checksums is decrypted and checked. If differences are found, denoting possible infection of one of the *VET* component programs, then *VET* refuses to execute further.

Once *VET* is installed, a copy of it is placed on the reference disk allowing the user to execute the program from there. This could be useful if you have to run around and check a number of machines quickly, since once *VET* is loaded, it makes no further reference to the diskette from which it was loaded. Alternatively, you may prefer to always run your anti-virus software from a floppy. *Cybec* makes this easy.

When installed, *VET* adds only three files to the hard disk: VET.COM and VET.DAT, the scanner files and VET_RES.EXE one of the TSR utilities. Together, these occupy only 98,304 bytes. In the age of disk-hungry applications, this makes a welcome change.

The rescue floppy, on the other hand, contains 8 files, in addition to any operating system files you placed there, two of which contain configuration information specific to the PC upon which it was installed.

A Fast Mover

VET is certainly one of the faster scanners available, even in its secure mode of operation. *Cybec* has developed a special algorithm it calls 'Polysearch' which is designed to speed up the complex recognition routines needed for the more complex polymorphic viruses. The company believes that this innovative technology could be applied to other software, such as maintaining database indexes, so it has applied for a patent.

Certainly its research and development in this area has paid dividends, as *VET* scanned my 'standard' hard disk in just 18 seconds in its turbo mode and in 44 seconds in its secure or full scan mode. In both cases, only files with executable extensions were scanned.

However, its rip-roaring speed has inevitable trade-offs as it failed to identify all the viruses in the test-sets. In both turbo and secure modes, it missed two instances of Whale and one of Spanish Telecom 1 from the *Virus Bulletin* 'In The Wild' Test-set. This is reasonably worrying and needs to be fixed - users have the right to expect their anti-virus software to tackle every single virus found 'in the wild'.

```
  CYBEC Pty Ltd, PO Box 205, Hampton. Vic. 3188, AUSTRALIA. (03/613) 521-0655
        VET #7.242 Virus Protection Program. (C) R.H.Riordan, 1989-93.
                    Friday, 23 April, 1993    10:09:21

 VET  reads the boot sectors of your disks, destroys known viruses, & offers to
 replace any unrecognised floppy boot sector. Do so  unless the disk is a game,
 copy protected program, or backup disk.  VET  also scans all files on the disk
 for viruses, and will recover most infected files.  VET will scan chosen files
 or directories, and subdirectories.  VET provides the following options:

     D  Show error messages           E  List name of EVERY file tested
     L  Log VET output to file        F  Full (dumb) scan for exotic viruses
     N  Do not delete viruses         T  (Thorough) Test EVERY file for viruses
     P  Check 1st 50 files found.     U  Rename all infected files; don't fix
     R  Scan hard disk recursively    Y  Rename suspected infected files
     S  Display all boot sectors      Z  Delete infected files; don't fix
     X  Quit after scanning disk      H=0  Do not check high memory
        (Normally asks for next)      B=215D  Check VET is loaded Here.

eg: Vet          Scan boot sector & programs on disk in A: delete any viruses
  vet c:|d:\./lrx Scan boot sectors & all exec files in all directories
                  on drives C & D, log errors & exit when done.
    vet d:\bin/tr Scan ALL files in dir D:\BIN and all sub-directories.

[C:\VET]
```

No 'pansy' interfaces here! *VET's* command line has a plethora of options making it a sure-fire hit with lovers of this dying art.

It did however, correctly identify all the boot sector infections that form an integral part of this suite; this is just as well because the most commonly occurring viruses at large are all boot sector infectors.

Against the 'Standard' test-set, and in turbo mode, it missed instances of 8 Tunes, Aids, Best Wishes 2, Machosoft, Number 1, Russian 696, Sentinel, Spanish Telecom 1, Terror and Whale. Using the same test-set, this time in its secure mode, it only missed Spanish Telecom 1 and Whale.

Although it claims to detect Mutation Engine viruses, it fails against our 'Mutation Engine' Test-set of 1,536 such infections: in its turbo mode it failed to find 98 specimens but improved its performance when switched to its secure mode by only missing eight infected files.

Options

There are two versions of *VET* supplied, both of which are the same size. VETHDFIX is identical to *VET* except that it will replace the whole of the Master Boot Record, including the Partition Table, in the event it is found to be corrupt or infected. *VET*, on the other hand, does not replace the Partition Table. The company advises users to run *VET* in preference unless the Master Boot Record has been so damaged that it cannot be repaired.

Both *VET* and VETHDFIX are run as command line driven programs and have no fewer than 23 command line switches available. Five of these are documented separately as they are specialist switches and should not be generally used. These control options such as 'automatic repair of infected hard disk Boot Sectors' and 'do not scan hidden files'.

This latter option has me somewhat perplexed. The documentation states that this switch is for 'use with proprietary security systems' - why? Viruses are no respecters of file attributes, and executable files that belong to access control packages - even if they are marked 'hidden' - are just as likely to become infected as any other executable program on the PC. As such executables are often designed to be executed each time the PC is booted-up, a virus could spread its infection within the PC and possibly beyond. Even for the sake of compatibility, using this switch sounds rather risky!

