ISSN 0956-9979 JUNE 1996

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Through the looking-glass.** *Windows 95* descended on the world last year with a media outcry reminiscent of a major discovery in the medical field. With it, inevitably, came the threat of viruses, followed by anti-virus software developed for the system. What is available, and how good are the products? *VB* has done an exhaustive series of tests: turn to p.10 for the whole story.

• **On being professional.** *NetPROT* has been reincarnated as *F-PROT Professional for NetWare*: an evaluation of *Command Software's* latest network baby can be found on p.26.

• **Yisrael Radai.** Just before going to print, *VB* learned of the death of Yisrael Radai, internationally recognised anti-virus researcher. Story on p.3.

CONTENTS

EDITORIAL

A Little Knowledge…

Readers will notice that the usual svelte figure of *Virus Bulletin* is this month distorted by the addition of eight pages. This is not due to some strange ailment, but to the *Windows 95* anti-virus products comparative review, an exercise which has taken a great deal of time and care, and has turned out to be well worth it. For the complete results, readers should turn to page 10; however, for now my mind turns to one particular facet of the review – the strange case of Stoned.Michelangelo.

"Windows 95 anti-virus products … have often not been written by people who have dealt with viruses before"

Michelangelo is perhaps the best-known computer virus of all time – since the hype-fest of 1992, almost every computer user has heard of it, and some are still wary of it. To illustrate its pervasive influence, an example: a friend, who only uses a PC for writing papers and sending email, sent me a message at the beginning of March last year, telling me that, as 6 March was Michelangelo Day, he would not be mailing me then; that he would not in fact even use his computer that day.

To return to the point, samples of Michelangelo are easy to obtain. Just about every half-cocked virus collection ever made public has had a disk image in it (or at the very least a boot sector); every anti-virus vendor will have a sample; there should be no problem with detection… right?

Wrong: in the *Windows 95* comparative, no fewer than seven of the sixteen products (over 40%) failed to detect it. My immediate reaction was that this was patently impossible, and that the sample must be damaged in some way and would not replicate. So I checked, and was surprised to find it intact and able to replicate without difficulty – in viral terms, a perfectly good sample. The only peculiarity was that it was on a 1.44MB diskette: this is significant because Michelangelo renders such diskettes unreadable by DOS. Attempts to access the disk result in a general failure error.

The problem is confirmed when we watch any of the seven products concerned attempting to scan the diskette: the drive clicks and whirrs, then the program informs the user that something is wrong with the disk, or (in extreme cases), that no disk is in the drive. The developers are of course aware that such a situation can arise with viruses; indeed, I was able to track down the DOS version of most of the products in question, all of which detected the same sample without problems.

This inconsistent behaviour can probably best be explained by looking at the pedigree of the two types of product: on the one hand are DOS anti-virus products, which have usually been maintained for several years; often by the same core team of anti-virus experts who originally wrote them; certainly by programmers with a good grounding in the ways of viruses. On the other hand are *Windows 95* anti-virus products, which have been released within the last twelve months, and (more importantly) have often not been written by people who have dealt with viruses before – and why should they? They use the same detection engine as the DOS product; their job is to make it compile under *Windows 95* and bolt a GUI on top. It probably wouldn't occur to them that standard *Windows 95* methods of accessing the disk are not enough to ensure the jobs get done under all circumstances.

This problem, on top of the fact that it is relatively unlikely that any quality assurance procedures which the company has in place will include such a test (though this will be changing about now, I would guess), means that this phenomenon is perhaps not as surprising as it at first appeared.

Further evidence, if any were needed, is provided by the unfortunate case of the *Windows 95* version of *Alwil's Avast* software: this product fails, in the version submitted for review, to detect boot sector viruses on disks without files. You can almost see the programmer's chain of thought: checking a diskette for viruses, what's the logical first step? See if there's anything on the disk to check! If the disk contains no files, it's blank and there's no need to check it. Later in the writing process, the programmer gets to the bit where he checks for boot sector viruses, but by this point he has forgotten his earlier assumption – and thus a bug is born.

The moral of the story? The first should be obvious: just because it is the same engine doing the virus detection does not mean the new product will do as well as the old. However, perhaps another would be a familiar refrain to the coders amongst us: never assume anything.

NEWS

Yisrael Radai

Just before *Virus Bulletin* went to print this month, staff were saddened to learn of the sudden death of Yisrael Radai, a long-standing member of the anti-virus community.

Radai was perhaps best known in recent years for his interest in the theory of integrity checkers and their uses against viruses. His paper on the subject. 'Checksumming Techniques for Anti-Viral Purposes', was presented at *VB '91*. The paper was updated in December 1994 and renamed 'Integrity Checking for Anti-Viral Purposes: Theory and Practice'. It remains one of the major works in the field.

Outside the specialist field of integrity checking techniques, Radai also wrote 'The Anti-Viral Software of MS-DOS 6' (1993), a detailed study of the security problems with the then-new *MSAV* and *VSafe* combination, which had earlier that year started to ship with *MS-DOS*. This paper has since spread across the Internet, and is an excellent guide to the delicate art of in-depth testing of anti-virus software.

Radai received an MSc in Computer Science from the Hebrew University of Jerusalem in 1975, but even before that he was on the staff of its Computation Centre. His first involvement with computer viruses came with a virus attack in January 1988. Prior to his death, he and the Editor of *VB* were engaged in a long-running discussion on subjects ranging from anti-virus product testing, to writing World Wide Web pages, to fractals. He will be sadly missed ▮

Scary Monsters and Super Creeps?

A recent release from *Reflex Magnetics* describes how, at May's *InfoSec* show in London, the company challenged people to break their *Disknet* anti-virus system by 'a *Reflex Disknet*-protected PC with a computer virus'. The prize for managing this was a Jeroboam of champagne.

According to the press release, an unnamed individual defeated the system by using a virus which 'uses a hitherto unknown way of activating'. The release goes on to say that *Word* has a 'security flaw … that allows this type of virus to activate without an auto macro'. Whilst little is known about this 'new technique', viruses which do not need auto macros to infect are not new: DMV is an example of the genre.

The virus author, states the release, offered to sell a key to 'unlock' the virus, which *Reflex* refused, saying they 'have no wish to encourage people to write new viruses'. An interesting statement, given that, as the virus 'went undetected', one must assume the author was given the champagne… ▮

Stop Press: *Macintosh* users should refer to the enclosed insert for important information on the MBDF.A virus which has been distributed on the cover CD of the UK magazine *MacUser*.

Prevalence Table – April 1996

Virus	Type	Incidents	Reports
Concept	Macro	74	19.5%
AntiEXE	Boot	35	9.2%
Form.A	Boot	34	8.9%
AntiCMOS.A	Boot	29	7.6%
Parity_Boot.B	Boot	27	7.1%
Empire.Monkey.B	Boot	18	4.7%
Ripper	Boot	17	4.5%
Sampo	Boot	14	3.7%
EXEBug	Boot	10	2.6%
Junkie	Multi	10	2.6%
NYB	Boot	10	2.6%
Quandary	Boot	10	2.6%
WelcomB	Boot	7	1.8%
Telefonica	Multi	6	1.6%
Stealth_Boot.C	Boot	5	1.3%
Stoned.Angelina	Boot	5	1.3%
Empire.Monkey.A	Boot	4	1.1%
Feint	Boot	4	1.1%
Jumper.B	Boot	3	0.8%
Manzon	File	3	0.8%
SheHas	Boot	3	0.8%
Stoned.Standard.A	Boot	3	0.8%
Tentacle	File	3	0.8%
V-Sign	Boot	3	0.8%
Boot.437	Boot	2	0.5%
BootEXE.451	Multi	2	0.5%
Bye	Boot	2	0.5%
Colors	Macro	2	0.5%
Da'Boys	Boot	2	0.5%
Form.D	Boot	2	0.5%
J&M	Boot	2	0.5%
Stoned.Manitoba	Boot	2	0.5%
Stoned.NoInt	Boot	2	0.5%
Swiss_Boot	Boot	2	0.5%
Unashamed	Boot	2	0.5%
Other [1]		21	5.5%
Total		380	100%

[1] The Prevalence Table also includes one report of each of the following viruses: AntiCMOS.B, Barrotes.1310, Burglar.1150, Cascade, Defo, Delight, Die-Hard.4000, DiskWasher, FatAvenger, Geek.734, Hot, Imposter, Int40, Jerusalem.?, One_Half, Shirley, Stoned.Dinamo, TaiPan.438, TPE:?, Yankee_Doodle.44.A, and W-Boot.

IBM PC VIRUSES (UPDATE)

The following is a list of updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 21 May 1996. 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 a dedicated scanner which contains a user-updatable pattern library.

Type Codes

- **C** Infects COM files
- **D** Infects DOS Boot Sector (logical sector 0 on disk)
- **E** Infects EXE files
- **L** Link virus
- **M** Infects Master Boot Sector (Track 0, Head 0, Sector 1)
- **N** Not memory-resident
- **P** Companion virus
- **R** Memory-resident after infection

NOTE: The template given in May's IBM PC Virus Update Table for the virus Alfons.1344 was inaccurate; the correct template reads as follows:

```
Alfons.1344     B436 83E2 1FCC 40C3 FC1E 06B4 52CD 2133 ED26 8B57 FE8E DA80
```

Alfons.1536 **CER:** A variant of Alfons.1344. In EXE files, it is appending and 1536 bytes long; in COM files, it is prepending and 1618 bytes long. It contains the encrypted text: 'COMMAND COM' and 'Synchronizing drive C:(do not interrupt this operation! 0% Done'. During the first week of a month which starts on a Sunday, the virus may overwrite the first hard disk.

```
Alfons.1536     83E2 1FB4 36CC 40C3 FC1E 06B4 52CD 2126 8B57 FE2E 8916 1205
```

AOS.833 **CER:** A stealth, encrypted, appending, 833-byte virus containing the text: 'AnGrY OoD ShOt 3 ViRuS'.

```
AOS.833         92CD 1692 92B9 A001 BB?? ??2E 8107 ???? 83C3 0283 E901 75F3
```

AOS.847 **CER:** A stealth, encrypted, appending, 847-byte virus containing the text: 'AnGrY OoD ShOt 2 ViRuS'.

```
AOS.847         5992 CD16 9292 9292 B9A7 01BB ???? 2E81 07?? ??83 C302 83E9
```

AOS.854 **CER:** A stealth, encrypted, appending, 854-byte virus containing the text: 'AnGrY OoD ShOt 1 ViRuS'.

```
AOS.854         9292 9292 92B9 AA01 BB?? ??2E 8107 ???? 83C3 0283 E901 75F3
```

AOS.Zlanted **EN:** A family of encrypted, appending, direct infectors. They all contain the text 'Zlanted 3/96' and '*.EXE'. The viruses set the BIOS variable which is responsible for the number of characters in a display line to 81.

```
AOS.Zlanted.736  5059 BA01 FAB8 4559 92CD 1692 9292 B96F 01BB ???? 2E81 2F??
AOS.Zlanted.744  BA01 FAB8 4559 92CD 1692 9292 9292 B973 01BB ???? 2E81 2F??
AOS.Zlanted.752  FAB8 4559 92CD 1692 9292 9292 9292 B977 01BB ???? 2E81 2F??
AOS.Zlanted.758  4559 92CD 1692 9292 9292 9292 9292 B97A 01BB ???? 2E81 2F??
```

Blue Nine.C **CR:** A stealth, appending, 925-byte virus which contains the plain-text message: 'It's only a lil lightfearing creature'. It is a slightly modified variant and can be detected with the template for Blue Nine.B [*see VB, September 1995, p.4*].

Blue Nine.1725 **CR:** A stealth, appending, 1725-byte variant containing the text from the 'I can't be with you' song by the Irish folk-rock group the Cranberries. This variant can detected with the template for Blue Nine.A [*see VB, September 1995 p.4*].

Body.884 **CER:** An encrypted, appending, 884-byte virus which contains the text: 'BODYBUILDING!' and 'OpalSoft3'.

```
Body.884        E800 005D 83ED 03B9 6103 8BFD 2EF6 5513 47E2 F9AF ADA9 E1F9
```

Compiac.379 **ER:** An appending, 379-byte virus which installs itself in the Interrupt Vector Table. It contains the text: 'COMPIAC'.

```
Compiac.379     B875 77CD 2102 C075 52B8 02FA BA45 59CD 16B4 02CD 1AFE C5B4
```

Eliza.1282 **CN:** A prepending, 1282-byte direct infector, based on the original Eliza virus (which was 1193 bytes long). The virus contains a destructive payload which triggers on Friday the 13th. It displays the following message, usually encrypted: '++ Hi! I am Venus. Good Luck ++'.

```
Eliza.1282      0200 5E81 C600 01BF 0001 5951 56AC AAE2 FC5F 5932 C0AA E2FD
```

HLLO.5520 **EN:** An overwriting, 5520-byte virus. It hides all EXE files in the current directory by setting their hidden attributes, and displays the text 'Seek and Distroy -Zalman viruS-'. The virus contains another plain-text message: 'Portions Copyright (c) 1993,94 Zalman 3'.

```
HLLO.5520       6E62 6F21 776A 7376 542E 9A00 0081 0055 89E5 B800 069A 7C02
```

Karol.1000

ER: An appending, 1000-byte virus which contains the plain-text string: 'KaRoLmArKs V2.3'. It incorporates some anti-debugging techniques (e.g. redirecting Int 01h). Two minor variants are known.

```
Karol.1000.A        33F6 8EDE 8EC6 BE10 00BF 0100 8736 0000 873E 0200 B452 CD21
Karol.1000.B        33F6 8EDE 8EC6 BE50 00BF 0100 8736 0000 873E 0200 B452 CD21
```

L&D.683

CR: An appending, 683-byte virus containing the text: 'Love & dedication to D.I. - 1992/93. - Croatia,VZ' at the end of infected files. The string 'Death' is found near the beginning of each infected file.

```
L&D.683             A108 0126 A386 0026 C706 8400 7301 26A3 7200 26C7 0670 00BD
```

MeiHua.1786

CER: An appending, 1786-byte variant of MeiHua.1826 (1819). It contains this encrypted message, which may be displayed when the SCAN program is executed: 'AntiVirus If your software has been infected by other viruses, run your software, then the virus will be cleaned ! THANK YOU! —Mr. MeiHua—'.

```
MeiHua.1786         E88B 042E C706 4005 4000 9C58 0D00 0350 9D90 9090 9090 9090
MeiHua.1826         E89D 042E C706 6105 3E00 9C58 0D00 0350 9D90 9090 9090 9090
```

Phalcon.1125

CN: An appending, 1125-byte, fast, direct infector which contains a destructive payload: on the second day of any month it tries to overwrite the contents of the hard disk. It contains the text: 'Immortal Riot' and 'Maria K lives..Somewhere in my heart..Somewhere in Sweden..I might be insane..&But the society to blame.. (The Unforgiven / Immortal Riot'.

```
Phalcon.1125        B402 99B9 0001 CD26 E900 00FA B003 B9BC 02BA 0000 8E5D 638B
```

Pretentious.680

CN: An appending, 680-byte direct infector which contains the plain-text string: '* Gdynia 1996 * v1.0 *'. Additionally, the virus contains the encrypted text '*.COM' and this displayed message: 'Windows 95 may be dangerous. OS/2 is the best operating system! I'll prove it soon...'

```
Pretentious.680     B95C 002E 8A07 32C1 2AC1 2E88 0743 E2F3 B409 CD21 61C3 602E
```

PSMPC.313

CN: An appending, 313-byte direct infector which contains the string '*.COM' at the end of infected files. The virus contains a procedure which reprograms the colour scheme of the video card.

```
PSMPC.313           B43F 80C4 01B9 3901 8D96 0401 CD21 B43E CD21 B440 80C4 0FEB
```

Redhack.1405

CR: An encrypted, prepending, 1405-byte virus which contains the text: '(c) Red Hacker, Zielona Góra'. The virus does not infect COMMAND.COM and WIN.COM.

```
Redhack.1405        EB03 1A?? ??B9 8C00 BE10 0146 8034 ??E2 FA?? (in files)
Redhack.1405        3D00 4B75 03E9 8E00 3D00 3D75 03E9 9400 80FC 4375 03E9 8600 (in memory)
```

Retail.1536

EN: An appending, 1536-byte, fast, direct infector which contains the plain-text strings: '*.*' and '*.exe' It looks for its targets on the C: drive, and contains a procedure which will corrupt the CMOS data. The file-size information in the EXE-header differs from the size specified in the directory entry – it is one page shorter.

```
Retail.1536         2ACD 2180 FA0E 7546 B012 E670 EB00 B00E E671 B033 E670 EB00
```

SillyC.138

CN: An appending, 138-byte virus which contains the text '*.COM'.

```
SillyC.138          2C03 8945 0AB4 40BA 80FF 01FA B18A CD21 B800 4231 C931 D2CD
```

SillyC.174

CN: An appending, 174-byte virus which contains the text '*.COM'.

```
SillyC.174          2D03 0089 450B B440 BA5D FF01 FAB1 AECD 21B8 0042 31C9 31D2
```

SuicidalDream.847.A

CN: An appending, 847-byte direct infector which contains the following plain-text messages: '[TU.Suicidal.Dream.B](c) 1996 The Freak/The UndergroundFrom the hypnotic spectre of wake I screamLocked in the depths of a Suicidal Dream', '.com *.zip anti-vir.dat', and 'Happy Birthday Freaky!'.

```
SuicidalDream.847.A 2E8B 8EE9 032E 8B86 2404 81C1 5203 3BC1 74BE 2D03 002E 8986
```

SuicidalDream.847.B

CN: Almost identical to variant 847.A. The only difference is one character in the virus message: 'happy Birthday Freaky!'. Both variants can be detected using the same template.

