ISSN 0956-9979 JANUARY 1996

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

Editor: **Ian Whalley**

Assistant Editor: **Megan Skinner**

Technical Editor: **Jakub Kaminski**

Consulting Editors:

Richard Ford, NCSA, USA
Edward Wilding, Network Security, UK

IN THIS ISSUE:

• **Highly interesting.** Harold Highland is one of the best-known figures of the early days of computing. *VB's* interview recounts some of his experiences – see p.6.

• **DOS sparring.** Twenty-five products have participated in the latest DOS scanner comparative: how do they fare? See p.13 for the tests and results.

• **Another **** virus!** FITW is the newest in the virus trends; stealth, polymorphic, and multi-partite. What risks does it pose to the user community? See p.9 for an in-depth analysis.

CONTENTS

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

EDITORIAL

Off with his Head!

Last month, legal history was made here in the UK when Christopher Pile, self-confessed author of the viruses Pathogen and Queeg and the encryption engine known as SMEG (Simulated Metamorphic Encryption Generator), was jailed for 18 months. This event received wide coverage even outside specialist journals, putting in appearances on national TV, radio and in the newspapers. However one looks at it, this was an event of no small importance – the first person in the UK to be given a custodial sentence for writing and distributing computer viruses.

“ it has always been difficult to assign a cost, in simple monetary terms, to the results of computer crime ”

It is undoubtedly the case that Pile is guilty of the crimes for which he was prosecuted – in addition to the array of damning evidence against him, he pleaded guilty. However, when I heard the sentence, my immediate reaction was that it was over-harsh, perhaps even inappropriate. This has also been the reaction of a number of people to whom I have spoken over the last few weeks, so it seems that the time is right to have a closer look.

In his summing up, Judge Griggs, presiding, made the point that the five year maximum penalty the law allowed should be reserved for those who commit the crimes for some form of monetary or material gain. This seems an eminently reasonable viewpoint – certainly, had Pile (as an extreme example) written SMEG in an attempt to cripple the security systems of the *Bank of England* in order to facilitate grand theft, a more serious penalty would have been in order. This was, of course, not the case.

An important issue in any case of this type is the question of the damage caused – it has always been difficult to assign a cost, in simple monetary terms, to the results of computer crime. The case of the E911 document stolen from *BellSouth* by the hacker, Prophet, in 1988 is a classic example. Whilst the specific issues in that instance were somewhat different from those under consideration here, the way in which such costs can be exaggerated is clearly shown. The three estimates of damage in the Pile case range from two at £1,000 to a third at £250,000. Much has been factored into this final figure, including the estimated loss of profits due to the delayed release of a new product.

However, in the case of a virus, there is an added difficulty when calculating the cost. To use a phrase which cropped up in court a number of times, once Pandora’s box has been opened, it can never truly be shut again. The software Pile wrote is freely available from a number of sources on the Internet and from BBSs around the world. It cannot be taken back from the underground.

There was also a clear intent to distribute the virus – the deliberate infection and subsequent uploading of utilities to BBS systems in such a way as to encourage their download by unsuspecting users can be in little doubt. In spite of this, the viruses are not prevalent in the real world today – Pathogen and Queeg survive on the *WildList* by the narrowest of margins, and the engine has not become as widely used as its author would perhaps, at least when he wrote it, have liked it to. It seems likely that minimal damage will result, although Pathogen’s generation counter is forever ticking forward…

Given these facts, Pile clearly deserved to be punished in some way. But did he deserve to go to prison for eighteen months? This is a difficult question. The sentence may be out of proportion to those imposed for other, seemingly more serious, offences – however, it is all too easy to fall into the trap of drawing parallels between the punishments for crimes which are completely different. Such comparisons are hard to make; each type of crime must stand alone, for it would never be simple to assign relative ‘levels of severity’ to individual types of crime.

Quite apart from punishing the individual, sending Pile to prison will set an example upon which other virus authors in the UK may ponder – this game they play is suddenly more complicated; now it’s a game which you go to prison for playing! And, despite my initial reaction to the contrary, I now believe that it was the right magnitude of sentence… perhaps it is just as well that Jeremy Griggs is a judge, and I am an editor.