Other options are far more straight forward and control, for example, whether a full secure scan is performed or not, whether or not just to check the first fifty files in a sub-directory and various reporting options, among others.

I applaud *Cybec's* steadfast refusal to turn its package into a full-blooded application by incorporating such annoyances as menus, dialogue boxes and WIMP screens. As I have often stated in the past, anti-virus software is not to be

played with: it is and should remain utilitarian and be fast, accurate and reliable - alas, such attributes are generally lacking in *VET's* GUI and CUI competitors. Neither has the company gone down the sticky path of trying to scan inside compressed executables and archive files.

VET_RES is the Terminate-Stay-Resident component which occupies between 8 and 14 Kbytes, depending on how it is configured. It monitors the DOS Load and Execute service (INT 21h, Function 4Bh) and scans the executable for around 200 of the more common viruses. If one is detected, VET_RES then loads the full *VET* program and instructs it to scan the infected disk. *VET* has to report that all viruses were removed before allowing the file which triggered the scan to be executed.

The TSR also checks the Boot Sectors of floppy disks the first time they are accessed. I did not notice any speed degradation imposed by this TSR, however if the main scanner, *VET*, has to be executed, everything stops until it has completed its clean up.

The documentation mentions a smaller TSR, VET_RES2, which contains no virus information but which automatically invokes *VET* each time a program is run. However, this program was strangely missing from the distribution disk.

The *VET* system contains a few other minor programs and device drivers which include the following:

VET_STOP.SYS a device driver which simply waits for virus 'Are you there?' calls and will terminate the program if a known call is received. Tests for individual viruses can be disabled in the event of conflicts with programs which use such calls. VET-STS.SYS is a similar device driver which has a much smaller memory footprint but does not allow individual viruses to be disabled.

VCRC.EXE is an equivalent program to *McAfee Associates' VALIDATE* and displays Cyclic Redundancy Checksum values for files whose path names are supplied on the command line. These values are not stored in a file, so it can not be considered a generic checker.

Conclusion

All in all, I am impressed with *VET's* performance. Top marks for the documentation, which is excellent and provides a great deal of useful information. The program runs swiftly, and is easy to use.

However the product lacks an integrity checker and is therefore totally scanner dependent. In addition to this, the detection results were rather disappointing, and while they are much improved from the last *VB* review, they still let the product down.

VET

Scanning Speed

Hard Disk:

Turbo Mode (897.4 Kbytes/sec)	18 secs
Secure Mode (362.1 Kbytes/sec)	44 secs

Floppy Disk:

Turbo Mode (62.1 Kbytes/sec)	5 secs
Secure Mode (31.0 Kbytes/sec)	10 secs

Scanner Accuracy

'*VB* Standard' Test-set[1]	Turbo	354/364
	Secure	362/364
'InThe Wild' Test-set[2]	Turbo	113/116
	Secure	113/116
'MtE' Test-set[3]	Turbo	1438/1536
	Secure	1528/1536

Technical Details

Product: *VET Anti-viral Software*

Version: 7.2

Serial Number: Not stated.

Author: *Cybec Pty. Ltd.*, Suite 3 350 Hampton Street, Hampton, Victoria 3188, Australia.

Telephone: +613 521 0655

Fax: +613 521 0727

Price: $A90 for first PC, $A30 for each additional PC.

Test Hardware: All tests were conducted on an *Apricot Qi486* running at 25Mhz and equipped with 16MB RAM and 330MB hard drive. *VET* was tested against the hard drive of this machine, containing 1,645 files (29,758,648 bytes) of which 421 were executable (16,153,402 bytes) and the average file size was 38,370 bytes. The floppy disk test was done on a disk containing 10 files of which 6 (310,401 bytes) were executable.

For details of the test-sets used please refer to:

[1] Standard test-set: *Virus Bulletin* - May 1992 (p.23)

[2] 'In The Wild' test-set: *Virus Bulletin* - January 1993 (p.12)

[3] 'MtE' test-set: *Virus Bulletin* - January 1993 (p.12)

CONFERENCE REPORT

Pascal Lointier

ICVC '93 - Virus Hunting in Bulgaria

On the 5th of April this year, anti-virus experts from around the world arrived in Varna, Bulgaria, to attend the first ACM conference on computer viruses. Theconference lasted 4 days, and more than 100 participants attended, representing a wide cross section of both users and researchers.

Not surprisingly, the conference had generated a significant amount of interest among the press. The venue had more than its fair share of attendees from television, local or national radio stations and international press agencies.

Agenda

The organizers of the conference had two main objectives: to help national users to fight against viruses (not every Bulgarian is a virus writer!), and to provide a technical source of information about this sensitive topic.

During many of the sessions, the need for a better relationship between various institutions was emphasised. Everyone remembered the massively hyped stories about Bulgaria which were published around 1990 - tales of lethal viruses spreading throughout the world and an approaching global calamity. The lack of communication between some of the virus research centres and the media has caused many problems - this situation needs to be improved.