Tanpro.524

CER: A prepending, 524-byte virus which contains the plain-text signature '(c)tanpro'94'.

```
Tanpro.524          601E 063D 004B 7403 E91A 01FC 8CD8 8EC0 8BFA B000 B9FF FFF2
```

Trivial.OW.284

CN: An overwriting, 284-byte virus which contains the texts '*.COM', '\DOS', '[Poopie/MDK]' and '????????COM'.

```
Trivial.OW.284      BA0E 02CD 218B D8B4 40B9 1C01 BA00 01CD 21B4 3ECD 21CB 0EE8
```

V.1458

CER: An appending, 1458-byte virus which contains the encrypted string 'COMMAND.COM.EXE'. While infecting COMMAND.COM, the virus overwrites its last 1458 bytes (usually filling with zeroes).

```
V.1485              B877 4BCD 213C 7875 48E9 C900 B462 CD21 8EDB 8EC3 A102 0080
```

V.773

CR: An appending, 773-byte virus. It contains a payload which may delete some files, other than .386, .COM, .DLL, .EXE, .OV?.

```
V.773               0300 50B8 0040 B905 0331 D2CD 2158 72C4 50B8 0042 31C9 31D2
```

Z-90.1500

CER: A stealth, prepending (COM) or appending (EXE) 1500-byte virus. It contains a destructive payload: in July and December, when the system internal clock shows 9:03, it encrypts the contents of the first cylinder (all sides, all sectors) of the first hard disk. Then, it displays the following, usually encoded, message: 'VIRUS Z-90. JUNIO 95. MEXICO SE MURIO TU DISCO DURO TU LO PUEDES REVIVIR HAS BAILADO ALGUNA VEZ CON TU NOVIA A LA TENUE LUZ DE LA LUNA?'

```
Z-90.1500           9C80 FC2B 750A 80FA CC75 05B8 FF33 9DCF 3D00 4B74 7080 FC3D
```

INSIGHT

The Road is Long…

…with many a winding turn: cryptography, military service, anti-virus research. The road *is* long, and there are certain to be many more turns in it for Padgett Peterson.

His 'first steps' in computing happened at a very early age; seeing a Mark II in operation while his father was teaching at Annapolis Military College: 'My first real memory of computers was playing tic-tac-toe on a *Univac* in '56 or '57.'

Peterson began as an automobile mechanic: a self-confessed fanatic, at one stage he owned seven Jaguars. He started his professional life as a *General Motors* graduate in Mechanical and Electrical Engineering. From there, he worked for contractors for the Department of Defense, where he assisted in designing engine and flight controls for the military. His next step was in LAN topology for the *FAA* National Airspace Plan – all this before a hobby, anti-virus research, became a career.

Footsteps in Time

The 1960s and 1970s saw Peterson gaining a wealth of experience on various machines: the *1401* circa 1962, *SEL 800s* and *Honeywell 601s* between 1967 and 1969, the *IBM 360* in 1971/72, and then *DEC PDPs*.

'I first "flew" an afterburning turbine engine on a testistand using an 8080-based controller in 1976, two years before DEEC was announced. The first real computer I used was a *VAX-11/780* (I think the serial number was 2 or 3!) in 1979.

'I wrote a terminal emulator program around 1981 which allowed me to use a *VAX* to dial into a similar system: it was at 1200 baud, and upload took forever – but it beat taking an airplane up there! I bought a *Columbia VP-1600* in 1983 (I had a couple of *Sinclair* toys earlier, but the PC was the first to come close to the capability of the *VAX*) and I have not needed to migrate since. I also have a *Mac SE/30* at home, but just for experimenting.'

Peterson finds it difficult to remember when computing became his career, but estimates it to have been sometime before 1962 – 'And there really was no training then. If you had access, you just did a lot of playing.

'Mathematics courses in college,' he explained further, 'were the biggest help, mainly in teaching a person how to think – boolean logic, statistics, linear programming (not what it sounds like!) and set theory were probably the most helpful to me.'

He still recalls the final exam for a training course presented by *Eagle Star* on their EPTAK programmable controller, which he took in 1975: 'It was to make a bulb flash. Passing was being able to do it in twelve instructions. The engineers had done it in nine… I made it flash in seven, and got chided for not initializing, but my defence was that initialization did not matter, since all that was needed was a state change – the rules did not specify where to start.'

Coming Full Circle

Peterson's first paying job was a stint with 'Uncle Sam' (the US Military), which included, as he described it, a 'free vacation' in south-east Asia.

'I took the aptitude tests, and did well in everything except cooking!' he chuckled. 'I decided on electronics, but ended up going to the school of cryptography, where I learned all there was to know about KW-26 and KG-13 – early cryptographic devices used with teletype machines.

'My first assignment was at Patrick AFB (relatively close to my home), which serviced Cape Kennedy – I travelled all over the Caribbean before I left to join the Air Force Security Service in south-east Asia.

'So, I learned crypto gear back when they had lots of little vacuum tubes. I also learned electronics; in fact, I still need an oscilloscope to think. I spent a few years doing that, but eventually returned to cryptography: I can now document thirty years of *paid* cryptographic experience, and forty years in computers – counting the games… Not bad,' he concluded, 'for a youth of 31h…!'

Peterson feels that the initiative in the cryptographic field has now passed into the commercial sector, quoting as an example the SET (Secure Electronic Transaction) specification for the credit cards Visa, AMEX, and MasterCard.

In his opinion, good encryption has been around for years: 'The big problem is, and always has been, key management. I seem to be one of the few people in favour of key escrow – but of course, only if I (or my employer) hold the keys!

'There is a problem with the US International Trade in Arms Regulation, a law which regulates cryptography as if it were a munition: this makes it illegal to export it without a licence. It is, however, only a minor problem, which I expect will soon go away.'

Virus Checks and Cures

The development of an integrity management program to combat low-level viruses is another of Peterson's achievements: *DiskSecure* loads from the Master Boot Record, and a 288-byte TSR resides in low memory as DOS or *Windows 95* loads: 'It's designed not to need updates,' he explained, 'and hasn't had any since 1993, though it has caught many new viruses since.'

Peterson has also developed a heuristic scanner for boot sector viruses, which, when it was developed a few years ago, was the only one to catch all the viruses in the then current Virus Test Centre collection.

He views the product, however, as something of a dead end, due to the fact that the possibility still exists of creating a virus undetectable using heuristics: 'I believe,' he said, 'that integrity management is the only real answer. Emulation might be effective in the short run, but this too will ultimately be a dead end.'

Of the newer threats, Peterson rates the current spate of macro viruses as less of a potential problem than Java viruses will be: 'You have seen what the macro viruses are doing, and that is limited to those who have *Word*,' he commented. 'In the near future, all successful Web browsers will run Java and, like this, will create a cross-platform environment like *Word*. Hopefully, *Sun* will learn from the *Word* experience, but today, who knows?'

It would be difficult, Peterson feels, for a new developer to write from scratch any anti-virus program which would be capable of accommodating all the new platforms: 'I keep meaning to learn to write a VxD,' he said, 'but haven't had time to do it yet. The other problem around at the moment has to do with the lack of available documentation on programs like *Word*.'

"virus writers are not to blame for the target-rich environments which make the spread of viruses possible"

On Legalities and Further

Whether or not the writing and/or distribution of viruses should be legal, in Peterson's eyes, cannot be approached from a personal viewpoint: 'Legal issues should be confined to those who distribute, or fail to protect against, viruses,' he clarified. 'For years I have been using the words "due care" and "culpable negligence". Virus writers are not to blame for the target-rich environments which make the spread of viruses possible.'

That being said, Peterson does not in any way condone their activities. The only contact he has had with virus writers was indirect – two self-professed 'reformed ex-virus writers', Patrick Tolme and Mark Washburn, frequented the *McAfee* BBS, Homebase, when that was the 'hang-out' for many anti-virus people.

Despite these indirect encounters, Peterson has never had the slightest inclination to experiment with writing viruses himself; however, he confessed to being appalled by the sloppy programming skills he has seen displayed by most virus authors.

At one stage, he even went so far as to determine that mistakes in one or two viruses could be repaired to make them viable, but was never tempted to pass on the knowledge he had acquired.

'I have never seen anything done,' he went on, 'or purported to be done by a virus that I could not do with more reliability and less complexity without one. I find what makes a virus a virus (the propagation) boring.'

Homing In on Things

Peterson is married, and has one son, Geoffery, aged 23, who plans to be an orthopaedic dental surgeon. Geoffery too has inherited a passion for computers: he has been known to build LANs in his spare time, according to his father.

His wife also spends a considerable amount of time working with computers: her great passion is genealogical research, large amounts of which she carries out on the World Wide Web: 'She was on the Net over 150 hours last month,' recounted Peterson. 'Fortunately, her service is flat rate for up to 250 hours.'

Most of Peterson's leisure seems, he says, to be spent on airplanes: when he has time, however, he likes to 'play with his Pontiacs'. He also has a collection of *Zenith* radios, including at least one of every model *Zenith TransOceanic* (portable shortwave radios) made during their production years, from 1942–1982.

'I do tend to collect things to a silly extent,' he admitted ruefully. 'I used to race Corvettes (I worked for *GM* at the time, so it was cheap) – when I got out of them in 1975, it took two semi-trailers and a flat-bed truck to haul it all off. Pontiac stuff is about as bad; I have four seven-litre engines sitting on dollies in the garage right now.'

Peterson's interests do not stop with cars and cryptography: he can also claim ownership to *SCCA National*, *FIA*, and *IMSA* licences ('Automobile competition licences for road racing,' Peterson clarified), and has qualified for the Society for Philosophical Inquiry.

Conclusion

Peterson lives modestly, by his own account, in his 2300-square foot home in Orlando, Florida: 'It's warm,' he explained, 'there's no state income tax, and we have a world-class airport.

'I'm fortunate not to need a lot of sleep,' he continued. 'I'm often at my most productive after midnight. If I have a major peeve, it's having to spend an hour a day commuting to work. I feel that this is "lost time", that could be better spent on other things.

'Really, when I think on it, it is quite remarkable how fortunate I have been to be in places where Things Were Happening for so long. I do not expect to stop anytime soon, either – I'm just having too much fun!'

VIRUS ANALYSIS

CNTV – New Technology

Eugene Kaspersky

The number of problems facing anti-virus researchers grows from year to year; from link viruses, through ever more advanced polymorphics, to the current fad for macro viruses. Their authors produce new ideas, which are later polished and improved with the speed of workers on a production line. They allow us no virus-free time; in fact, quite the reverse – their new ideas continually lengthen the list of unsolved virus problems.

The latest fad is for writing 'entry-point obscuring' (EPO) viruses, which shun the standard technique of placing a JMP to their code at the head of COM files and modifying the entry point values in the headers of EXE files. There are about ten of these: the first, which appeared about three years ago, was Lucretia.

Entry Point Obscuring Viruses

At the time, Lucretia was the only one of its genre, and as I remember it, anti-virus researchers saw it as a type of challenge. The virus was inevitably only the first in a long line. Late 1995 and early 1996 brought more: three examples of Zhengxi [*VB, April 1996, p.8*], one of Positron [*VB, February 1996, p.8*], two Mid-Infectors, and lastly Markiz and Nexiv_Der [*VB, April 1996, p.11*]. There may be others.

All these viruses patch themselves into the middle of the file: here they write a JMP instruction or part of the virus code, either of which will pass control to the main virus body. If the virus is encrypted, it may be decrypted by the code section inserted in the middle of the file, or by the main body, which is usually at the end of the file.

EPO viruses use various methods to decide where to place the patch within the file. The techniques so far seen are:

- look for specific C/Pascal subroutine headers: MOV BP,SP; PUSH SP (Lucretia and Zhengxi).
- disassemble the file code (Mid-Infector)
- load the file into memory, and trace its execution (Nexiv_Der)
- wait for the first Int 21h call, and overwrite the code which performed the call (Positron and Markiz)

There are, of course, other viruses which do not modify the start of COM files, or the entry point fields in EXE and SYS file header – for example, Willy and Lapiddan. Willy infects only COMMAND.COM, and patches it at different offsets, according to which version of that file is present. Lapiddan writes itself to the end of SYS files, but does not modify the Strategy and Interrupts fields; instead, it alters the field containing the address of the next driver in the sequence of drivers within the same file. Both viruses infect specific files, and have no JMP to the virus code.

In the case of unencrypted viruses, detection is not difficult: the solution is simply to move to the end of the file, read it, and look for the virus there. Where polymorphic viruses are concerned, the problem becomes much more complex. Unfortunately, as the number of such viruses grows, they become more devious: CNTV is one of the new generation.

The Infected File

In an infected file, CNTV is divided into two blocks of code, which are placed in the file at different offsets. The first block contains the decryption routine, and is placed at a randomly-selected offset in the file. The second block is placed at the end of the file, and contains the (encrypted) main body of virus code.

Execution

When a file is executed, the flow of execution will eventually bring control to the first block of virus code. Because the virus does not patch the file to pass control here immediately, this is in some ways nondeterministic. However, if and when the first block gains control, it decrypts the second block and passes control to it.

CNTV's first action is to restore the code of the host file. At the time the first block of code was inserted (i.e. when the file was infected), it overwrote part of the host file – this must now be patched back in memory. The virus then checks which version of DOS is being used, and returns control to the host file if it is version 3.0 or earlier.

If the version of DOS is sufficiently recent, the virus proceeds to call the undocumented DOS function Int 21h, AH=52h (Get List of Lists). It uses the data returned from this call to calculate the location in memory of the lowest disk buffer. It checks this to see if it is already resident (it looks for the first two bytes of its own code, which are 50h and 53h). If not already resident, it modifies the disk buffer chain to remove the first five buffer areas, and overwrites them with virus code.

The virus uses the same code to manipulate buffers under all different DOS versions. It works without problems on all versions of DOS except 4.x: the virus cannot infect a DOS 4.x system.

The fact that the virus places its code into the disk buffer area means that the virus will not always go resident at the same memory location – if DOS has been loaded high, the virus will be found in the HMA; otherwise, it will be in conventional memory.

Once the virus has installed itself into memory, it hooks Int 21h. The virus searches for the memory block occupied by the resident copy of COMMAND.COM, and writes a 22-byte long routine which passes control to the virus code there – it would seem that this routine is meant to confuse the expert user searching for the virus in his system. Finally, the virus returns control to the host program.

Infection

CNTV intercepts only one DOS function, Load and Execute (Int 21h, AX=4B00h), and infects files when they are executed. The virus first checks the file name, and does not infect it if its name matches one of the patterns *MAN?.* (COMMAND.COM), *CV??.*, *SCA?.* (SCAN.EXE), *LEA?.* (CLEAN.EXE), *UAR?.* (*GUARD.EXE), or *IEL?.* (*SHIELD.EXE). If the filename matches one of these strings, it is allowed to execute normally.

Otherwise, the virus loads the file into system memory using Int 21h, AX=4B01h (Load But Do Not Execute). This call is more usually used by debuggers: considering what the virus is about to do, the use of this function is not surprising.

It then opens the file and reads the header to determine whether the file is of COM or of EXE format. If it is an EXE file, CNTV calculates the length of the EXE header, and checks the stack pointer held within it to prevent stack overlap when an infected EXE file is executed.

The virus also checks the length of the file, and does not infect small (less than 3K) or huge (over 400K) files. If the file is within these limits, the virus hooks Int 01h (Single Step), and passes control to the image of the file already in system memory: from this position, CNTV is able to trace the execution of the file.

Tracing

The virus' Int 01h handler is invoked after every instruction in the target file is executed. This handler takes the pointer to the code being traced from the stack, and examines the code to which it points.

First, to avoid multiple infection, the virus checks the code against the two bytes of its entry block – if they are 9Ch and 0Eh, the file is seen to be already infected, and infection is abandoned. If these two bytes are not present, the virus skips approximately 100 assembler instructions in the traced file (exactly 100 if there are no Int 21h calls or string instructions), then calculates and stores the corresponding disk offset in the file.

CNTV next compares the images of that part of the executable file which is to be patched on the disk and in memory. This action is necessary to prevent damaging, through misinfection, packed files and EXE files containing relocated addresses. Were the virus to continue the infection process on such files, the chances are high that those files would become corrupted.

Next, the virus writes its first block of code to the target file, at the offset calculated in tracing. It then encrypts the whole virus body and writes it to the end of the file. Infection complete, the virus allows the program to execute normally.

During infection, the virus uses the block of Video Memory starting from address B800:3400 or B000:3400 (depending on whether the current video mode is standard text or Hercules) as a read/write buffer. It then hooks Int 24h to prevent any DOS error messages appearing if it tries to infect a file on a write-protected disk. The virus also obtains and restores the file's date and time stamp, and clears the read-only attribute prior to infection; however, it does not subsequently restore the latter.

Trigger Routine

Whenever the virus infects a file, it stores within that file a number which represents the day on which that copy of the virus is due to trigger. This number is the current date either plus 14 or plus 28. Of course, the virus adjusts the trigger day to ensure that it is less than 31.

Once installed in memory, the virus checks the system date and time on any DOS call except Int 21h, AH=4Bh (a superset of that used for file infection). It executes the trigger routine if the virus went resident in memory more than about an hour before, if the current date is September 1995 or later, and if the current day is a 'trigger day'.

Under these conditions, the virus displays this message:

```
¡ A CuBaN NeW TeChNoLoGy ViRuS By SoMeBoDy !
```

CNTV

Aliases:	None known.
Type:	Memory-resident, parasitic COM and EXE file infector, encrypted but not polymorphic, entry-point obscuring virus. 2630 bytes long.
Self-recognition in Files:	Checks file code during tracing.
Self-recognition in Memory:	Checks the system buffer area for the presence of virus code.
Hex Patterns in Files and in Memory:	`9C0E E800 0051 1E56 8BF4 368B 7406 2EC6 4426 E2EB 008C CE81 C6?? ??8E DEBE ???? B923 051E 5681 34?? ??46 46B4 F8CB`
Trigger:	Depending on system date and time, a message is displayed.
Removal:	Under clean system conditions, identify, delete and replace infected files.