NEWS

McAfee to Target UK

In recent months, there have been a number of announcements in the press concerning acquisitions by American-based software company *McAfee Associates* both in the United Kingdom and in Europe.

Peter Watkins, *McAfee* Vice-President (International), stated in a press release that the company has ‘ambitious plans for Europe and in particular the UK’. The company intends to raise its profile significantly in the anti-virus and the network management markets, with the ultimate ambition of becoming the number one supplier of such software in the United Kingdom.

Firms acquired by *McAfee* include *Saber Software*, *Brightworks*, and *ADS*. As reported in a recent edition of *VB*, *IPE*, a distributor of *McAfee* products in the *UK*, has now also been added to the ever-growing *McAfee* stable.

Other former distributors in various corners of Europe (*Assurdata* in France, *Kirschbaum* in Germany) have also become part of the multi-national corporation, and a *McAfee* European anti-virus research centre has been established in the Netherlands.

Information on the strategies outlined above is available by contacting Paul Slattery (sales), Caroline Kuipers (European Marketing Manager), or Fiona Dineen at *McAfee*, on Tel +44 1420 542598 ▮

Dear Santa...

In the middle of December, *Ziff-Davis UK Ltd* sent out an electronic Christmas card. The card took the form of a *Windows* program, and was distributed on a 3.5-inch, 1.44MB floppy disk. Unfortunately, said disk contained not only a Christmas greeting, but also an unwanted extra ‘present’: a fully functional sample of Parity_Boot.B.

The disk is recognisable by its label. This reads, in a large, script-like font, ‘Merry Christmas’, and then in smaller letters ‘Go to Program Manager, File/Run menu and type A:RUNCARD’, ‘Computer Life, Ziff-Davis UK Limited’.

This latest incident comes hot on the heels of the heels of one reported by *VB* in its November 1995 edition [*see p.3*], when the Sampo virus was distributed by *PC Magazine*, another *Ziff-Davis* publication. This case also has added irony: the magazine with which the diskette appeared is called *Computer Life*.

Next year, might *VB* recommend the more traditional Christmas *card*? Apart from the reduced risk of it carrying a virus, data corruption is less likely, and the old format is compatible with all known systems ▮

Prevalence Table - November 1995

Virus	Incidents	(%) Reports
Form	42	11.2%
Parity_Boot	39	10.4%
AntiEXE.A	34	9.1%
Concept	29	8.5%
Empire.Monkey	21	5.6%
Ripper	20	5.3%
AntiCMOS	19	5.1%
Sampo	16	4.3%
Junkie	15	4.0%
NYB	10	2.7%
Stoned.Angelina	9	2.4%
Jumper	8	2.1%
Telefonica	8	2.1%
V-Sign	7	1.9%
BUPT	6	1.6%
Unashamed	6	1.6%
Manzon	5	1.3%
Stoned.June_4th.A	5	1.3%
EXEBug.A	4	1.1%
Helloween	4	1.1%
Natas	4	1.1%
Stealth_Boot.C	4	1.1%
Boot.437	3	0.8%
Bye	3	0.8%
Cascade	3	0.8%
Delwin	3	0.8%
Diablo	3	0.8%
J&M	3	0.8%
Stoned	3	0.8%
Int7f-e9	2	0.5%
One_Half	2	0.5%
Quicky	2	0.5%
Stoned.No_Int	2	0.5%
Taipan	2	0.5%
Trojector	2	0.5%
Other	26	7.0%
Total	374	100%

* The Prevalence Table includes reports of one of each of the following viruses: Crazy_Boot, Da’Boys, Dark_Avenger.2100, Darth.409, Disk_Killer, Flip, Green_Caterpillar.1575.B, Jackal, Kenya, Maltese_Amoeba, Mutagen, November_17, Ontario.1024, PSMPC, Rainbow, Russian_Flag.A, Sibel_Sheep, Stoned.Kiev, Stoned.Mongolian, Swiss_Boot, Tamsui, Tequila, Urkel, Vulga, WBoot.A .