Nobody attempted to deny the existence of sophisticated Bulgarian viruses, but many criticised the way the Bulgarians had been type-cast as computer hackers hoping to spread their foul seed all over the world.

Hot Debate

Two of the speakers were Mr E. Nikolov, chairman of the Laboratory of Computer Virology of the Bulgarian Academy of Sciences at Sofia, and Mr V. Habov, co-author of a book about the 'Bulgarian Connection'. Nikolov's team consists of approximately ten people, who are responsible for studying computer viruses and advising users on methods of protecting themselves from virus attack.

Unfortunately, not all of those giving papers and talks were there in person. For instance, the paper by Sara Gordon had to be read by someone else, because she was not able to attend the conference. Indeed many of the familiar faces from the anti-virus community were missing.

The question on everyone's lips: Who is the Dark Avenger? More importantly, did he attend the conference...

Nevertheless, we enjoyed the Internet connections during the coffee breaks. Various chats took place either with well-known specialists or with members of the computer underground... from other countries.

As many readers may suspect, yes, there was at least one virus writer who gave a presentation. An interesting panel session followed where hot topics were discussed: should we condemn virus writing or just virus spreading? Is piracy the major factor of contamination in Bulgaria? How could a country be held responsible for the action of small minority?

As is always the way with such meetings, many of the most useful discussions took place in the evenings. Delegates were given plenty of chances to meet up outside sessions, as we were treated to a cocktail party one evening, and the official conference dinner the next.

Summing Up

Summing up the end session, participants agreed the following resolutions:

- To study legal and insurance matters, taking advantage of the experience in western countries.
- To establish the conference as an annual meeting dedicated to anti-virus fighting and computer security.
- To find a way to support the Laboratory of Virology in its efforts to help users.

The conference was marred only by the lack of international attendance. However, for all who came, it was a chance to evaluate the so-called 'Bulgarian Connection' first hand, and to make up one's own mind on how these complex problems should be solved.

END-NOTES AND NEWS

Sophos Ltd* has announced the release of an *OS/2* version of its scanner, *Sweep. Dr Jan Hruska, Technical Director of Oxford-based *Sophos*, commented 'The new version of *Sweep* puts the security function of virus detection in the hands of the network supervisor, rather than the users. The process can function in the background and is completely transparent.' Tel. 0235 559933.

Massachusetts Governor, William F. Weld, has proposed **new legislation to combat computer crime** in the state. The measure proposes a maximum penalty of five years in prison and a fine of $50,000 for theft of commercial computer services.

***Central Point Software* has announced 'Safe Six'**, an upgrade service for users of *MSAV*. While 'Safe Six' expands the number of viruses *MSAV* detects, it will not modify the features of the program. 'Safe Six' costs £39.99+VAT for a total of three updates. Tel. 081 8481414.

S&S International has announced *Dr Solomon's Anti-virus Toolkit for NetWare*. Competitors already selling well-established anti-virus NLMs will be intrigued by the *Toolkit*'s claim to be 'the first complete protection package on the market for servers and workstations using *Novell* networks.' Further information from Pat Bitton. Tel. 0442 877877.

Congratulations to *S&S International*, for winning a **Queen's Award for Technological Achievement** for its flagship product, *Dr Solomon's Anti-virus Toolkit*, and the techniques inherent in its development. 'To say that we are delighted to have this award conferred on *S&S* would be the understatement of the decade', Doctor Alan Solomon commented.

Reports are coming in of a bug in *MS-DOS 6* which can cause system instability under *NetWare 3.11*. A text file supplied with DOS 6 states that those affected should upgrade their Network shell to version 3.22, but some users claim that this does not solve the problem.

The White House has unveiled a new 'phone scrambling device which law enforcement agencies can tap. The 'Clipper' chip is designed to help provide secure telephone communication for legitimate purposes, while letting the US Government tap the lines of drug smugglers and terrorists. The US Government will maintain a database of all the chips manufactured and the keys which they contain. Privacy-paranoid Americans are known to be less than happy with this development.

While many would agree that the laws regarding computer security are somewhat lax, few would sanction the draconian measures imposed in China, where a man accused of computer hacking and embezzling $192,000 **has been executed**. The local news agency said that 'the crime was the first case of bank embezzlement via computer' in China. Hackers around the world will be enquiring anxiously about extradition arrangements with the People's Republic.

Patricia Hoffmans *VSUM* ratings for April: 1. *McAfee SCAN v102* 93.2%, 2. *Sophos Sweep 2.48* 90.7%, 3. *Frisk Software F-Prot 2.07* 89.2%, 4. *Dr Solomon's AVTK 6.04* 86.4%, 5. *IBM Anti-virus/DOS 1.0* 72.6%. **NLMs:** 1. *Sophos Sweep NLM 2.48a* 91.2%, 2. *McAfee Netshield V102* 89.4%, 3. *Intel's LanProtect 1.53+1/93S* 59.0%, 4. *CPAV/NLM 1.0* 56.4%.

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

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