COMPARATIVE REVIEW

When I'm Cleaning Windows

It is now fast approaching one year since the release of *Windows 95*, with all its associated hype and advertising frenzy, played out on international TV to the accompaniment of Rolling Stones hits. Well, one Rolling Stones hit.

However, beta versions of *Windows 95* were available to anti-virus companies many months before its full release, and many developers had products ready to ship soon after. Special mention must be made of *Symantec*, whose *Norton AntiVirus for Windows 95* (along with *Utilities* and *Navigator*) was on the shelves of US software stores several days *before* the operating system required to run them.

Windows 95 – Not Quite Everywhere Yet…

Eventually, *Microsoft* hopes, *Windows 95* will push *Windows 3.1* off the desktop entirely. Already we see new versions of applications only being released for *Windows 95*, a trend which can only be expected to continue.

However, corporates are understandably still reluctant to reinstall software on their desktop PCs. They see no reason to – after all, *Windows 95* does not yet have its killer application (an application so useful that people will upgrade just to be able to use it). This will change in time.

The more relevant question is not 'will people upgrade from *Windows 3.1*?', but 'when they do, what will they upgrade to?'. This is not as bizarre as it sounds. *Windows NT Workstation 3.51* only requires marginally more powerful hardware than *Windows 95*, and the former is more fully featured and stable than the latter. Nevertheless, right now, *Windows 95* is a credible system for the desktop.

Testing

Manufacturers were asked to submit the anti-virus product they would sell to a customer who wanted to protect a *Windows 95* system. This does not necessarily mean a *Windows 95* native product; however, as expected, most of the products submitted were GUI-based 32-bit applications.

There were three major tests, against virus test-sets (see next section for information), a clean test-set, and a diskette speed test. The clean test-set consisted of 2,500 COM and EXE files, occupying 167,028,828 bytes in 50 directories. This provided both the clean hard disk scan time figures and the false positive results. In the floppy speed tests, products were run over two 1.44MB floppies, one containing 43 clean files (one directory, 997,023 bytes), and the other containing the same files, but infected with Natas.4744 (one directory, 1,201,015 bytes). This allows comparisons of a product's speed on infected and uninfected files.

Calculations

The percentages are calculated in the same way as for the last DOS scanner comparative. For detailed information, readers are referred to the last two DOS scanner protocols [*VB, February 1995, p.12; VB, November 1995, p.14*], copies of which are available to readers on request.

The scoring in the Polymorphic set is biased to favour those products which detect *all* samples of a virus. Scores in the other sets are calculated on a per-virus basis, and the percentages are combined to give the percentage for each set. This prevents the number of samples of a virus having too much effect on a product's score; however, it does mean that two products that each miss one of 300 samples will probably get different percentages: e.g. if one misses the only sample of a virus and the other misses one of six.

Viruses Used

This review divides the viruses used into four similar categories as on previous occasions – Boot Sector, In the Wild, Standard, and Polymorphic. Complete listings of the test-sets can be found at the end of the review.

A change has been made to the In the Wild test-set. Previously, this only contained file infectors; the Boot Sector set was separate. Now, however, the viruses in the Boot Sector set are also taken from the WildList, allowing an overall score to be calculated for In the Wild detection.

The Standard set remains unchanged since the NLM comparative review [*VB, April 1996, p.14*] – this will be revamped before the DOS scanner comparative next month. Finally, the Polymorphic set has been slightly enlarged – the addition of Natas.4744 sees it grow to 8000 infected files, made up of 500 samples of each of sixteen viruses.

Technical Notes on Testing

Testing *Windows 95* anti-virus products is considerably harder than testing the same products for DOS. Whereas almost all DOS products can be run from the command line, completely unattended, most *Windows 95* products cannot. The presence of the GUI renders automated testing, at least at this stage, highly impractical. Whilst it was possible to set up scheduled jobs for some products to fire at suitable times, more often than not the products had to be driven directly, and by hand.

The test machine was configured as for the macro comparative [*VB, May 1996, p.10*]. After testing a product, the OS was reinstalled from a sector-level image before installing the next product, thus ensuring all products were tested with the machine in the same configuration. Also, no hardware changes (e.g. increasing the amount of onboard memory)

	ItW Boot		ItW File		ItW Overall	Standard		Polymorphic		Overall
	Number	Percent	Number	Percent	Percent	Number	Percent	Number	Percent	Percent
Alwil Avast	68	97.1%	300	100.0%	98.6%	305	100.0%	6500	79.7%	94.2%
Cheyenne InocuLAN	68	97.1%	295	99.0%	98.1%	279	93.5%	5705	67.5%	89.3%
Command F-PROT	68	97.1%	300	100.0%	98.6%	288	96.6%	4916	53.9%	86.9%
Data Fellows F-PROT	68	97.1%	300	100.0%	98.6%	288	96.6%	4916	53.9%	86.9%
EliaShim ViruSafe	69	98.6%	279	94.1%	96.3%	263	88.7%	3042	31.6%	78.3%
ESaSS ThunderBYTE	70	100.0%	295	98.9%	99.5%	300	97.7%	7421	78.9%	93.9%
IBM AntiVirus	68	97.1%	299	99.5%	98.3%	300	98.2%	6424	69.6%	91.1%
McAfee Scan	69	98.6%	300	100.0%	99.3%	289	96.0%	5430	65.0%	89.9%
Norman Virus Control	63	90.0%	294	98.5%	94.2%	296	98.1%	7998	98.4%	96.2%
Norton AntiVirus (NAV)	70	100.0%	298	99.5%	99.7%	289	96.0%	4734	58.4%	88.5%
On Technology Doctor	67	95.7%	258	86.0%	90.9%	274	90.6%	2192	25.2%	74.4%
RG Software Vi-Spy	68	97.1%	288	96.5%	96.8%	297	97.7%	4758	52.4%	85.9%
Sophos' SWEEP	70	100.0%	297	99.2%	99.6%	305	100.0%	7998	98.4%	99.4%
Stiller Integrity Master	67	95.7%	296	98.9%	97.3%	293	96.3%	3268	36.9%	81.9%
S&S Dr Solomon's AVTK	68	97.1%	300	100.0%	98.6%	305	100.0%	7488	90.5%	96.9%
Trend PC-cillin	67	95.7%	298	98.8%	97.2%	269	91.1%	5337	61.0%	86.6%

Table 1: Some products came close to scoring an overall 100% in this comparative review, but no-one quite made it.

were made during testing, as such a change would invalidate the speed figures. Hardware specifications on the test machine can be found at the end of the review.

Another problem arose from the number of boot sector viruses. With sixteen products and 70 boot sector samples, there were 1120 disk swaps. Currently, however, there is no workaround (other than extreme patience).

Within the review are two tables and four graphs. The graphs contain no extra information over and above that in the tables, but make that data easier to interpret. Also, virus detection results are reproduced in each product's section. False positives, where encountered, are mentioned there as well.

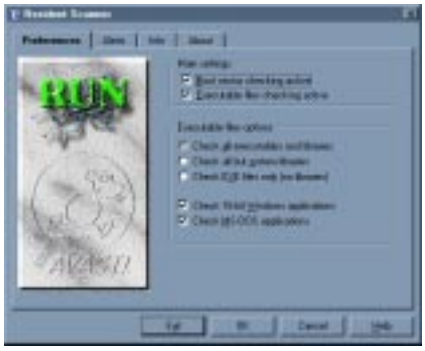
Alwil Avast 1.0

ItW Boot	97.1%	Standard	100.0%
ItW File	100.0%	Polymorphic	79.7%
ItW Overall	98.6%	Overall	94.2%

Avast, from Czechoslovakian *Alwil Software*, always does well in *VB* comparatives – its combination of a full features list and good detection makes it a winner. This version is no exception – the reviewer's first reaction to its interface was 'this is gorgeous!'. Friendly bitmaps abound, and the user is guided through setting up a scan via a wizard-based system.

The package includes the same components as the DOS version – behaviour blocker, resident scanner, on-demand scanner, and integrity checker. A combination of these should be enough to protect any machine.

The installation process is easy: once the user information and a serial number have been entered, installation is carried out. If the user follows the defaults and opts to use the behaviour blocker, this is activated immediately, and thus promptly picks up the installation program deleting temporary files as it completes. On-screen instructions warn about this fact, which it can be viewed either as annoying (one part of a product really should not trigger another in this way), or as reassuring (the user learns that the behaviour blocker is working, and what its alert looks like).



However, one serious problem was found: the boot sector of diskettes is not checked if the disk is blank. This put a crimp in testing against the Boot Sector set, as the sample diskettes contain

only the virus. To get figures, replicants were made, a file copied onto each disk, which was then scanned. *Alwil* informs us that the problem is fixed in the next version.

Avast is easy to use for the novice, but it is slightly annoying for the experienced user to have to work through all the dialogs to start a scan. This is offset by the ability to save and restore configurations – once a scan configuration is set up, it is saved, and can be requested again. Documentation is concise and accurate, but could be better organised.

Detection results, once the boot sector problem was dodged, were good – full marks on the ItW File and Standard sets raise the overall figure, and detection of polymorphics was above average. In the Boot Sector set, the product only missed Crazy_Boot and Stoned.Michelangelo.A.

Cheyenne InocuLAN 1.0

ItW Boot	97.1%	Standard	93.5%
ItW File	99.0%	Polymorphic	67.5%
ItW Overall	98.1%	Overall	89.3%

Another pretty product; a plethora of options are made manageable by a sensible screen layout and different types of input tools. Installation uses a wizard to guide the user through the available options, and a VxD (enchantingly called WImmune) is included for on-access scanning.

The scanner is easy to use, but includes fewer features than we have come to expect from *Cheyenne* (their NLM, for example, is one of the best-equipped on the market) – one definite lack is a scheduled scanning facility. The manual does describe how to configure *System Agent* to trigger the command-line scanner, but it would be nice to have a built-in scheduler for those people who do not wish to invest in *Microsoft Plus!*.

InocuLAN boasts one nice *Windows 95*-specific feature; a shell extension. This allows it to provide a menu option on the context-sensitive menu (displayed when the user clicks on the right-hand mouse button) which allows the user to scan the currently-selected object for viruses.

Other highlights are the ability to scan within certain archives, and the clear, well-illustrated documentation – though this latter appears incomplete on the subject of WImmune, and contains some oddities: the statement that a virus 'can reproduce by attaching itself to other files' ignores boot sector viruses, and elsewhere, the product is shown finding Michelangelo in a .BAT file, which seems highly unlikely.

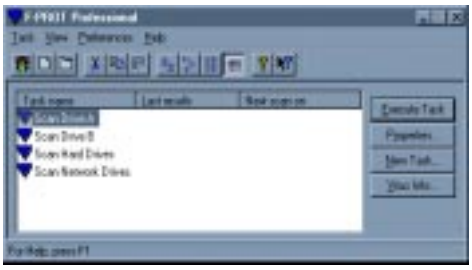
On the detection front, in the In the Wild File set the product only slipped up on Ph33r.1332, but missed both Chance.B and Stoned.Michelangelo.A from the Boot Sector set. Of the polymorphics, the product could not identify DSCE.Demo, PeaceKeeper.B, Russel.3072.A or SMEG_v0.3, and had only incomplete detection of Neuroquila, Nightfall.4559.B (both of which are in the wild) and Sepultura:MtE-Small. An overall score of 89.3% is, however, very creditable and shows considerable improvement over previous scores.

Command F-PROT 2.21a

ItW Boot	97.1%	Standard	96.6%
ItW File	100.0%	Polymorphic	53.9%
ItW Overall	98.6%	Overall	86.9%

This is a professional-looking piece of software – it hangs together nicely and functions well with *Windows 95*. It also features one of the more powerful scheduling systems seen in this comparative – a particularly clever feature is that once the user has set up scheduled jobs, they will fire even if the main program is not running – the resident on-access scanner is able to trigger them.

This package features a flexible scheduling system – any number of tasks may be scheduled to fire at any time on certain days in every week, every day, or on a certain day in every month. The schedule cannot be set to execute a scan more often than once a day, unless one opts to scan after a certain number of minutes of inactivity; a useful feature for machines with a high risk of infection.



As far as other features go, the user can control the type of scan, and which objects and areas to scan, in a fairly intuitive manner. The use of a context-sensitive (right-hand mouse button) menu in the main task display adds to the *Windows 95* feel, and this was one of the few products which it was possible to operate without referring to the manual.

In terms of detection, the high point was a note-perfect score on the In the Wild File set. The Overall In the Wild score is lower because the product missed two samples in the Boot Sector set; Chance.B and Stoned.Michelangelo.A.

The Polymorphic set yielded a depressingly low 53.9%: the product badly needs an overhaul of its polymorphic engine. *VB* understands that such an overhaul is underway, and expects Polymorphic, and hence Overall, detection rates to rise within the next few months. Meantime, the product still provides an admirable defence against the real world threat, signified by its good In the Wild results.

	Clean Floppy		Infected Floppy		Clean Hard Drive	
	Scan Time (min:sec)	Data Rate (KB/s)	Scan Time (min:sec)	Data Rate (KB/s)	Scan Time (min:sec)	Data Rate (KB/s)
Alwil Avast	0:30	32.5	1:00	19.5	2:21	1156.8
Cheyenne InocuLAN	0:30	32.5	0:36	32.6	3:05	881.7
Command F-PROT	0:25	38.9	0:38	30.9	2:15	1208.3
DataFellows F-PROT	0:39	25.O	0:44	26.7	2:06	1294.6
EliaShim ViruSafe	0:24	40.6	0:30	39.1	1:17	2118.4
ESaSS ThunderBYTE	0:25	38.9	0:33	35.5	1:04	2548.7
IBM AntiVirus	0:29	33.6	0:37	31.7	3:37	751.7
McAfee Scan	0:26	37.4	0:36	32.6	1:55	1418.4
Norman Virus Control	0:36	27.O	0:53	22.1	1:54	1430.8
Norton AntiVirus (NAV)	0:34	28.6	0:30	39.1	1:14	2204.2
On Technology Doctor	0:23	42.3	0:26	45.1	1:59	1370.7
RG Software Vi-Spy	0:24	40.6	0:29	40.4	1:29	1832.7
Sophos' SWEEP	0:35	27.8	0:32	36.7	2:37	1038.9
Stiller Integrity Master	0:27	36.1	0:38	30.9	2:32	1073.1
S&S Dr Solomon's AVTK	0:29	33.6	0:56	20.9	1:29	1832.7
Trend PC-cillin	0:33	29.5	0:55	21.3	2:37	1038.9

Table 2: The speed figures show the expected wide range of data rates.

Data Fellows F-PROT 2.21a

ItW Boot	97.1%	Standard	96.6%
ItW File	100.0%	Polymorphic	53.9%
ItW Overall	98.6%	Overall	86.9%

This product and the *Command* version of *F-PROT* achieve the same detection results, and have similar speeds: given that the engine for both is the same, this might be expected. However, the user interfaces of the two are totally different.

Installation, performed using *InstallShield*, offers the choice to install as a stand-alone workstation, or as network administrator; the idea being that the administrator installs a central copy, which he configures to suit his users, who then install from there. The administrator can thus prevent his users changing some aspects of configuration, which is very useful for site installations.

This product initially presents itself as a small window with a standard menu bar along the top. Several large buttons occupy the remainder of the space. One button folds a window out to reveal a task list allowing the creation and control of various jobs – these can be scheduled or simply available so that a user can scan different areas of his machine in different ways.

There is a fascinating section on the scan configuration dialog, where the user may select what to scan for; 'Viruses and Trojan Horses' and/or 'Document Macro Viruses'. It's curious that users are given the chance to turn everything off!

One elegant feature is the ability to add new buttons to the main button bar – this is a terrific way to make it easy for users to execute the most commonly-used tasks with the minimum of hassle. The only niggle is that this is done by holding down the CTRL key and double-left-clicking – using the right mouse button would fit the *Windows 95* ethos far more.

That niggle applies to the whole product: it is powerful and well thought-out, but feels more like a recompiled 16-bit version than a complete *Windows 95* product. As for detection, there is nothing to say above that said for the *Command Software* version of *F-PROT*.

EliaShim ViruSafe 1.1

ItW Boot	98.6%	Standard	88.7%
ItW File	94.1%	Polymorphic	31.6%
ItW Overall	96.3%	Overall	78.3%

Another nice install routine (it's not a wizard – in fact, it's very similar to the 16-bit *Windows* installation routine) puts *ViruSafe* on the hard disk, following the usual installation questions. Before installation is complete, the product places a minimised *Window* at the bottom of the screen – the install process says the product is 'marking' the computer's

executable files. A swift look at the manual revealed that it was not modifying files, but creating checksum files (VS.VS) across the disk – safer than any form of 'marking'.

The product, once installation is complete, is replete with *Borland*-style large friendly buttons, surrounding a list of available drives. The interface is reasonably intuitive, though it is almost impossible to guess what some of the buttons do without clicking on them. Another oddity is that the large button labelled 'Check' does not immediately perform a check on the selected area; rather, it pops up a menu – decidedly non-standard, and unintuitive.

There are no facilities to define tasks within the interface, although it is possible to use a command-line version of the program and set up short-cuts to start it with any desired options. This can be more powerful than built-in task lists, but is harder to set up. Scheduling is made available via an external program, TimeRun, allowing the user to set up any number of scheduled jobs (calls to the command-line scanner).