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 December 1995. 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	**M**	Infects Master Boot Sector (Track 0, Head 0, Sector 1)
D	Infects DOS Boot Sector (logical sector 0 on disk)	**N**	Not memory-resident
E	Infects EXE files	**P**	Companion virus
L	Link virus	**R**	Memory-resident after infection

AirRaid.330

CR: An appending, 330-byte virus containing the plain-text string ‘Air Raid’ at the end of files. Infected programs have the signature ‘AR’ at offset 3. The virus does nothing apart from replicate.

```
AirRaid.330         B872 61CD 210A C075 4C56 33FF 1E8C C848 8ED8 BB1A 00C6 054D
```

Andris.843

CR: An encrypted, appending, 843-byte virus, which contains a destructive payload: the virus overwrites the contents of the first partition of the hard disk between 6 and 16 August. It contains the texts ‘<ANDRIS>’, ‘????????COM’ and ‘*.com’.

```
Andris.843          EB05 5380 7710 ??90 B9?? 0383 C31F 908A 0734 ??88 0743 E2F7
```

Annihilator.596

CN: A family of appending, encrypted, direct infectors. On 2 February, 4 April, 6 June (etc), when they fall on Fridays and Saturdays, the virus halts a PC after displaying the message: ‘This file is infected with Annihilator by [HtTM] - 10.08.1991/1993’.

```
Annihilator.596     60E8 0000 582D 8B01 958D B6AC 01E8 0200 EB13 B917 0918B FEBA
```

Annihilator.599

CN: After infecting a file, on 2 February, 4 April, 6 June (etc), when they fall on Fridays and Saturdays, the virus hangs the PC after displaying the message: ‘Your harddisk has been infected with HtTM’s Annihilator v3.21 - 10.08.1991/1993’.

```
Annihilator.599     60E8 0000 5B81 EB0A 018D B72B 01E8 0200 EB13 B918 018B FEBA
```

Annihilator.603

CN: After infecting a file, on 2 February, 4 April, 6 June (etc), the odds are 1:10 that the virus will hang the PC after displaying the message: ‘our harddisk has been infected with [HtTM’s Annihilator v3.00 - 10.08.1991/1993]’.

```
Annihilator.603     0195 8DB6 2C01 E802 00EB 153E 8B96 5E03 B919 018B FEAD 33C2
```

Annihilator.607

CN: After infecting a file, on 2 February, 4 April, 6 June (etc), when they fall on Fridays and Saturdays, the virus hangs the PC after displaying the message: ‘— Your harddisk has been infected with — [HtTM’s Annihilator v3.10 - 10.08.1991/1993]’.

```
Annihilator.607     60E8 0000 582D 0A01 958D B62B 01E8 0200 EB13 B91C 018B FEBA
```

Annihilator.610

CN: After infecting a file, on 2 February, 4 April, 6 June (etc), when they fall on Fridays and Saturdays, the virus hangs the PC after displaying the message: ‘— Your harddisk has been infected with — [HtTM’s Annihilator v3.10 - 10.08.1991/1993]’.

```
Annihilator.610     60E8 0000 582D 0A01 958D B62B 01E8 0200 EB13 B91E 018B FEBA
```

Annihilator.673

CN: After infecting a file, on 2 February, 4 April, 6 June (etc) on Thursday, Friday or Saturday, the odds are 1:20 that the virus will hang the PC after displaying the message: ‘— Your harddisk has been infected with — [HtTM’s Annihilator v3.00 - 10.08.1991/1993] The slightly polymorph COM infector Virus!’.

```
Annihilator.673     E800 0058 2D09 0195 8DB6 2E01 E804 00EB 17?? ??B9 3C01 8BFE
```

AtomAnt.564

CR: An appending, 564-byte virus containing the plain-text signature ‘AtomAnt v1.00’ at the end of every infected file. When active in memory, the virus prevents a user from deleting files. Instead, on every such attempt, it shows this message: ‘Hát igazán nem kedveltek bennünket ?’.

```
AtomAnt.564         3D00 4B74 3F3D FF35 740F 80FC 4174 0F80 FC13 740A 2EFF 2EEC
```