The scanner is one of the fastest of those tested, on clean and infected files, though detection is decidedly unexciting: the high point must be the single omission in the Boot Sector set (15_Years). However, missing 31 samples from the In the Wild File set is inexcusable. Finally, coming second from bottom on the polymorphics (completely detecting only Natas.4744 and SatanBug.5000.A, and not detecting any of seven of the sixteen) is, to say the least, disappointing.

ESaSS ThunderBYTE 7.00a

ItW Boot	100.0%	Standard	97.7%
ItW File	98.9%	Polymorphic	78.9%
ItW Overall	99.5%	Overall	93.9%

This arrived in standard *ESaSS* packaging – a CD-style case containing a single diskette (documentation is provided on the disk). The installation routine is simple and uncluttered.

Readers will recall that whenever this product comes around in DOS comparative reviews, its speed never escapes mention – *VB's* thesaurus has become extremely dog-eared over the years in the search for new adjectives to describe it.

However, whilst the *Windows 95* version is faster than its competitors (at least on the clean hard disk test), it is not nearly as far ahead as we might have expected. This is not simply because of the presence of *Windows*, but also, according to *ESaSS*, because they were not so concerned about speed for *ThunderBYTE's Windows* incarnations.

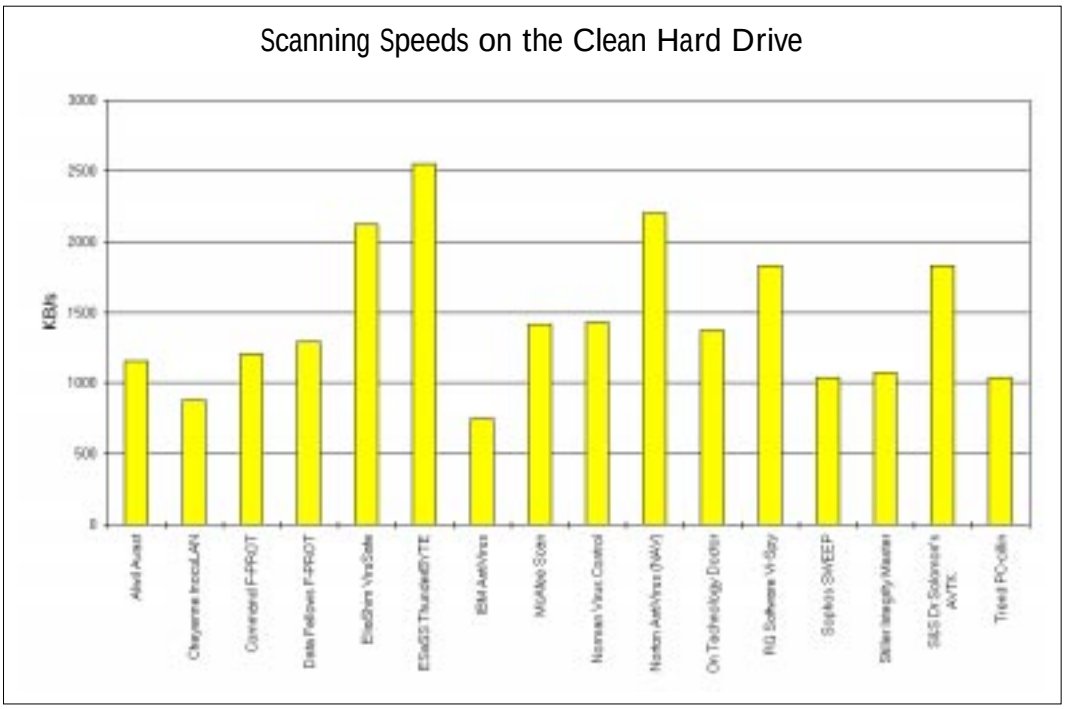
In terms of features, the only concession to the *Windows 95* environment is the ability (chosen at install time) to install a shell extension, which places a new option ('TbScan for viruses') on the context-sensitive menu. When selected, a new instance of the scanner is created, which immediately scans the object for viruses. The menu option only appears when the selected object is 'scannable'; e.g. if the file README.TXT is highlighted, the option will not be available, but it will be if INSTALL.EXE is highlighted.

Directories and drives are also scannable. The menu option appears for 'My Computer' and 'Network Neighbourhood', but when chosen, the scanner fails with the error 'Couldn't find files to scan'. Also, the shell extension's idea of 'scannable' is different from that of the scanner: for example, the menu option does not appear for DOC or DOT files. Despite the fact that the scanner includes these, the help file does not, at least in one location, list them as members of the default executable extensions list.

The product includes a resident on-access scanner, which is handled differently from the other products in the comparative. The others use a separate resident component which sits minimised, either on the toolbar or in the tool tray, but with *ThunderBYTE* the application itself remains loaded. It does not minimise properly, but becomes a tiny window just above the toolbar.

When it comes to detection, this product is one of only three to come out of the Boot Sector test unscathed. This was marred by dropping five samples on the In the Wild File set, including two of the four samples of Concepts Two of the remaining three missed samples were large (close to the 65,278 byte limit) COM files – it is possible, given *ThunderBYTE's* code emulation and heuristic techniques, that this is significant.

The result in the Polymorphic set is disappointing, but the poor percentage (78.9%) masks the fact that the scanner only missed 579 files. However, it only achieved complete detection of six of the sixteen viruses, so the percentage was dragged down. Finally, *ThunderBYTE* was one of the few products to suffer a false positive – one file in the clean set was incorrectly identified as 'infected with Golgi'.



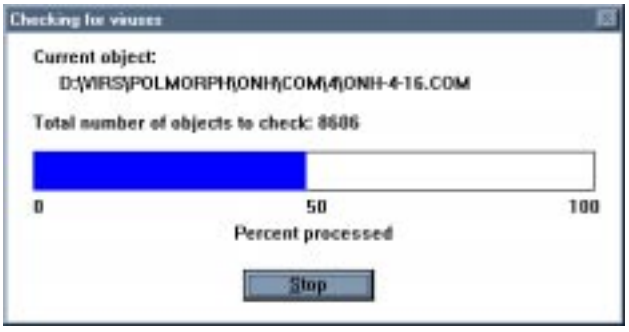
IBM AntiVirus 2.4.1

ItW Boot	97.1%	Standard	98.2%
ItW File	99.5%	Polymorphic	69.6%
ItW Overall	98.3%	Overall	91.1%

IBM AntiVirus for Windows 95 installs itself without difficulty – the user is offered plenty of options if they request a custom installation, but things proceed in silence if they opt for the default, express, method.

After installation, the system runs the scanner across the PC's fixed disks; to ensure that no viruses are present, and to build up its checksum list, which is used in subsequent scans, and speeds up scanning greatly: before it checks a file for viruses, *IBMAV* first checks whether the file has changed since it last looked at it. If not, it cannot be infected, so there is no need for a complete check. Clearly, this technique is only applied to fixed disks, but there it is very effective.

This creates a problem for speed testing, however – should the speed be quoted with or without the checksum file present? For this comparative, the speed given in the table and the graph is that without the checksum file: the speed only of the scanner, not that of the checksummer, has been tested. With the checksum file present, the clean test was completed in 49 seconds (a data rate of 3,328.9 KB/s), placing the product first in terms of clean hard disk scan speed. As can be seen from the table and graph, the product, without the checksum file, is the slowest in this category.



The product displays the *IBM* 'single big button' approach – the application's small main window is dominated by a large button bearing the legend 'To check your system for viruses now, push here'. If the user can resist such a tempting button [*I couldn't. Ed.*], the configuration options are available from the menu bar.

The scheduling facilities are limited: it is only possible to have one scan scheduled at once, and overall the product is not at all integrated into the *Windows 95* environment. It is a 32-bit application, but like some of the others in this test, that is not really the point – it would be nice to see features such as better scheduling, multi-threading and the usability enhancements which *Windows 95* has to offer.

In the detection tests, the product missed two samples (Feint and Stoned.Michelangelo.A) in the Boot Sector set, and only one (one of the two samples of No_Frills.Dudley) in the In the Wild File set.

Against the polymorphs, it suffered incomplete detection of groups of samples of one virus, just like *ThunderBYTE*. Full detection was only achieved in six of the sixteen sets, with detection in five of the others above 490 out of 500 samples: complete detection is a must here. As a result, the product drops to a disappointing sixth overall in detection: this will rise sharply when polymorphic detection improves slightly.

McAfee Scan 2.01.218

ItW Boot	98.6%	Standard	96.0%
ItW File	100.0%	Polymorphic	65.0%
ItW Overall	99.3%	Overall	89.9%

The installation for this product (which fits onto a 1.44MB diskette and still leaves over 250K free) is as straightforward as we have begun to expect from *Windows 95* products:

there is the usual choice between custom and express, and the default was express – this did not cause problems on the test machine.

After installation and a post-install reboot, *Scan* is ready to go. The default install sets up the on-access resident scanner, VShield – when it detects a virus in a program file in the boot sector of a diskette, it displays a warning screen in DOS text mode, as opposed to a window superimposed over the GUI which most of the other products choose. This aside, the resident scanner is nice to use: configuring it is simply a matter of double-clicking on the VShield icon, selecting 'Properties', and the manipulating options on a tabbed dialog.

The on-demand scanner (ViruScan) fits nicely into the *Windows 95* look and feel, and is a doddle to use – at the loss of some power and flexibility, however. Configuration options are few; indeed, it appears that VShield is more configurable than ViruScan. There is no facility to schedule scans, though it is possible to save and restore settings, giving the user the choice of multiple saved configurations.

A shell extension allows ViruScan to insert itself into the context-sensitive menu of objects which it can scan in the same way as *InocuLAN* and *ThunderBYTE*; however, this product only opens the scanner and sets the options. It does not actually start the scan: the user must click 'Scan Now' to trigger it, which seems an odd design decision.

When a previously-saved scan configuration file (.VSC) is highlighted in Explorer, choosing Properties from the context-sensitive menu shows that *McAfee* has done more with its shell extension than other companies – on the tab dialog are two custom sheets allowing the user to configure a particular saved scan without having to start ViruScan, stop the scan, change settings, and then save again.

An increasingly common solution to the problem of log file display (the terms 'activity log' and 'log file' are used by ViruScan seemingly interchangeably) is illustrated here: instead of building a complex text viewer into the scanner, the simple and sensible option is to use an external viewer.

Scan uses, predictably, Notepad, which is invoked whenever the user asks to view the activity log. The disadvantage is that there is no possibility for slick handling of the display: other products arrange it so that, from the internal viewer, double clicking on a line in the log will bring up information about the virus mentioned there, for example. This is not possible with an external viewer.

The on-line help deserves a mention: *Windows 95* applications often offer a 'What's this?' option on the context-sensitive menu, which brings up help on the object the pointer is over. *Scan* is no exception.

Speed tests show the product hovering around the middle of the field in all categories. For detection, *McAfee* has come a long way in the last year: the In the Wild File score is spot on, and it dropped just one (Stoned.Michelangelo.A) on the Boot Sectors. Polymorphic detection is the low point, but then it is the job of this set to test the products intensively: *McAfee* clocks in just above the halfway point in the rankings here, helped by very complete detection: 500 out of 500 in nine of sixteen groups, 497 in another (Coffee_Shop and Groove), 431 in SMEG_v0.3, and against Sepultura:MtE-Small, a lucky two. These results show that the technology is now very much there for *McAfee* to come up trumps. The company seems to have been concentrating on raising its In the Wild detection rates, with resounding success.

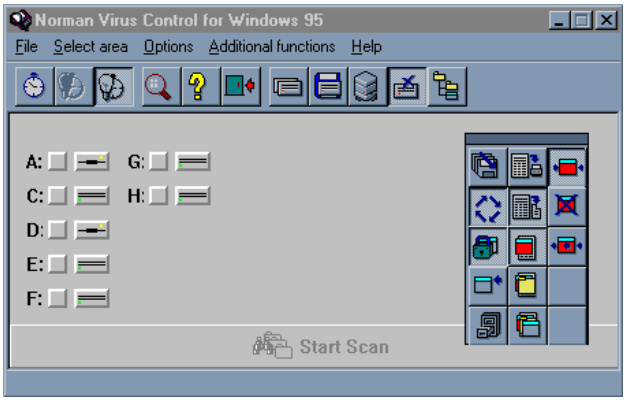
Norman Virus Control 3.50

ItW Boot	90.0%	Standard	98.1%
ItW File	98.5%	Polymorphic	98.4%
ItW Overall	94.2%	Overall	96.2%

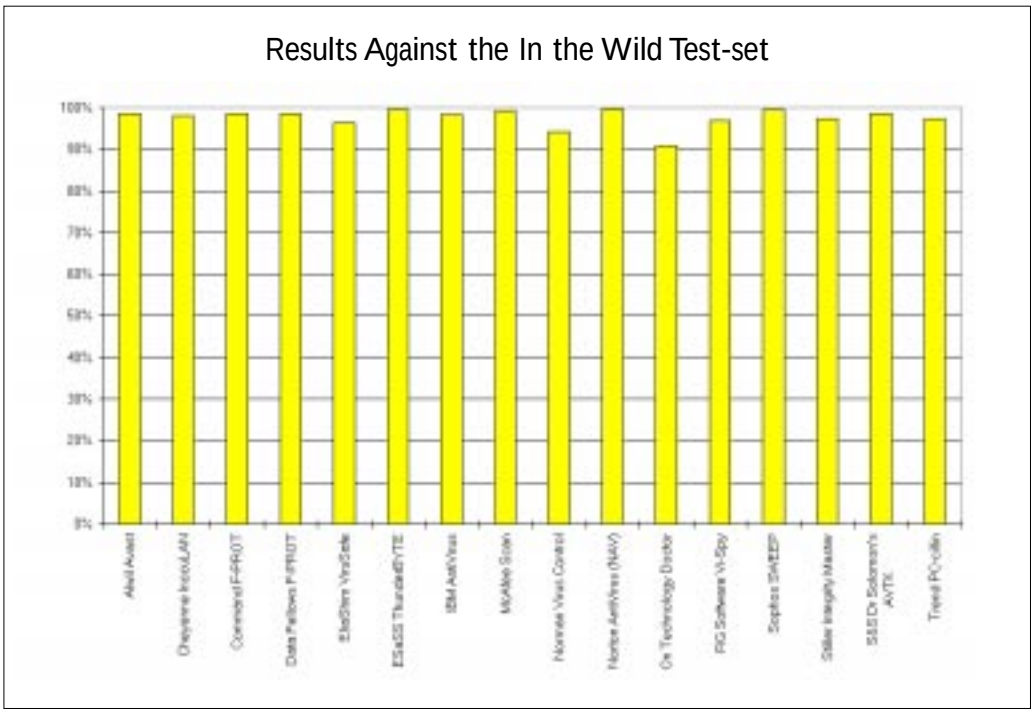
NVC for Windows 95 is one of the relatively few *Windows 95* products reviewed here which does not include a resident component – the manual states that a version of Norman's 'smart behaviour blocking device driver, NVC.SYS' is under development. For the moment, the install process deposits a GUI-driven scanner onto the hard disk, and creates the traditional program group and icon.

Executing the scanner presents an interface best described as different: the main area of the window carries a selectable list of drive letters, and floating above this is a toolbar which can be moved around independently of the main window. This toolbar carries push buttons to control the options of the scan that is to be performed – whether or not to attempt to disinfect infected files, write a verbose log file, etc.

Fortunately, tool tips have been included (these are messages that pop up when the pointer is left stationary over a button, explaining what the button does), because the icons



are decidedly cryptic [*It can't be easy designing an icon to convey the message 'Look for EXE Header'. Ed.*]. Scan settings can be altered from a property page accessed via the menu bar (indeed, some settings can only be controlled from here); however, once a user learns what the buttons on the toolbar do, it will be much quicker configuring the scanner that way.



At first, there appeared to be no way to save and restore scan settings; however, *NVC* deals with 'styles'. The user can create three new styles in addition to the default, and these can be loaded into the scanner and used as desired, used for scheduled scanning, and used to create icons on the desktop (the user creates a short-cut to *NVC* with a command line parameter and the name of the style, enabling a certain scan to be started immediately).

Scheduling is carried out using a component called NVS95, the Norman scheduler. This alone needs to be placed in the startup group, removing the need to keep the whole scanner running, minimised, to ensure that scheduled jobs fire. The scheduler is not, however, completely flexible: there are restrictions on how many scans (and what type – daily, weekly, or monthly) may be scheduled concurrently.

Again, this is a product which gets the job done and is perfectly usable, but it does have the appearance of a recompiled 16-bit version. There is little integration with the *Windows 95* environment, which is a shame, as all the building blocks are in place. Hopefully, *Norman* will gradually incorporate such features. Scanning inside archived files is an option, but requires the unarchiving program to be present and in the path: the scanner calls the archiver to decompress the files.

As to scanning, *NVC* is neither very fast nor very slow. In detection, the high point is without doubt the Polymorphic set: missing just two samples from 8000 places the product joint top with *Sophos' SWEEP* in this category – 98.4% is an exceptional score here.

Unfortunately, the scanner missed seven samples in the Boot Sector set, four more than the next worst, giving it a score here of 90.0%. The In the Wild File set was slightly better; here the product missed only six, which in this set is enough to place it fourth from bottom, and second from bottom in the Overall In the Wild ratings.

Norman undoubtedly has the technology in place to detect anything the current virus situation can throw at it. It is now only a question of adding information to their scanner about the few viruses they do not currently detect.

Norton AntiVirus 95.0.a

ItW Boot	100.0%	Standard	96.0%
ItW File	99.5%	Polymorphic	58.4%
ItW Overall	99.7%	Overall	88.5%

Fresh from the advertising battle currently underway in the US between *McAfee* and *Symantec*, *Norton AntiVirus* arrived in the *VB* office on three 1.44MB diskettes, plus the usual update diskette. On the update diskette, *Symantec* needs to provide a utility to install signature updates – an option on the scanner interface, perhaps, like some other products – as installing the updates is not currently as easy as it should be.

The rest of the install process is, however, well thought-out, and the user is guided through the options by the most wizardly of wizards. The install process encourages the user to make a rescue disk set (two diskettes), to be used in event of problems with the boot sectors of the bootable drive to boot from and attempt to fix the problem.

After the computer has rebooted following installation, *NAV* reveals its presence via two icons sitting in the tool tray: one is for the resident on-access component, AutoProtect; the other is for the scheduler (of which more later). AutoProtect is similar to *McAfee's* VShield in that when a virus is detected, a warning screen is displayed in DOS text mode. The user is offered options on virus detection controlled by settings from the Options section.

The main scanner window is nicely laid out, and has large friendly buttons along the top just beneath the menu bar, along with a drive letter display which occupies most of the window. Selecting 'Options' brings up a window which has eight property pages, allowing all parts of *Norton AntiVirus* to be configured in almost any way imaginable.

The Scheduler handles its task flexibly and comprehensively, and has finer resolution than most of the other products, which enables the user to schedule jobs to occur at time intervals right down to hourly, if he so wishes.

The product integrates better than some into the new environment; the right hand button produces a context-sensitive menu offering help on the current object. However, there are neither shell-extensions nor other clever things *Windows 95* makes possible; here *McAfee* has the edge.

Also, whilst it is possible to start a scan from a certain directory, it is annoying to have to navigate a *Windows 95*-style directory browser window to do so. This is fine for novice users, but more experienced ones may wish simply to type in the name: it is much quicker that way.

Interestingly, this is one of the few products tested which comes out as faster over the infected diskette than over the clean one; unusual, but certainly not a problem. On a clean hard drive, the product is second only to *ThunderBYTE*.

In the Boot Sector set, there were no problems: complete detection here is nice to see. The score against the In the Wild File viruses was only marred by missing two samples of Virogen.Pinworm. The Polymorphic set presented the only real problem, and even here there has been marked improvement from a year ago: this version achieved complete detection in nine of the sixteen groups, detected approximately half of the samples of SMEG_v03, and detected nothing in the other groups; which gave a combined score of 58.4%.

The detection engine has come a long way, and has made the leap quickly, at least from a user's point of view. Congratulations to *Symantec – VB* looks forward to more of the same.

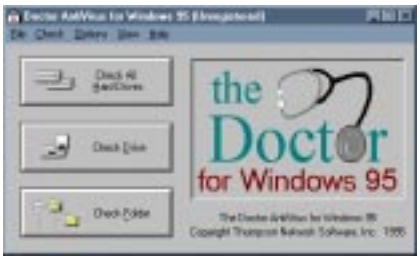
On Technology Doctor 96.05

ItW Boot	95.7%	Standard	90.6%
ItW File	86.0%	Polymorphic	25.2%
ItW Overall	90.9%	Overall	74.4%

Perhaps better known as a product developed by *Thompson Network Software*, this package has been renamed following the company's recent acquisition by *On Technology*. *The Doctor* arrived on a single high-density floppy disk, and the install process proceeded without incident.

Once the install is complete, a reboot results in the resident on-access scanner, The Doctor Anti-Virus Monitor, becoming active. Configuration options (fairly limited in scope) are available by double-clicking the icon in the tool tray.

Several problems were encountered when it came to using the on-demand scanner. Whilst running across the test-sets, it generated an address exception. In fact, the original version of the product received failed to detect the In the Wild File samples, and several sample groups within the polymorphic set. A swift call to the developers revealed that they had had some problems with that version of the product – a new one was retrieved electronically [*Hence the fact that this product has a later date than the rest of those reviewed. Ed.*], and testing proceeded.



Product features are basic – there is no way to save multiple sets of scan settings, and no scheduling capability. On-line help is sufficient to guide the user around the product without problems. Integration into the *Windows 95* environment seems non-existent, which is a shame – the product appears to be a recompiled 16-bit *Windows* version of the Doctor. One notable peculiarity is the fact that selecting the 'Options' item from the menu bar pops up a dialog box – this is very non-standard behaviour.

The speed tests place the product fastest in terms of scan time on both clean and infected floppy disks, although it clocked in around the middle of the field for the clean hard disk scan. Unfortunately, even with the fixed version, the product was unable to complete scanning the samples of One_Half.3544 without generating an address exception. The product ranked bottom in all sets apart from Boot Sectors, and requires considerably more work before it places reasonably. Further, the product encountered four false positives whilst scanning clean files.

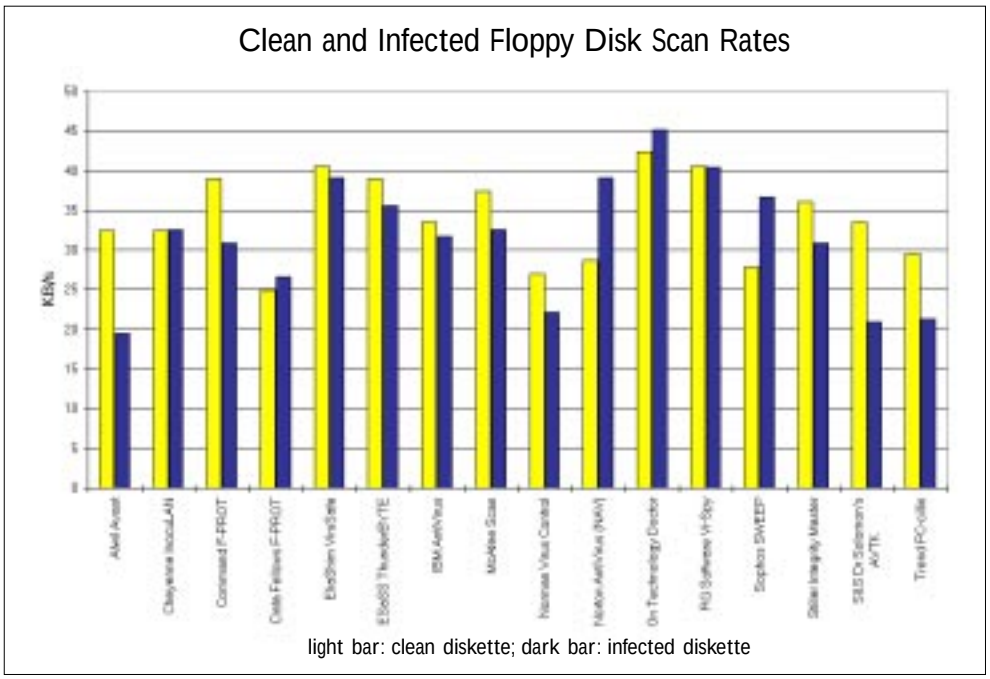
RG Software Vi-Spy 14.02.96

ItW Boot	97.1%	Standard	97.7%
ItW File	96.5%	Polymorphic	52.4%
ItW Overall	96.8%	Overall	85.9%

This version of *Vi-Spy* arrived on one high-density diskette, labelled 'Win95 Release Candidate'. The software was accompanied by a comprehensive manual. Installation is via a DOS program, which ran without difficulty in *Windows 95*:

it even ducks back to *Windows* at the end to create a program group and install icons.

The resident software provided with the product is set apart from all the others in this test: it is not a VxD. How it works is not relevant to the user; the important thing is that it does work. Message boxes are displayed when an infected file or disk is accessed, and a program called RVS can be used to display information about the last alert.



The *Windows* parts of the scanner are really just programs which assemble the command-line to call the engine: this appears messy after watching all these products which have pure graphic user interfaces do their stuff, but again, it gets the job done, which is all that is important. However, the fact that the product works this way does preclude many *Windows 95*-specific features.

The product comes out towards the top of the heap on speed tests (due in part to not having to drive a GUI), and scan speeds on diskettes were particularly high. The detection rates hit a high in the Boot Sector test-set, where missing two samples (Chance.B and Crazy_Boot) gave it a score of 97.1%. However, missing twelve samples from the In the Wild File set was very disappointing. In the Polymorphic test-set, full detection was achieved in five of the sixteen groups – all of the other groups were partially detected.

Sophos' SWEEP 2.83

ItW Boot	100.0%	Standard	100%
ItW File	99.2%	Polymorphic	98.4%
ItW Overall	99.6%	Overall	99.4%

Sophos is well known for the high detection rates of its virus scanner S*WEEP*, which is now available in a *Windows 95* incarnation. The installation process is just as straightforward as all the others, and no problems were encountered using either the defaults or custom settings. Following installation, an icon for S*WEEP* has been created within the expected new group under the Start button.

In the past, *Sophos* has not produced a version of S*WEEP* with a graphic user interface for any operating system; consequently, users might well be expecting a command-line scanner. Those who do will, one hopes, be pleasantly surprised, because *SWEEP for Windows 95* boasts a nice graphical user interface.

This interface is laid out in a fairly standard manner: a selectable list of areas to scan sits beneath a menu and button bar. Drives may be added to and removed from the list using buttons to its right. On the right of each drive is a light: when lit green, this drive will be included in the next immediate scan. Clicking on the light changes the status.

Scanning options are changed from a standard tabbed dialog box – some of the pages are rather peculiarly laid out, but there is no loss of usability.

Sweep's scheduling capabilities are more flexible than most: any scan or combination of scans may be set to trigger at any number of times on any combination of days of the week. It is not easy, however, to repeat the scan every hour (the user would need to add all the times explicitly).

The on-line help is readable and contains all the information required to drive the more complex parts of the software, although it is not context-sensitive, and the *de facto Windows* standard of pressing F1 for help does not work (on most of the other products it does). Finally, there is little integration into the *Windows 95* environment: shell extensions and the use of the right-hand button are absent. No resident software is provided, although a *Windows 95* client for *Sophos'* InterCheck client-server virus detection system is available.

Sweep is another product which scans faster on infected files: infected disk scan time is three seconds lower than the clean floppy scan time. Overall, it is one of the slowest tested: clean hard disk scan rate of just over 1MB/second ranks it joint third from bottom here.

Detection, however, is becoming hard to fault: when a product only misses five samples across all the test-sets, the tester begins to wonder what he's done wrong. The five samples were two of Code.3952:VICE.05 and three of Concept (missed only because at the time of the test .DOC was not in the default list of scan extensions – an omission which needs to be fixed). Other than that, it got everything. It is hard to find problems with a performance like that.

Stiller Integrity Master 2.61b

ItW Boot	95.7%	Standard	96.3%
ItW File	98.9%	Polymorphic	36.9%
ItW Overall	97.3%	Overall	81.9%

This product is not specifically designed for *Windows 95*, but *Stiller Research* states that it works equally well under *Windows 95*, *Windows 3.1*, and DOS; consequently, it was included in this test. Installation is, as expected, via a DOS character-mode menuing system, and proceeded without incident. The only nod in the direction of *Windows 95* was a mention at the end of the installation process.

Integrity Master presents itself as a DOS menu-based system, although it is possible to drive it from the command line. The menus are explained in full by pop-up boxes which appear by each item as you move up and down the menus. The product offers, via these menus, a vast range of options, allowing the user to specify the type of checking – *Integrity Master* has a major integrity checking component which can also be controlled here.

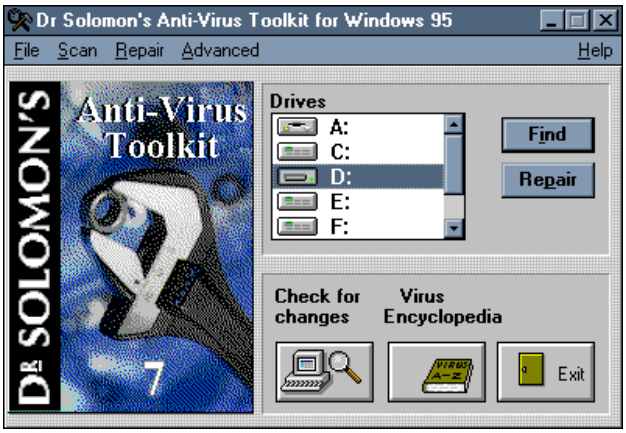
Integration into the *Windows 95* environment is low: after all, this is really a DOS product. The help is thorough; however, one thing which stands out is the decidedly unintuitive way in which the user is able to change the drive which is about to be scanned. Such a seemingly trivial task really should be easier to perform.

The speed tests place the product towards the bottom of the field on the clean hard disk scan and around the middle for the floppy tests. Detection rates are at their best in the two In the Wild sets, although missing three (Feint, Chance.B, and Crazy_Boot) in the Boot Sector set was disappointing. Polymorphic detection is the low point: this should be improved to raise the overall score.

S&S Dr Solomon's AVTK 7.57

ItW Boot	97.1%	Standard	100.0%
ItW File	100.0%	Polymorphic	90.5%
ItW Overall	98.6%	Overall	96.9%

Dr Solomon's AVTK is one of those products which is always expected to do well in comparatives such as this: it always comes close to the top. Backed up by a large team of experts (*S&S* has the honour of being one of the anti-virus companies with the most CARO members), its pedigree is high.



The *Windows 95* version comes on three diskettes, and installs via the usual wizard without any problems. After installation and a reboot, WinGuard (the resident scanner) is active and the *Toolkit* is ready for use.

The product's main window is identical to that of the *Windows 3.1* version of the *Toolkit* – a large logo, and a drive selection area with associated buttons. The user makes a selection of drives to scan, and can scan them immediately. Scanning a specific subdirectory is slightly more tricky: for some reason, this is classified as an advanced option. The configuration options are plentiful, and easy to change, and all setups may be saved and restored as desired.

Both the documentation and the on-line help are complete and informative. The on-line help even offers the VIRUS-L FAQ for browsing, although it is a 1992 version. [*This FAQ has recently been updated after a long period of stagnation;*

however, it appears that this change has yet to make it into the Toolkit. Ed.] Alas, once again there is no attempt to blend the product into the *Windows 95* way of doing things. There is, as has been seen so often in this review, no use of context-sensitivity or suchlike; this too is a recompilation of the 16-bit version of the product.

WinGuard is a nicely-done piece of software: when it finds a virus, it displays a proper *Windows 95* dialog box informing the user of the fact. It is also easy to configure and manage.

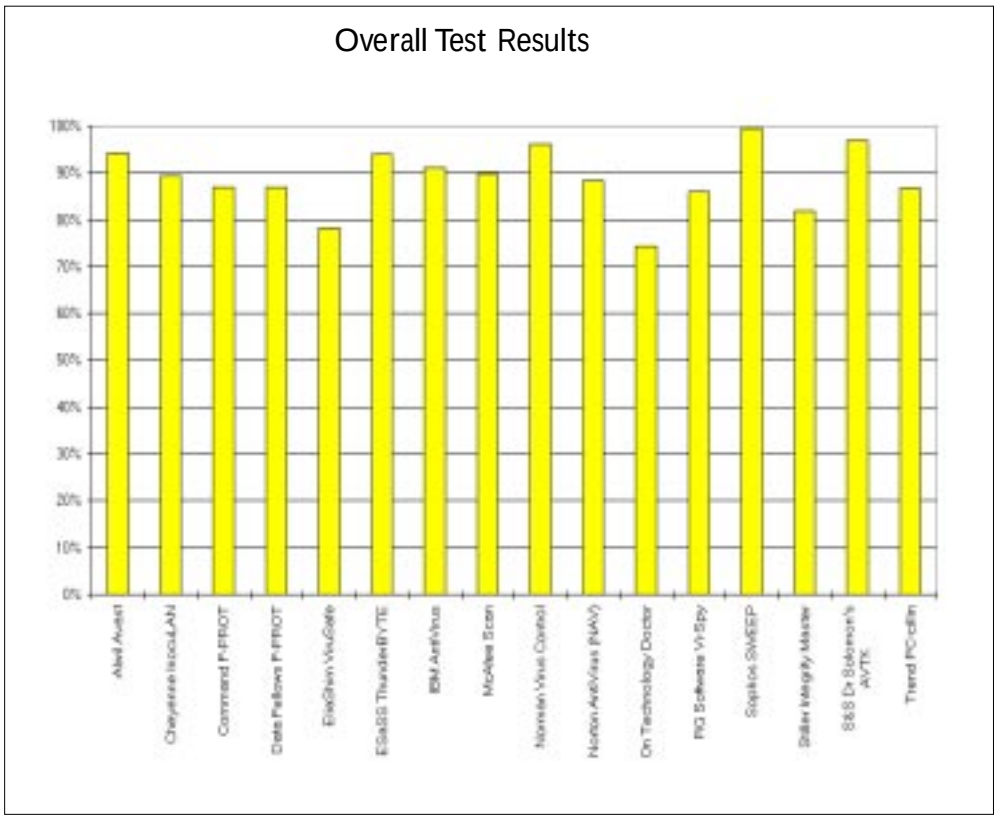
Scheduling functionality is provided via two external applications: the Schedule Editor, which creates, edits, and manages scheduled jobs, and the other, which sits in the background waiting to trigger jobs when the time comes. For some reason, this latter does not dock with the tool tray, which would be nicer than having an icon in the main section of the tool bar. Both do their job without apparent difficulty.

One oddity, however, is that the main *Toolkit* application is written using the *Borland* C compiler, and the Schedule Editor uses *Microsoft* C. This may not seem important, but both applications have a scan options dialog, each of which contains the same options and works in the same way. However, they look completely different, due to the different layouts chosen by the designers, and the different widgets used by the compilers.

As to the test results; in terms of speed, the product clocks in at joint fourth fastest on the clean hard drive test, and slightly further down the field on the clean floppy test: there are no problems here. Detection is excellent, with the product not missing a trick (or a sample) on the In the Wild File or Standard sets. Two missed boot sectors (Chance.B and QRry) were all that stood between the *Toolkit* and 100% for overall In the Wild detection.