Avalon.814

CER: An appending, 814-byte virus with a destructive payload: on the 31st of each month, it overwrites the MBR, installs a new Int 1Ch service routine, and displays the message: ‘AVALON por OSoft’.

```
Avalon.814          B4FF CD21 80FC EE74 0683 EE06 E808 00BF 0001 57C3 B003 CF06
```

Bad_Sectors.3627

CER: A stealth, appending, 3627-byte new member of the Bad Sectors family. It contains the strings ‘COMEXE’, ‘SCAN’ and the text ‘Bad Sectors 1.3’ located at the beginning and end of the virus code.

```
Bad_Sectors.3627    1E1E 5848 8ED8 803E 0000 5A75 4681 3E03 0040 1172 3EB4 30CD
```

Beta.4028

CER: A polymorphic, appending, 4028-byte virus. On 17 November and 6 February, the virus displays the message: '** BRAIN2 v2.00beta - upgrade from POJER **', 'BETA tester, thank you', '...have a nice day in cyberspace...', 'This crazy program is (c) 12/93 by SB'. The template detects the virus in memory.

```
Beta.4028          5150 B800 FABA 4559 CD16 81FF 5945 7508 B801 FABA 4559 CD13
```

Cli&Hlt.1345

CER: An encrypted, appending, 1345-byte virus with stealth capabilities and anti-debugging features (e.g. hooking Int 01h and Int 03h). It contains the plain-text string: '** CLI&HLT Hackers Group **'.

```
Cli&Hlt.1345       5651 53B8 ???? B946 0333 DB2E 3180 5A00 43E2 F85B 595E 58CF
```

Cosenza.2034

CR: A polymorphic, circa 2034-byte virus containing a payload which triggers in October. When an infected file is run, the virus overwrites the first two sectors of the first hard disk and displays the text: 'the [BillGates] Virus is power-on!! Have you got a BACKUP of your HD??!? [BillGates] Virus: RamResident .COM Infector Semi-Stealth Virus, Variable Crypto-Key and, Polymorphic Encryption!! (c) Microsoft Written in COSENZA (Italy, April 1995) Freddie (Mercury) lives ... somewhere in time'. The template below detects the virus in memory.

```
Cosenza.2034       86DB 3D21 3575 0B2E 8B1E 2B01 2E8E 062D 01CF 3D21 2575 0B2E
```

Cosenza.3212

CER: Another polymorphic, circa 3212-byte-long virus from the Italian town of Cosenza. In November, after 10am, the virus hangs the system after displaying the message: '[C*O*S*E*N*Z*A] Virus! QUESTO ViRuS e STATO DISTRIBUTADA: (COMPUTER POINT <-> COSENZA, c.so d'Italia, 0984/481166) (CALIO' <-> COSENZA, via N.Serra, 0984/38861) (COMPUTER DISCOUNT <-> COSENZA, via Rodota 15,0984/71230) Advanced Semi-Stealth Virus with <P.V.E.> (P)olymorphic (V)ariable (E)ncryption * C-y-b-e-r L-o-r-d * '. The virus can be detected in memory with the same template as variant 2034.

Cybercide.2299

CR: An appending, 2299-byte virus with several payloads, including overwriting the hard disk and visual and sound effects. It contains the plain-text messages: 'I SHALL FEAR NO EVIL', '>>>A.N.O.I.<<<<', '** CYBERCIDE **', 'FLOATING THROUGH THE VOID', 'COPYRIGHT (c) 1992-93 A.N.O.I. DEVELOPMENT'. Another string read backwards says: 'never be CRUEL to ENTITY'.

```
Cybercide.2299     B822 DD5D CD21 3D33 3D75 058D 567C FFE2 B821 35CD 2189 9E8E
```

Desperate.633

CN: An appending, 633-byte virus containing the encrypted text: '*** Desperate chemist ***' and a plain-text string: '*.com'. The virus infects COMMAND.COM by overwriting its last 633 bytes (usually zeros) and setting a file time-stamp to 60 seconds. Other COM files are infected by appending the virus code. The virus will occasionally (based on an internal counter) corrupt the CMOS data and after 255 runs, it overwrites the contents of the MBR on the first hard disk.

```
Desperate.633      BF00 01FC A5A5 8BFE B078 E670 E471 3C4D 7511 B079 E670 E471
```