With the Polymorphics, it seems that the problem the engine developers have been experiencing for the last few comparatives is very nearly behind them: complete detection of thirteen of the sixteen groups is a great improvement. One sample of Sepultura:MtE-Small was missed, as were 35 of Code.3952:VICE.05. All remaining misses were samples of PeaceKeeper.B.

All round, an excellent performance from the *Toolkit*.



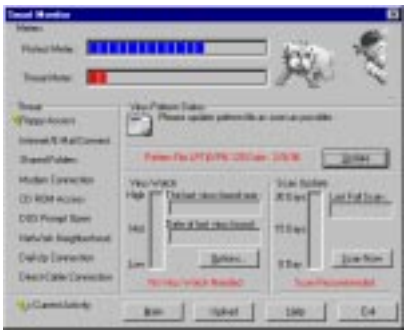
Trend PC-cillin 1.0

ItW Boot	95.7%	Standard	91.1%
ItW File	98.8%	Polymorphic	61.0%
ItW Overall	97.2%	Overall	86.6%

This is one of the few times that *PC-cillin* has been submitted for a *VB* comparative review, and the reviewer was understandably keen to test it. The product is widely used in the Far East, and *Trend* has been moving in on the US and UK market with a vengeance over the last few months.

The product comes on two diskettes, along with a nicely-written manual and a registration card. Installation requires a serial number from this card before it will proceed. Following the install process (which takes the standard form), a reboot brings the resident scanner (called PC-cillin Monitor) to life, and the system is ready to use.

The scanner component offers the usual configuration options via a large main window upon which the user may click down a *Windows 95*-style drive tree to choose areas to scan or clean. A notable options feature is the configurations controlling installation of virus pattern updates: besides installing the updates from diskette, *PC-cillin* allows the user to retrieve updates from the *Trend* FTP site or BBS. The download is not performed automatically: the user is able to press a button to retrieve the latest set. From here it is merely a step to making the process entirely automatic. There are security risks here, but that is too specific an issue for this review.



One notable feature of the Monitor part of the software is the fascinatingly complex Smart Monitor screen, which features the 'Protect Meter' and 'Threat Monitor'. One of the ways in which the product says that it works is by adjusting the protection level in accordance with what the user is currently doing and the history of virus infection (if you have previously encountered a virus, the resident software works harder; the theory being that you are at more risk).

It is not entirely clear from the documentation exactly what form this adjusting takes, or why the system does not simply not bother with all this and leave itself on permanent full alert (although educated guesses can be made at both of these). It proved impossible to tell under review conditions whether or not any actual protection changes were made as the threat meter rose or after a previous infection.

PC-cillin offers only fairly basic scheduling facilities, and only one job can be scheduled at a time. The quality of the on-line help varies quite considerably: most of it is perfectly adequate, but, for example, the help for the automated update configuration screen is almost useless.

The product has several useful features, including the ability to scan inside many different types of archive files, and the use of a shell extension to provide an additional item on the context-sensitive menu (which only appears when the selected item is a drive or a directory, not a single file).

In the speed tests, *PC-cillin* was joint third slowest (with *Sophos' SWEEP*) on the clean hard disk test, and the floppy scan times were also on the slow side of average.

Detection rates on the In the Wild sets were fairly good. Missing three (Chance.B, Feint, and Satria.A) on the Boot sector set places it below average, though missing only two (one sample each of Ph33r.1332 and Trakia.653) on the In the Wild File samples is a good result. The product's place at third from bottom in the Standard test-set reflects its relative youth. In the Polymorphic set, a score of 61% is far from disgraceful, but leaves considerable room for improvement.

After Testing

As we emerge, rather breathless, into the light after testing sixteen fairly new anti-virus products in some depth, what conclusions can be drawn?

Overall, the look and feel of the products was fairly unimpressive – the hoped-for '*Windows 95*-ness' was not present. If anti-virus products must be given flashy user interfaces, they should make full use of the flexibility offered by the OS.

On a more positive note, it is extremely gratifying to see so many good scores on the In the Wild sets: all sixteen products scored over 90% on the combined In the Wild score, with a massive ten getting over 98%.

However, had this been an *NCSA* certification, none of the products would have qualified: no-one got 100% on the combined In the Wild score. Even if the effect of Michelangelo had been removed [*see Editorial, p.2, for more information*], only *McAfee Scan* would come out with 100%.

The Polymorphic set, as usual, caused the most problems: the current set contains much more recent viruses than most. It is this that puts the cat amongst the pigeons and causes the low scores.

This test-set is a useful indication of the level of scanning technology that the products have, but should not be seen as more important than the In the Wild set. *Norman Virus Control*, for example, missed only two polymorphic samples (of a total of 8000); unfortunately, across both In the Wild sets it missed thirteen (from 370) – a fascinating dichotomy. It is difficult to recommend this product with such an In the Wild result; however, once that is improved…

As far as the In the Wild set is concerned, *Norton AntiVirus* triumphs here, closely followed by *Sophos' SWEEP*, which is just as closely followed by *ESaSS' ThunderBYTE*. Then comes *McAfee Scan* in fourth place, and behind that, in joint fifth, are *Alwil Avast*, the two *F-PROTs*, and *S&S Dr Solomon's AVTK*.

Looking at the Overall rankings, *Sophos' SWEEP* comes in first with a 2.5% lead over *S&S Dr Solomon's AVTK*. *Norman Virus Control* is third, *Alwil Avast* fourth, and *ESaSS' ThunderBYTE* fifth – but recall that *ThunderBYTE* suffered a false positive.

In terms of speed, *ESaSS* has the edge by a quite considerable margin in the figures used for the graph (although once *IBM AntiVirus* has done its initial scan, it comes out in the lead). *Norton AntiVirus* is not that far behind.

Leave detection and speed for a moment, and look at the overall usability of the scanner in *Microsoft's* new environment. In the reviewer's opinion (and this is subjective: what is a usable and elegant interface to one man could be the antithesis of the same to another), *McAfee Scan* walks away with this one, head and shoulders above the rest. It is the only product which gives the impression that people sat down and designed a *Windows 95* anti-virus product, rather than one that simply runs under *Windows 95*.

In Conclusion

Once again, we are left with different products coming out ahead in each category: for In the Wild detection, *Norton* has it by a nose; for Overall detection, it's *Sophos* and *S&S*; for speed, *IBM* and *ESaSS*; for usability, *McAfee*.

Nothing is ever easy, is it?

TECHNICAL DETAILS

Hardware used: *Compaq ProLinea 590*, 16MB RAM, 2.1GB disk and a 270MB *SyQuest* removable drive.

Software: *MS Windows 95* with Service Pack One installed. The only unusual software present was *Windows 95* native *SyQuest* drivers.

Other technical information: After reviewing each product, a complete disk image of the operating system and associated applications was restored to the test machine from a sector-level backup on a *SyQuest* cartridge. The next product was then installed. Samples of file infecting viruses were held, and scanned, on a write-protected *SyQuest* cartridge. Boot sector viruses are all genuine infections, and are held (one each, of course) on write-protected 3.5-inch floppy diskettes. The February 1996 WildList was used as the source for the In the Wild test-sets.

TEST-SETS

In the Wild Boot Sector viruses: This test-set contains one sample each of the following 70 viruses:

15_Years, AntiCMOS.A, AntiCMOS.B, AntiEXE, Boot.437, BootEXE.451, Brasil, Bye, Chance.B, Crazy_Boot, Da_Boys, Diablo_Boot, Disk_Killer, DiskWasher.A, Empire.Int_10.B, Empire.Monkey.A, Empire.Monkey.B, EXEBug.A, EXEBug.C, EXEBug.Hooker, Feint, Finnish_Sprayer, Flame, Form.A, Form.C, Form.D, Frankenstein, J&M, Joshi.A, Jumper.A, Jumper.B, Junkie, Kampana.A, Leandro, Music_Bug, Natas.4744, NYB, Parity_Boot.B, Peter, QRry, Quox.A, Ripper, Russian_Flag, Sampo, Satria.A, She_Has, Stealth_Boot.B, Stealth_Boot.C, Stoned.16.A, Stoned.Angelina, Stoned.Azusa.A, Stoned.Bravo, Stoned.Bunny.A, Stoned.Dinamo, Stoned.June_4th.A, Stoned.Kiev, Stoned.LZR, Stoned.Manitoba, Stoned.Michelangelo.A, Stoned.No_Int.A, Stoned.NOP, Stoned.Standard, Stoned.Swedish_Disaster, Stoned.W-Boot.A, Swiss_Boot, Unashamed, Urkel, V-Sign, WelcomB, Wxyc.A.

In the Wild File viruses: There are 300 samples in this test-set, made up of the following viruses (number of samples of each in parentheses after the virus name):

_814 (3), Accept.3773 (5), Anticad.4096.A (4), Anticad.4096.Mozart (4), Arianna.3375 (4), Avispa.D (2), Backformat.A (1), Bad_Sectors.3428 (5), Barrotes.1310.A (2), BootEXE.451 (1), Bosnia:TPE.1_4 (5), Byway.A (1), Byway.B (1), Cascade.1701.A (3), Cascade.1704.A (3), Cascade.1704.D (3), Cawber (3), Changsa.A (5), Chaos.1241 (6), Chill (1), Concept (4), CPW.1527 (4), Dark_Avenger.1800.A (3), Datalock.920.A (3), DelWin.1759 (3), Die_Hard (2), Dir-II.A (1), DR&ET.1710 (3), Fairz (6), Fichv.2_1 (3), Finnish.357 (2), Flip.2153 (2), Flip.2343 (6), Freddy_Krueger (3), Frodo.Frodo.A (4), Ginger.2774 (2), Green_Caterpillar.1575.A (3), Helloween.1376.A (6), Hi.460 (3), Hidenowt (1), Jerusalem.1244 (6), Jerusalem.1808.Standard (2), Jerusalem.Sunday.A (2), Jerusalem.Zero_Time.Australian.A (3), Jos.1000 (3), Junkie (1), Kaos4 (6), Keypress.1232.A (2), Lemming.2160 (5), Liberty.2857.A (2), Little_Brother.307 (1), Little_Red.1465 (2), Macgyver.2803 (3), Maltese_Amoeba (3), Manzon (2), Markt.1533 (3), Mirea.1788 (2), Natas.4744 (5), Necros (2), Neuroquila (1), No_Frills.Dudley (2), No_Frills.No_Frills.843 (2), Nomenklatura (6), November_17th.800.A (2), November_17th.855.A (2), Npox.963.A (2), One_Half.3544 (5), Ontario.1024 (3), Pathogen:SMEG.0_1 (5), Ph33r.1332 (5), Phx.965 (3), Predator.2448 (2), Quicksilver.1376 (1), Sarampo (6), SatanBug.5000.A (2), Sayha (5), Screaming_Fist.II.696 (6), Sibylle (3), Sleep_Walker (3), SVC.3103.A (2), Tai-Pan.438 (3), Tai-Pan.666 (2), Tequila.A (1), Three_Tunes.1784 (6), Trakia.653 (1), Tremor.A (6), Trojector.1463 (6), Vacsina.TP-05.A (2), Vacsina.TP-16.A (1), Vampiro (2), Vienna.648.Reboot.A (1), Vinchuca (3), Virogen.Pinworm (4), VLamix (1), Xeram.1664 (4), Yankee Doodle.TP-39 (5), Yankee_Doodle.TP-44.A (1), Yankee_Doodle.XPEH.4928 (2).

Standard File viruses: This test-set comprises 305 samples, from one to eleven examples of each of the following:

405 (1), 417 (1), 492 (1), 516 (1), 600 (1), 696 (1), 707 (1), 777 (1), 800 (1), 905 (1), 948 (1), 1049 (1), 1260 (1), 1600 (1), 2100 (2), 2144 (2), 5120 (1), 8888 (1), 8_Tunes (1), AIDS (1), AIDS-II (1), Alabama (1), Ambulance (1), Amoeba (2), Amstrad (2), Anthrax (1), Anti-Pascal (5), Argyle (1), Armagedon (1), Attention (1), Bebe (1), Big_Bang (1), Black_Monday (2), Blood (1), Burger (3), Butterfly.Butterfly (1), Captain_Trips (4), Cantando.857 (1), Casper (1), CeCe.1998 (6), Coffeeshop (2), Crazy_Lord (2), Cruncher (2), Dark_Avenger.2100.DI.A (2), Dark_Avenger.Father (2), Darth_Vader (3), Datacrime (2), Datacrime_II (2), December_24th (1), Destructor (1), Diamond.1024.B (1), Dir (1), DiskJeb (1), DOS_Hunter (1), Dot_Killer (1), Durban (1), EarJob.405 (3), Eddie (1), Eddie-2.A (3), Fax_Free.Topo (1), Fellowship (1), Fish_1100 (1), Fish_6 (2), Flash (1), Fu_Manchu (2), Genesis.226 (1), Greets.3000 (3), Halley (1), Hallochen.A (3), HLLC.Even_Beeper.A (1), Hymn (2), Icelandic (3), Internal (1), Invisible_Man (2), Itavir (1), Jerusalem.PcVrsDs (4), Jo-Jo (1), Jocker (1), July_13th (1), Kamikaze (1), Kemerovo (1), Kennedy (1), Lamer's_Surprise (1), Lehigh (1), Liberty (5), Liberty.2857.D (2), Loren (2), LoveChild (1), Lozinsky (1), Macho (2), Maresme.1062 (3), MIX1 (2), MLTI (1), Monxla (1), Murphy (2), Necropolis (1), Nina (1), Nothing (1), November_17th.768.A (2), NukeHard (1), Number_of_the_Beast (5), Old_Yankee (2), Oropax (1), Oxana.710 (3), Parity (1), Peanut (1), Perfume (1), Phantom1 (2), Pitch (1), Piter (2), Plague.2647 (2), Poison (1), Polish-217 (1), Power_Pump.1 (1), Pretoria (1), Prudents (1), Rat (1), Revenge (1), Riihi (1), SBC (1), Screaming_Fist.927 (4), Semtex.1000 (1), Senorita.885 (3), Shake (1), ShineAway.620 (3), Sibel_Sheep (2), Sofia.432 (3), Spanz (2), Stardot.789.A (6), Stardot.789.D (2), Starship (2), Subliminal (1), Sunday (2), Suomi (1), Suriv_1.01 (1), Suriv_2.01 (1), SVC.1689.A (2), Sverdlov (2), Svir (1), Sylvia (1), Syslock (1), Syslock.Macho (2), Syslock.Syslock.A (1), Taiwan (2), Telecom (4), Terror (1), Tiny (12), Todor (2), Traceback (2), TUQ (1), Turbo_488 (1), Typo (1), V-1 (1), V2P6 (1), Vacsina.634 (1), Vacsina.Penza.700 (2), Vacsina.TP.? (6), Vcomm (2), VFS (1), Victor (1), Vienna.Bua (3), Vienna.? (11), Virdem (1), Virdem.1336.English (1), Virus-101 (2), Virus-90 (1), Voronezh.1600.A (2), VP (1), Warrior (1), Whale (1), Willow (1), WinVir_14 (1), Yankee_Doodle.TP.? (5), Zero_Bug (1).

Polymorphic viruses: There are 8000 samples in this set; 500 each of:

Code.3952:VICE.05, DSCE.Demo, Girafe:TPE, Groove and Coffee_Shop, MTZ.4510, Natas.4744, Neuroquila.A, Nightfall.4559.B, One_Half.3544, Pathogen:SMEG, PeaceKeeper.B, Russel.3072.A, SatanBug.5000.A, Sepultura:MtE-Small, SMEG_v0.3, Uruguay.4.

VIRUS BULLETIN **©1996 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel +44 1235 555139. /96/$0.00+2.50**

Developer contact information:

Alwil AVAST! Alwil Software, Prubezna 76, CS-100 00 Prague 10, Czech Republic.
Tel +42 278 22050, fax +42 278 22553.
WWW: http://www.anet.cz/alwil/

Cheyenne InocuLAN Cheyenne Software Inc, 3 Expressway Plaza, Roslyn Heights NY 11577, USA.
Tel +1 516 465 5700, fax +1 516 484 1853.
WWW: http://www.cheyenne.com/

Command F-Prot Command Software Systems, 1061 E. Indiantown Road, Suite 500, Jupiter FL 33477, USA.
Tel +1 407 575 3200, fax +1 407 575 3026.
WWW: http://www.commandcom.com/

Data Fellows F-Prot Data Fellows Ltd, Paivantaite 8, 02210 Espoo, Finland.
Tel +358 0478 444, fax +358 0478 44599.
WWW: http://www.datafellows.com/

EliaShim ViruSafe EliaShim Software, PO Box 25333, Mifrats Haifa (Haifa Bay) 31250, Israel.
Tel +972 4 516 111, fax +972 4 852 8613.
WWW: http://www.eliashim.com/

ESaSS ThunderBYTE Management ESaSS BV, Saltshof 10-18, NL 6604 EA Wijchen, Netherlands.
Tel +31 24 642 2282, fax +31 24 645 0899.
WWW: http://www.thunderbyte.com/

IBM AntiVirus IBM AntiVirus Products, P.O. Box 201960, Austin, TX 78720-1960
Phone 1-800-742-2493 or (512) 434-1554FAX (512) 434-1536
Email IBM_ANTIVIRUS@HARTE-HANKS.COM, WWW: http://www.brs.ibm.com/ibmav.html/

McAfee Scan McAfee Associates, 2710 Walsh Avenue, Santa Clara CA 95051-0963, USA.
Tel +1 408 988 3832, fax +1 408 970 9727.
WWW: http://www.mcafee.com/

Norman Virus Control Norman Data Defense Systems, 3028 Javier Road, Suite 201, Fairfax VA 2203, USA.
Tel +1 703 573 8890, fax +1 703 573 3919.
WWW: http://www.norman.com/

Norton AntiVirus Symantec Corporation, 2500 Broadway, Suite 200, Santa Monica CA 90404-3063, USA.
Tel +1 310 449 4257, fax +1 310 453 0636.
WWW: http://www.symantec.com/

On Technology Doctor On Technology, 15 Hamby Road, Marietta GA 30067, USA.
Tel +1 770 971 8900, fax +1 770 971 8828.
WWW: http://www.on.com/

RG Software Vi-Spy RG Software Systems Inc, 6900 East Camelback Road, Suite 630, Scottsdale AZ 85251, USA.
Tel +1 602 423 8000, fax +1 602 423 8389, email rayglath@aztec.inre.asu.edu.

Sophos Sweep Sophos Plc, 21 The Quadrant, Abingdon Science Park, Abingdon, Oxon OX14 3YS, England.
Tel +44 1235 559933, fax +44 1235 559935.
WWW: http://www.sophos.com/

Stiller Research Integrity Master Stiller Research 1265 Big Valley Drive, Colorado Springs CO 80919-1014, USA.
Tel +1 719 533 1879, fax +1 719 533 1728.
WWW: http://delta.com/stiller/

S&S Dr Solomon's AVTK S&S International, Alton House, Gatehouse Way, Aylesbury, HP19 3XU, England.
Tel +44 1296 318799, fax 1296 318 755.
WWW: http://www.drsolomon.com/

Trend PC-cillin Touchstone Software Corporation, Huntington Beach CA 92648, USA.
Tel +1 714 969 7746, fax +1 713 969 1555.
WWW: http://www.trendmicro.com/

CONFERENCE REPORT

IVPC '96: Exponentially Yours

Ian Whalley

The *NCSA's International Virus Prevention Conferences* are always enjoyable: the locale (on the outskirts of central Washington DC) is very pleasant, the hotel comfortable – even the natives are friendly!

This year's conference was in the same venue as the one I attended last year, except that, in an effort to confuse the attendees, the hotel had changed its name from the Stouffer Concourse to the Washington National Airport Hilton. The latter is at least easier to pronounce, if considerably longer. However, it seems that the calibre of delegates was sufficiently high that no one was fooled by the hotel's cunning ruse, and everybody turned up.

Opening Salvos

Things kicked off on Sunday evening with registration, a strange ritual whereby delegates approach the staff and tell them they're here; to which staff confirm that they are. Following registration was a drinks reception (highlight here was definitely the fountain in the middle of the nibbles table). And so the first evening passed, with only the gentle interruption of a WildList committee meeting to disturb the slumbering editor.

On Monday morning, things really got started; at 8am, no less, which had all the Europeans wondering why Americans start their conferences so early in the morning. After a keynote session with Therese Padilla of *Command Software*, Dr Peter Tippett presented the results of an *NCSA* study into virus prevalence – statistics abounded for ninety minutes.

One of the most interesting comments from the panel (made up of representatives of the sponsors of the *NCSA* study, some of whom agreed with its findings, others of whom did not) came from Scott Gordon of *McAfee*. He had gone into a printing shop across the street from the hotel the day before to get some copying done, only to discover that all their PCs were infected with the Concept virus. A touch of reality at this point was welcome.

At 10:30, the tracks split, and every delegate entered frantic decision-making mode: which of the talks to attend? The speaker billed in track two, Judy Edwards, was unable to hold her presentation, so a last minute talk from Padgett Peterson filled the gap. In the other stream, Scott Gordon cashed in some of the credit he had gained earlier when, as part of his talk on evaluating network anti-virus products, he recommended that administrators should obtain their own large virus collections from the Internet with which to test the products… oh well.

Following Scott came Sarah Gordon (no relation…), who talked about the virus simulators; from those that demonstrate virus effects to those that claim they can be used as detection tests without having to use 'real' viruses.

Directly following Ms Gordon came the other half of the husband-and-wife team, Richard Ford, former *VB* editor. In a talk that was far from as doom-laden as its title suggested ('Why viruses are and always will be a problem'), he gave a brief overview of the current state of play in the virus world. Then came an excellent talk by Jason Khoury of the *NCSA*, who discussed the legal position of virus authors in States across America: an impressively well-researched paper.

Cessation of Hostilities

Following the usual late night/early morning in the bar, day two commenced at the not-quite-so-horrifying hour of 8:30, with an address from *Symantec*. Half an hour of product placement later, the 'Great Virus Debate' kicked off. The promised 'Anonymous Virus Writer' was not present (is absence the safest form of anonymity?), but George Smith (author of *The Virus Creation Labs*) was. Despite his presence, the debate was rather one-sided, albeit with intriguing asides into the fanciful world of computer viruses as weapons of war.

Paul Ducklin (*Sophos*) gave a humorous and seemingly spontaneous talk about different ways to protect a network against viruses, which was much enjoyed by everybody. Despite his overrunning, I managed to make the second half of a talk on generic decryptors by Carey Nachenberg (*Symantec*) – an enthusiastic and involving speaker with excellent animated demonstrations which greatly helped the audience to understand his fairly technical subject matter.

Joe Wells, the man behind the WildList, began the closing sessions of the conference with a thoughtful look at the relevance of detecting viruses which are not in the wild, and also the directions in which he intends to take the WildList over the coming months and years: it has already grown from a small project to an internationally-recognised semi-regular document containing the best information available about which viruses are a real threat.

Conclusion

One criticism of this conference was its level of commercialization: vendors buying lunch in exchange for a half-hour keynote does not strike an objective note. This aside, worthwhile material abounded. The trip was well worth it.

Now, where did I put that gin and tonic...

The paper presented at the conference by *VB's* editor, on viruses and *Windows 95*, can be viewed and downloaded from http://www.virusbtn.com/.

PRODUCT REVIEW 1

F-PROT Professional for NetWare

Martyn Perry

F-PROT Professional for NetWare (*FPN*) is the latest offering from *Command Software*. This is the server version of its well-known workstation product, and supports both *NetWare 3.x* and *NetWare 4.x*. In addition to virus detection, the package provides multi-server administration. *VB* last reviewed this product, then known as *Net-PROT*, about two years ago. How has it fared in the interim?

Presentation and Installation

F-PROT Professional for NetWare comes with two manuals and three diskettes for the server. It is licensed on a per-server basis, and workstation packs are also available, for DOS, for *Windows 3.1*, for *Windows 95* and for *OS/2*.

Workstation protection is provided using *F-PROT Professional for DOS and Windows*, which requires separate licensing. The alert management is handled separately by software, bundled with the package, called *AlertTrack Lite*.

Installation can be performed either from a *Windows*-based workstation or from the server console. The advantage of using the workstation installation is that the scanner initialisation file, F-PROT.INI, can easily be created and edited.

In both cases, the installation process creates a subdirectory called SYS:SYSTEM/F-PROT: this contains the bulk of the *F-PROT* files. The two NLMs (F-PROT.NLM and F-DELAY.NLM) are placed in the SYSTEM directory.

Installation of *AlertTrack Lite* is performed from the workstation. This is carried out in two stages: first, the workstation components are installed using a *Windows* set-up utility and the destination directory of *AlertTrack* components is defined. Second, a server is chosen to store the server components and the files are copied (supervisor or equivalent rights are needed to perform this installation).

Operation

The *FPN* program is loaded from the server console prompt (the CLIB must be version 3.12H or greater for *NetWare 3.x* and version 4.10G for *NetWare 4*). The following options can be added to the standard 'LOAD F-PROT' command where desired:

- SYS:<WorkDir> – this specifies a working directory. By default this is SYS:\SYSTEM\F-PROT.
- Buffers=X – this controls the number of concurrent real-time scans. The default here is five.
- Quiet – turns off the status screen.

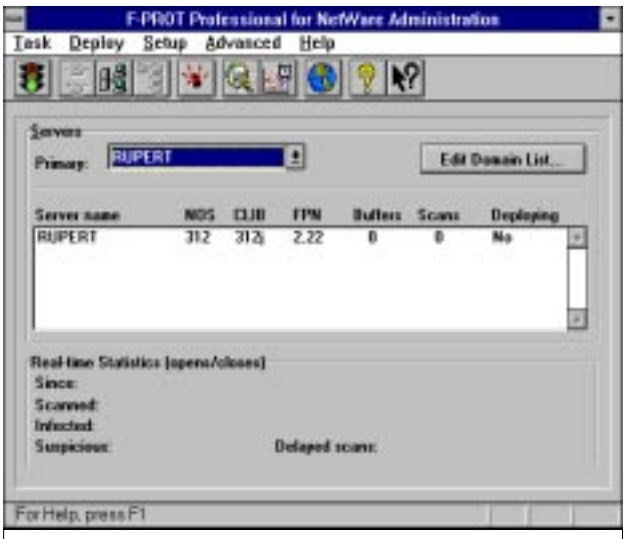
The number of buffers can range from 2 to 200: each requires 51KB of memory. If there are not enough buffers available, real-time scans will be delayed until the situation changes, and enough buffers have become available. It is possible to query the software to find out how many scans have been delayed since a specified date, but *FPN* can now dynamically adjust the buffer allocations, so there should not be a problem.

Once loaded, it is possible to control the operation of *F-PROT* from the console by using the command 'F-PROT' with various command-line options, as detailed in the manual. The alternative is to define settings in F-PROT.INI, the initialisation file. This file can be edited, either with a text editor or using *FPN's* administration utility.

The software organises servers into security domains. Each domain has a primary server, the default being the first one to be installed.

Software configuration is performed from the administration workstation using the *Windows* control utility. Configuration involves, among others things, selecting servers to include in the security domain, managing log files and reports, deploying upgrades to server and workstation scanners, and defining the various scanner modes. *FPN* has three modes of scanner operation: manual, real-time, and scheduled.

The manual scan checks the server on demand, using the current Manual settings. Such a scan can be started both from the Manual menu on the Management Workstation, and from the server console. In the version tested, however, it was not possible to stop a manual scan, either from the control utility or the server console – a curious omission.



F-PROT's configuration program is both powerful and easy to use. It allows control of all servers in the *F-PROT* domain.

The real-time scan allows files to be checked as they are opened or closed (the administrator may choose either or both) on the server.

The scheduled scan provides two types of timed scans. The first allows the administrator to specify a start time, and then the interval between scans (which may be hourly, daily, weekly, or monthly). The second offers a more precise specification of the time to elapse between scans – anything from one hour to ten months. A combination of these two would allow an administrator to keep one area of his server under close scrutiny, in addition to scanning all volumes once a night.

Configuration Options

For each mode of operation, various selections may be made:

- File extensions that are to be included in the scan: the defaults are COM, DOC, DRV, EXE, FON, OV?, SYS, PGM. Extra extensions may be added as necessary.
- Volumes, directories or files to be excluded from the scan. The defined defaults are any quarantine directory, SYS:BACKOUT.TTS, and the bindery files.
- Action on finding a virus: the product can delete the file, report to a defined list of users, rename the file, disinfect the file, move the file to a quarantine directory (this is the default action), or do nothing.

Each volume has its own quarantine directory; however, the default is set as \F-PROT\QUARANT.INE.

Administration

The *Windows* administration tool provides for configuration of its tasks to be divided into four main sections; 'Task', 'Deploy', 'Setup', and 'Advanced'.

'Task' deals with the definition of the scan options. 'Deploy' allows servers to be added or removed from the domain, and any updates can be applied and their deployment monitored.

'Setup' deals with various options for virus notification. These include:

- The configuration of *AlertTrack Lite* (if this is installed), determines who will receive a warning, and how it will be received
- Log file maintenance: options are available to control the size of the log file, which can be useful in ensuring that it does not become unwieldy and consume unnecessary space
- Reports: this option allows the administrator to define where reports of Infected Files, Scan Summaries and Scan Progress may be sent. This can include the System Console or the *F-PROT* screen, or simply their respective log files. In addition, it is possible to define a master log server, which can be different from the primary scan server.

AlertTrack Lite is the third-party NLM provided with *FPN*. This allows configurable messages to be sent in many ways: shown here is the *Windows* control program.

'Advanced' administration manages the default settings for the scanner. These are over-ridden by the settings for each particular type of scan – manual, real-time, and scheduled – but it is useful to be able to control some of the settings in a central location such as this. Finally, it is possible to revert to the hard-coded ('Ready To Go') defaults.

AlertTrack Lite

AlertTrack Lite is an additional NLM supplied with *FPN* which enables forwarding of alerts to selected users via a wide range of transmission facilities. The available communication methods are: Numeric Pager, Alphanumeric Pager, Faxware, *MHS* E-Mail, *Pegasus Mail*, SNMP traps and *Novell* network broadcasts.

The only option tested was the *Novell* network broadcast. This notifies the selected users about any change in the status of the scanner, including when it was being unloaded and when it is no longer providing protection.

The ALERTTRK.NLM is loaded at the server console, and must be present when the F-PROT NLM loads. If a pager is used, an additional NLM must be loaded. The *AlertTrack Lite* NLM is loaded only on to one server. One limitation seems to be that bindery emulation is required before this NLM can work on *NetWare 4.x*.

Detection Rates

The scanner was tested using the usual three test-sets; In the Wild, Standard and Polymorphic (see summary table for details). The undetected viruses were identified by using the 'move to quarantine directory' option, then observing which files were left behind in the virus directories.

The tests were conducted using the default (Ready To Go) scanner configuration supplied. Against the In the Wild test-set, the result was an excellent 99.7%: it only failed on

Concept – this because DOT is not a default extension. The sample was detected when *FPN* was asked explicitly to scan it. *Command Software* states that this omission has now been rectified.

Testing against the Standard test-set produced a creditable 93.4%. Unfortunately, the Polymorphic test only yielded a disappointing 58.0%.

Real-time Scanning Overhead

To determine the impact of the scanner on the server when it is running, 63 files, occupying 4,641,722 bytes (EXE files from SYS:PUBLIC), were copied from one server directory to another using *Novell's NCOPY*. Using *NCOPY* keeps the data transfer within the server itself and minimises network effects. The directories used for the source and target were excluded from the virus scan so as to avoid the risk of a file being scanned while waiting to be copied.

Because of the different processes which occur within the server, the tests were run ten times for each setting, and an average was taken. The test was performed for eight different sets of conditions.

First, four tests were run without *AlertTrack Lite* loaded, to establish the effect of the scanner alone on server performance. Then *AlertTrack Lite* was loaded and the same tests were repeated to gauge the impact of this NLM. The results are shown in the summary table at right, and the tests were:

- F-PROT not loaded. This establishes the baseline.
- F-PROT loaded, and on-access scanning set to scan on file close only (the default setting), but the immediate scanner not running. This tests the impact of on-access (real-time) protection.
- F-PROT loaded, on-access scanning configured as above, but with an immediate scan running.
- F-PROT unloaded. What is significant in this test is that F-PROT has been loaded – this checks whether or not the NLM leaves anything of consequence behind in server memory, which may slow the server down. In the case of *F-PROT*, the only file that is unloaded with the unload command is the main NLM – when *AlertTrack Lite* is active, this remains loaded.

The overhead of the scanner on its own is one of the lowest recently tested. When it is unloaded, it appears to release the bulk of the memory used. Having the alert facility loaded has little or no impact until the full scanner is running, in which case the overhead jumps significantly. This is probably due to communication between the F-PROT and the AlertTrack NLMs.

Conclusion

F-PROT Professional for NetWare is well put together, and has good documentation. The scanning options are comprehensive and their configuration is straightforward, and the multi-server environment is well handled. The inclusion of a separate alert system is also a good idea, as it allows the developers to focus their efforts on detecting viruses rather than on implementing notification features, which can easily be left to a third party. *AlertTrack Lite* is both flexible and easy to use.

The continued poor detection of polymorphic viruses must be cause for concern; however, the results of tests against the Standard and In the Wild test-sets were admirable. When the engine developers finally resolve their problems with polymorphs, *F-PROT Professional for NetWare* will be back where it belongs.

F-PROT Professional for NetWare

Detection Results

Test-set [1]	Viruses Detected	Score
In the Wild	303/304	99.7%
Standard	267/286	93.4%
Polymorphic	3851/7500	58.0%

Overhead of On-access Scanning:

Tests detail the time taken to copy 63 EXE files across 4.6MB; average time in seconds for 10 tests. On-access scanning was enabled only on File Close.

	Time	Overhead
AlertTrack Lite not loaded		
A. NLM not loaded	11.0	n/a
B. NLM loaded; no manual scan	11.5	4.6%
C. NLM loaded; manual scan	15.3	39.1%
D. NLM unloaded	11.0	0%
AlertTrack Lite loaded and active		
E. NLM not loaded	11.0	n/a
F. NLM loaded; no manual scan	11.5	4.6%
G. NLM loaded; manual scan	23.8	116.4%
H. NLM unloaded	11.0	0%

Technical Details

Product: *F-PROT Professional for NetWare*.

Developer/Vendor: *Command Software Systems*, 1061 East Indiantown Road, Suite 500, Jupiter, Florida 33477, USA. Tel +1 407 575 3200, fax +1 407 575 3026, BBS 1 407 575 1281, Internet: http://www.commandcom.com/.

Distributor UK: *Command Software Systems*, 4 Sloane Street, Knightsbridge, London SW1X 9LA. Tel +44 171 259 5710, fax +44 171 259 5753, BBS +44 171 259 5752.

Price: Per server, inclusive of quarterly updates. Single server licence £350; 2–5 servers: £317; 6–10 servers: £260.

Hardware Used: Server – *Compaq Prolinea 590* with 16MB RAM and a 2 GB hard disk, running *NetWare 3.12*. Workstation – *Compaq 386/20e* with 4MB RAM and a 207 MB hard disk, running DOS 6.22, and *Windows 3.1*.

[1]Test-sets: For a complete listing of all the viruses used in these tests, please refer to *Virus Bulletin*, January 1996, p.20.