Devastator.636

CEN: An encrypted, appending, 636-byte virus which contains the text: 'Devastator', 'Lame Virus #11'. The virus corrupts EXE files, as it incorrectly sets pointers within the EXE header).

```
Devastator.636     E800 005D 81ED 2001 8DB6 3701 89F7 B932 01AD 35?? ??D0 C8AB
```

Finnish.378

CR: An appending, 378-byte new member of the Finnish family. It can be detected with the template used for the 357-byte variant [*see VB, October 1992 p.4*]. It is impossible for the user not to be aware of the infection since, while active in memory, the virus produces a long beep before each COM file is executed.

Finopoly

CR: A polymorphic, appending, circa 2300-byte long virus which targets some anti-virus products. It contains the encrypted text: 'EBOLA@RNA1.polymorphic.virus.FI' and 'DISCLAIMER: THIS PROGRAM HAS BEEN SPREAD "AS IS" SO NO WARRANTY OF ANY KIND'. This is a part of the message displayed if a day number is the same as a month number (beginning with 1 January 1996). The following template detects the virus in memory.

```
Finopoly           CF52 B42A CDF1 81F9 CB07 7607 3AD6 7503 E979 015A B800 3DCD
```

LTS.271

CN: An appending, 271-byte direct infector containing the plain-text message: '(C) Long Tall Silver'. The payload triggers on the third day of every month, displaying a message and corrupting data on the disk in the default drive.

```
LTS.271            81C2 2401 B440 B9EB 0053 CD21 5B5A 81C2 3302 B440 B924 0053
```

LTS.279

CN: A prepending, 279-byte direct infector containing the plain-text message: '(C) Long Tall Silver'. This text is displayed on the second day of every month.

```
LTS.279            E87E 00A3 FC01 B440 B929 01BA E8FD E870 00B8 0042 33C9 33D2
```

OpalSoft.390

CN: An appending, 390-byte direct infector marking all infected files with the 62 seconds time stamp. The virus contains the encrypted text: 'OpalSoftCHKLIST?.*'.

```
OpalSoft.390       DF83 C31F 8B17 8BDF 83C3 02B9 1B00 8B07 33C2 8907 43E2 F7C3
```

Trivial.OW.119

CN: A simple, overwriting, 119-byte virus containing the string '*.com'.

```
Trivial.OW.119     B4FE CD21 80FC 5274 1BB8 2135 CD21 891E 7701 8C06 7901 BA2B
```

INSIGHT

High on the Happy Side

Harold Joseph Highland is a name known to all but the youngest generation of computer users. He has seen computing evolve from its earliest birth pangs in the 1950s to the highly sophisticated scene of today, and has been involved in every stage of its development.

Unlike many who remain in one field all their lives, Highland has had a varied career. He began in the military as a chief machine gun officer, later serving in cryptographic analysis and air topology intelligence, and carrying out special assignments in counter intelligence analysis. He also became Provost Marshall. He admits to entering the intelligence field some 67 years ago, when he was in high school and spent a day with Count Felix von Luckner, the WWI Q-boat commander, who was visiting New York.

Starting Out

He began his professional life as a physicist: by his own admission, he was 'lousy' at calculus, and decided not to pursue physics as a career. In addition to the military, and work on *The New York Times*, Highland was a research statistician, an economist, a management consultant, a methods engineer, a magazine editor and publisher, and owner of an advertising/PR organization.

He was also Dean of Roth Graduate School at *Long Island University*, Associate Dean of Connelly College, director of various computer centres, a consultant, a teacher, and has written 27 books. He has worked with government agencies, and even today serves as computer security consultant to the Beijing government, and advisor to several others.

'I was a social scientist, and primarily a writer, editor and publisher. I did all sorts of work, in publishing and other areas.' To quote Esther, his wife of over 55 years: 'He was in so many different jobs because he couldn't hold one!'