PRODUCT REVIEW 2

Vi-Spy

Dr Keith Jackson

Vi-Spy is a well-established product which has been reviewed by *VB* three times before: June 1994, August 1992, and May 1990. I have written all these reviews – regular as clockwork, every two years, the latest *Vi-Spy* drops on to my doormat.

The product, which works under DOS or *Windows*, provides a scanner, memory-resident anti-virus software, checksumming features, and disinfection facilities. Also included was a disk marked 'Windows 95 Release Candidate #1'; however, I do not yet run *Windows 95*, so this review does not discuss that disk. Likewise, many of the *Vi-Spy* features are network-aware, but I have no means of testing these.

Documentation

The printed documentation comprises two A5 books, a *Guide to Operations* (154 pages), and a *Computer Virus Primer and Troubleshooting Guide* (67 pages). Both manuals seemed very similar to those provided for the last review: closer inspection revealed that the *Guide to Operations* was identical to that used two years ago. Not even a minor revision.

On the whole, even given the passage of time, *Vi-Spy's* documentation is very good, and very easy to use; however, it must be said that the *Windows* parts are not well documented. In fact, I do not remember seeing the words *Windows 95* in either book; however, the issue is discussed in READMEs and on extra sheets enclosed. This is perhaps not surprising, as the *Windows 95* product was not a full release at the time this review was written.

The *Computer Virus Primer* has been updated: it now dates from January 1995. I put the latest and the previous versions side by side and I'm hanged if I can spot a difference. Both Tables of Contents are identical, even to the length of each individual section. The differences must be minor indeed.

Installation

The DOS/*Windows* version of *Vi-Spy* occupied a single 3.5-inch (1.44 MB) floppy disk. When installed as described below, *Vi-Spy* placed 50 files, occupying 1.4MB, on the hard disk of my test PC. The documentation explains how to install *Vi-Spy* files manually should this prove necessary.

Installation of *Vi-Spy* onto a hard disk has always been very straightforward. When executed from floppy, the installation program scans 'critical system areas', then decides if this is an upgrade or a new installation. Amusingly, it found a copy: I had stored a copy of the master disk in a subdirectory. I just told the installation program to ignore it.

The user is asked whether the *Windows* part of *Vi-Spy* should be installed, and for the name of the subdirectory to hold the *Vi-Spy* files. Changes are made to WIN.INI and AUTOEXEC.BAT (if confirmed by the user). The memory-resident components of *Vi-Spy* are installed by means of extra lines added to the end of AUTOEXEC.BAT.

If *Vi-Spy's Windows* components have been requested, the installation program fires up *Windows*, requests that paths to desired subdirectory locations are specified, leaves the user in *Windows* to test things, and states that installation will only be complete when *Windows* is exited. On leaving *Windows*, the DOS installation program completes its tasks, and provides a summary of what has been done.

Scanning

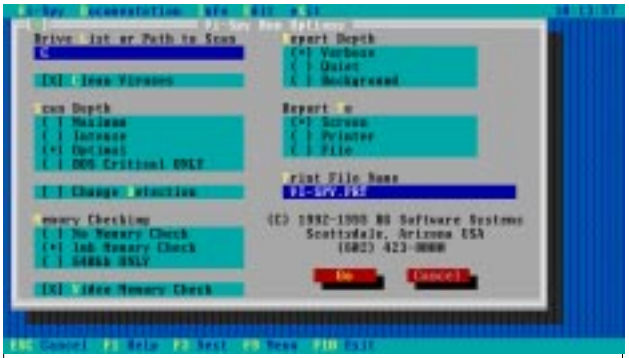
Vi-Spy claims knowledge of 4104 virus 'names' (each name will detect multiple viruses), a claim made subject to a caveat that users should beware the virus numbers game. The warning is useful, even though *Vi-Spy* is more restrained than other scanners regarding such claims. Of course, the number of viruses known to *Vi-Spy* has risen inexorably: in May 1990, it knew of 46 viruses. Two years later the total was 750, and two years after that, 1879.

The scanner is available as a command-line driven DOS program, a DOS program which uses drop-down menus, and a *Windows* program. They all use the same core engine.

Scanning Speed: DOS and Windows

In its default state, the DOS version of *Vi-Spy* reported that it scanned the hard disk of my test PC (714 files in total, 293 files scanned, 23.0 MB) in 1 minute 54 seconds. The time as measured by a stopwatch was 2 minutes 27 seconds.

The reason for this is clearer when memory checks are removed from the scan. Here, the time taken is 2 minutes 8 seconds, and time reported onscreen stays the same. The time reported onscreen seems not to include time taken to check memory.



The *Vi-Spy* configuration screen.

Vi-Spy has various options to tailor scanning. The default, 'Optimal', (aka 'Turbo') scans only parts of files 'where viruses are most likely to exist'. The fastest mode, 'DOS critical only', checks the computer's CMOS and the hard disk's boot sector and partition table – in four seconds!

A scan can also be 'Intense' (executable files are scanned byte by byte), which took 8 minutes 12 seconds, or 'Maximal' (all files scanned), which took 13 minutes 33 seconds. As above, scan times reported onscreen were just under 30 seconds less than those measured. 'Optimal' and 'Maximal' scans do not commence until the user presses a key. I do not know why this feature only applies to these two scans.

In comparison, *Dr. Solomon's AVTK* scanned the hard disk of the test PC in 4 minutes 21 seconds; *Sophos' Sweep*, in 7 minutes 38 seconds – considerably slower than *Vi-Spy*. Both of the products used for comparison have onscreen scan time less than measured scan time; however, the discrepancy between the two times is less than that of *Vi-Spy*.

Though *Vi-Spy* knows of many types of compressed files – e.g. ZIP, ARC – it only warns that they exist, and does not scan within them. The TSR, like all of its type, will pick up the files as they are decompressed by the user. I was, however, surprised that *Vi-Spy* said I had LZH files on my disk, it turned out that they were *Vi-Spy's* own – I have no LZH files on my test PC!

The *Windows* version of *Vi-Spy* is a front-end which garners settings for invoking the DOS *Vi-Spy* scanner. When this version was tested, scan times always increased over the figures reported for the DOS scanner, as expected.

Using a stopwatch, 'Optimal' scan time rose to 2 minutes 47 seconds, an 'Intense' scan took 9 minutes 59 seconds, and a 'Maximal' scan took 18 minutes 39 seconds. Each figure includes a 25-second discrepancy between the times shown above and times reported onscreen. In comparison with the DOS scan times reported above, 'Maximal' scan time is affected by *Windows* more than the other methods.

Detection

I tested the virus detection capability of *Vi-Spy* against the test-sets listed in Technical Details. Against the In the Wild viruses, and using default settings, *Vi-Spy* detected 281 of the 286 test samples (98.3%). It failed to spot the three samples of Markt.1533, and the two of Bosnia:TPE.1_4.

Against the Standard test-set, again using default settings, *Vi-Spy* did almost as well, detecting 260 of the 265 test samples (98.1%). The only viruses missed were the two samples of Phantom1, the two of Cruncher, and the single Kamikaze. All in all an excellent performance.

When *Vi-Spy's* settings were changed from its default values, the results were somewhat curious. The 'Intense' and 'Maximal' scanning methods are intended to provide a more in-depth scan for viruses. When run against the In the Wild test-set, an 'Intense' or a 'Maximal' scan detected, in each

The detection capability of *Vi-Spy's* memory-resident software was identical to that offered by its stand-alone scanner when run in 'Optimal' mode.

case, one virus *less* than when the Optimal scan was invoked. All nine viruses which were missed by the Optimal scan remained undetected, but for some strange reason, one EXE sample of One_Half.3544 also went undetected.

When the Standard test-set was used, things became stranger. Both Intense and Maximal scans detected the Kamikaze sample which the Optimal method had missed. However, the Intense scan failed to detect December_24th and one of the two Vcomm samples.

Therefore, the Maximal method tested against the Standard test-set was the only occasion when any of the more in-depth scanning methods performed better than the Optimal (Turbo) scanning method.

I suppose some of these odd results reflect the fact that all the *Vi-Spy* scanning methods get close to 100% successful detection on both test sets… though I'm hanged if I can explain why a more thorough look for a virus should actually perform *worse* than a quick (Turbo) scan.

Of the polymorphic samples, *Vi-Spy* detected 3988 of the 5500 test samples, a detection rate of 72%. The overall figure is quite good; however, it does hide a more complicated picture when the results are examined in greater detail.

Vi-Spy detected all samples of DSCE.Demo, Groove and Coffee_Shop, Pathogen:SMEG, and SatanBug.5000.A. All bar one sample of One_Half.3544 were detected: this was the very virus (albeit a different sample) which caught out the more in-depth scanning methods from the In the Wild test-set.

Detection of the other polymorphic viruses was variable, ranging from 89% of the Neuroquila.A test samples, to just 8% for the MTZ.4510 test samples. *Vi-Spy* also detected all of the twenty boot sector test samples.

The product has always been very good at detecting viruses, and nothing much has changed in that department. In its last review, it detected all the non-polymorphic test-set, 83% of the polymorphic viruses, and all the boot sector viruses.

Since those halcyon days, the test-sets, particularly the polymorphic, have expanded greatly, but *Vi-Spy's* detection capability has kept up admirably. The apparent fall in the polymorphic detection rate is almost certainly due to the fact that the test-set is much more demanding: the samples therein are more varied and much more difficult to detect with certainty.

Memory-resident Software

Three separate memory-resident programs are provided. The default, called RVS, checks program files as they are executed or otherwise accessed, prevents the user from accidentally booting from a floppy, inspects all floppy boot sectors, warns when a program is about to go memory resident or changes in size, and also prevents anything writing to the partition table or boot sector on the hard disk.

The second of these, RVSCDF, has all the features of RVS coupled with checksum verification for each executable program before execution. RVI_SPY, the last memory-resident program, checks floppy disks, attempts to become memory-resident, and changes in executable program size.

This software installs itself in one of seven ways, using the smallest possible 'footprint' in lower memory, and mixes of Expanded and Extended memory. On my PC, the message 'Running in EMS swapping mode' showed the storage strategy. When the line 'DOS=HIGH UMB' was not in CONFIG.SYS, RVS occupied 16 KB; RVSCDF 17 KB; and RVI-SPY, 7 KB. With this line, base memory usage dropped to zero. All figures refer to lower memory.

Against In the Wild and Standard test-sets, the detection capability of the memory-resident component was identical to that offered by *Vi-Spy's* stand-alone scanner in 'Optimal' mode. Few products can claim such a 100% match. Certainly, I have reviewed nothing capable of that in the past year or so.

Any memory-resident monitoring program which carries out tests before allowing access to a file must have an impact on system performance. I measured the overhead imposed by copying 40 executable files (1.25 MB) from one subdirectory to another. With no memory-resident software present, this took 23.8 seconds, rising to 46.4 seconds with RVS present, or 45.2 seconds under RVSCDF. With RVI-SPY present in memory, the time to copy the files dropped to 22.2 seconds. This I cannot explain: the result is, however, consistent.

Checksums

Vi-Spy can create a database of checksum information about each executable file present on a hard disk. The manual states that 17 bytes are required for each database entry.

The checksumming component of *Vi-Spy* adds, as expected, a huge amount of time to a hard disk scan when it is first executed and it creates its database of checksums. Reams of onscreen messages report the files added to the database.

After this first run, scan time increases only marginally. For instance, an 'Optimal' scan of the hard disk of my test PC rose to an onscreen reported time of 2 minutes 8 seconds, just 14 seconds more than the default time reported above.

The Rest

Vi-Spy claims to be able to clean viruses from infected files, but in common with my usual stance, I have not tested this. Infected files should be replaced with known clean copies.

Vi-Spy still maintains some of its files in a subdirectory (RGVSPYDB) in the root of drive C. This is a nuisance, but unlike previous versions, the files can now be placed in any desired subdirectory by using a command-line switch.

Vi-Spy includes a scheduler which can be used to invoke a scan at any desired interval. The installation program uses this to ensure that a scan is carried out at least daily – assuming that a PC is rebooted at least once every day.

Conclusions

Vi-Spy lives or dies by its scanning ability (from either the stand-alone scanner, or the memory-resident software). This was true four years ago, it was true two years ago and it remains true today. Given some of the more complex anti-virus software I have seen in recent years, *Vi-Spy's* simplicity stands out, at least to me, as a virtue. It is refreshing to review an anti-virus product with less features than a modern word processor.

Vi-Spy has long had an enviable record as far as its memory-resident software is concerned. This remains so. Its detection rate capabilities are particularly impressive.

I previously concluded that *Vi-Spy* is 'simple to understand, easy to use, and fleet of foot in searching for virus signatures on a disk'. That remains true, although it is a shame that the speed impact of the resident software is as large as it is.

The results show that this product was at least twice as fast at scanning the hard disk of my test computer as the packages used for comparison. It also performed well at virus detection. The percentage detected has dropped slightly in recent years, but this is more to do with the explosion in virus numbers, and the expansion of the *VB* test-set, than anything else.

In summary; *Vi-Spy* was 'heartily recommended' in my last *VB* review. It still is.

Technical Details

Product: *Vi-Spy v14*, Rel.02.96. No serial number visible.

Developer/Vendor: *RG Software Systems Inc*, 6900 East Camelback Road, Suite 630, Scottsdale AZ 85251, USA. Tel +1 602 423 8000, Fax +1 602 423 8389, BBS +1 602 970 6901.

Availability: PC with an 8088 processor or above with 256 KB available RAM, and 1.5 MB free hard disk space. *Windows* components require higher specifications. Memory-resident components require *MS-DOS v3.2* or above.

Price: US$149.95 for a single-user licence. Corporate discounts are available, starting at US$1043 for a 25-user licence.

Hardware used: A *Toshiba 3100SX*; a 16 MHz 386 laptop with one 3.5-inch (1.4 MB) floppy disk drive, a 40 MB hard disk and 5 MB RAM, running under *MS-DOS v5.00* and *Windows v3.1*.

Viruses used for testing purposes:

For a detailed listing of the contents of the Boot Sector test-set, see *VB*, March 1996, p.23. The Standard, Polymorphic, and In the Wild test-sets are listed in *VB*, January 1996, p.20. For a complete explanation of each virus, and the nomenclature used, please refer to the list of PC viruses published regularly in *VB*.

SUBSCRIPTION RATES

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, Oxfordshire, OX14 3YS, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email editorial@virusbtn.com
CompuServe address: 100070,1340
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA

Tel +1 203 431 8720, fax +1 203 431 8165

END NOTES AND NEWS

S&S International is presenting **Live Virus Workshops** at the *Hilton National* in Milton Keynes, Bucks, UK on 1/2 July, 2/3 September, and 7/8 October 1996. The company has also announced the launch of *WinGuard for Windows NT*, classed in a recent press release as a 'major enhancement' adding on-access scanning to the extant *Dr Solomon's Toolkit for Windows NT*. Details available from the company: Tel +44 1296 318700, fax +44 1296 318777.

Former *VB* editor Richard Ford, who left the journal to become Director of Research at the *NCSA*, has once again moved on, taking up a post with his wife, Sarah Gordon, at *Command Software* in Florida, USA. Further information is available from the company on Tel +1 407 575 3200, fax +1 407 575 3026.

***Sophos Plc's* next anti-virus workshops** will be on 24/25 July and 25/26 September 1996 at the training suite in Abingdon, UK. The two-day seminar costs £595 + VAT. One single day may be attended at £325 + VAT (day one: Introduction to Computer Viruses; day two: Advanced Computer Viruses). For information, contact Julia Line, Tel +44 1235 544028, fax +44 1235 559935, or access the company World Wide Web page (http://www.sophos.com/).

Precise Publishing Ltd will be holding more **Live Virus Workshops** (12 June 1996, 17 July 1996). Details are available from the company; Tel +44 1384 560527, fax +44 1384 413689.

Reflex Magnetics has several courses coming up: **Live Virus Experiences** (12/13 June, 9/10 October), The Hacking Threat (24-26 July), Internet Security and Firewalls (30 May, 22 July), and DTI Security Codes of Practice (31 May). The company has also awarded the American distribution rights to its access control package, *Disknet*, to New York-based Global Data Security Inc. For further information, contact Rae Sutton: Tel +44 171 372 6666, fax +44 171 372 2507.

Australia-based ***Cybec Pty* has announced plans to open offices in the UK** by the end of June. Billed as 'the first of *Cybec's* international offices', it follows rapid expansion in its home territory: the company now has three offices there. Further information on the move is available from *Cybec*; Tel +61 3 9521 0655, fax +61 3 9521 0727.

A native Java virus scanner for Java applets sent over the Internet has been released by *Symantec Corporation.* An extension to *Norton AntiVirus*, it is said to provide real-time protection and monitor for virus activity within any Java-supporting Web browser. Details on this innovation can be obtained from the company's Web site; access http://www.symantec.com/.

Following *IVPC 96*, the *NCSA* is hosting another event; the ***Web, Internet Security and Firewall Conference***, to be held in San José, California from 30 September–1 October. Details available from the *NCSA's* WWW site: www: http://www.ncsa.com/fw96west.html/; or email fwcon96west@ncsa.com.

ON Technology, recent acquisitors of *Thompson Network Software*, has released a new anti-virus product in the UK. ***Macro Virus Track 6.0***, reviewed in last month's *VB* Macro Comparative Review, is said to detect and remove five Word macro viruses and one Trojan horse. Information can be obtained from WWW – http://www.on.com/, or via the Internet, email info@on.com.

The world's first full-scale IT exhibition in Cyberspace, *Virtex*, will open on the Internet in October 1996, and will be sponsored by ElectronicTelegraph, the Internet version of the UK national newspaper The Daily Telegraph. The exhibition will run for eleven months: companies involved include *IBM*, and *Digital.* The exhibition will run on a 486 with 8MB RAM. Further information is available from ElectronicTelegraph – WWW: http://www.telegraph.co.uk/.