Highland's own modest explanation differs: 'Once I got in a job, got proficient at it and won a top award, there was nothing more there for me. In fact, if it hadn't been for my wife, I'd have probably switched jobs more often than I did.'

Coming to Computing

His earliest experience with computers was on becoming Dean of Roth Graduate School in 1957. He examined the facilities, and on asking where the computer was, was told: 'We don't have any'. Never one to sit back and let life happen, Highland made an appointment to see the university president ('Who looked at me as though I had come down from Mars,' recalled Highland). He was informed in no uncertain terms that there was no money to buy or rent one.

'So, knowing some of the members of the board of trustees of the university, I made sure that I had lunches or dinners with them, and brought up the question of a computer.'

He scored a Pyrrhic victory: a computer was purchased, but no budget allocated for staff. Undaunted, Highland, with one of the graduate students from his school, set about learning how to operate the machine, an *IBM 650*.

Eventually, Highland took up a post at the *State University of New York's Technical College*, originally to set up their computer system, and to inaugurate a very specialised training program in computing – by this time, his knowledge of computers had extended to include the *IBM 1620* and *360*, in addition to the *650*.

He spent a great many years in tertiary education, culminating in the 1970s with a stint as a Fulbright Professor in Finland (1970), receiving the State University Chancellor's award for Excellence in Teaching (1976), and promotion to the rank of Distinguished Professor (1978).

'This title is one step above a full professor,' explained Highland. 'It meant extra money – but more important was the recognition of my work in computing, which was modelling and simulation.'

The Next Generation

An important breakthrough for computers in the 1970s was the development of the microprocessor: 'When the first micro came along,' recalled Highland, 'I had two students who took a piece of plywood, three feet by four feet, and we built our first microcomputer by stapling the wires onto the plywood. We used a five-and-a-half inch screen, and all our entries were in octal.

'I'm very sorry that I don't still have that board,' he commented ruefully. 'I should have taken it from the school when I retired. It was a lot of fun. We had to do the octal translation – I can just see people doing that today!'

Eventually, retirement beckoned. 'My wife's university,' Highland related, 'had a government rule that you had to retire at 65. When she reached 65, and said she was going to have to retire, I said, "My, I'd like to as well." I did take an appointment with one of the other schools for a year, but even before I left I started thinking about what I would do when I retired.

'If I was going to teach, I wanted to be paid as a full-time faculty member or not at all. Part-time teaching does not get terrific pay. So I looked over the field, and thought about one of the areas I was in – I taught computer security; I had created a course at the university on computer security and encryption. I realized that there was no publication in the field. And I said, what we need is a referred publication.'

On Writers and Publishers

Highland's next challenge was starting a computer security journal for Netherlands-based *Elsevier*. He and his wife ran *Computers and Security* for almost its whole first decade; he as editor, she as managing editor. The job, he found, was not the easiest of tasks. For articles to be meaningful, he felt, the writer must know his subject inside and out.

'One of the things I found,' he remembered, 'and I'm not sure it isn't true today, is that the articles from the academic world were generally poor, either because their bibliography was really obsolete, or because their ideas had been superseded in the real world, and they didn't know it. I found that our rejection rate on the academic side was as high as 60%.

'The industrial papers had more solid meat to them, but how practical they were… I decided to found an editorial board to solve the problem. Now this board, when I ran it, picked papers based on the individual's knowledge in the field. If such knowledge was lacking, a paper was not considered.'

Once a paper was passed 'fit for publication', Esther Highland took over, copy-editing and liaising with the author, in some cases virtually rewriting articles. After her work was complete, the article would then go to the reviewers, and finally back to Highland.

'Those sort of procedures, I feel, are necessary,' he said. 'You don't send an article to a reviewer unless it's been properly edited. Many authors have said to me, "When your wife finished editing my copy, my paper was really good".'

Highland's 'adventures' in publishing lay not only in computers: his first experience in the field was to edit the Do-It-Yourself Encyclopedia in the 1950s, a huge American success, which is still being sold today.

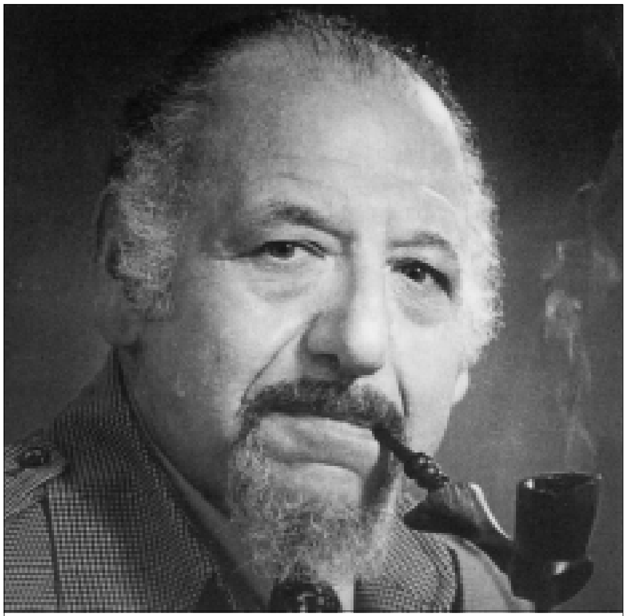
The Advent of Viruses

Highland became involved with computer viruses, as an interested bystander in February 1984: 'I was speaking at a security conference, and had a PC with me that needed data input. There was a young fellow sitting there in a shirt, with no tie. So I said: "Would you stand by and help on the PC?" When we finished, he was asking me questions, and I saw he was wearing a speaker's badge. I asked about his paper, and he told me, saying: "You probably won't know anything about it; it's on computer viruses." It was Fred Cohen.

'So I went to his lecture, and realized he had something. As I was still editor of *Computers and Security*, I asked him to do a piece for us on his paper. He started sending me material. It was not until several years later that he told me that he had never handled a PC before that conference.'

When is not a Problem a Problem?

On his early experiences with viruses, Highland stated that, having some knowledge from the security viewpoint, he had always felt that the problem with viruses was somewhat overstated: to most security people, PC viruses were simply a minor annoyance. He also believes that the media are in no small part to blame for the hysteria continually aroused by viruses, citing as a case in point an interview he gave to a *New York Times* reporter.



Harold Highland has been involved with computers for decades, and is still 'going strong', serving as consultant to many governments and organizations.

'There was a freelance reporter,' he related, 'who was doing a story on the Brain virus, and he called me to ask what I knew about it. At that stage, I was waiting for a copy of the virus to be shipped to me; it was due to arrive that day. I told him to ring me back later, after I would have had a chance to look at it.

'When he did, I still hadn't looked at it, and he had deadlines to meet, so he couldn't wait for me. "Look," he said, "this Pakistani virus – when you were at the graduate school, you had a lot of Indian and Pakistani students. You know them. The virus asks for a ransom – how much would they ask for?"

'So of course I said I didn't know; that I hadn't seen the virus, and he said: "Think of your students – how much would they ask for?" '

I said it would depend on the students: really poor ones might ask for twenty bucks. A little greedy, they might want two hundred. The really greedy ones would demand two thousand. I couldn't give him a precise figure. I told him that. Then the *Times* came out: "Brain virus asks for ransom of two thousand dollars" – and quoted me by name.

Highland became aware of the idea of macro viruses as early as 1987/88, when he had reports from colleagues in Europe of a spreadsheet virus: 'I got a call from a company in the US about the same thing; I said I couldn't imagine it being in a spreadsheet unless it was in the executable portion.

‘Well, it fascinated me, and I started asking several friends who worked with spreadsheets if it was possible to do with a spreadsheet some of the things a virus would do.

‘Based on what I’d been able to learn about viruses and what I’d been told, supposing I wanted to write a macro virus, what elements would I have to have in the program? The first was to be able to copy itself from a template to another template. I had that coded by one person. Another element was, if I wanted to manipulate data, then I wanted to simplify it. Someone else wrote that.

‘In manipulating the data, we did not want a large number of changes in any given cell. The author did just that, keeping the figure change to under plus or minus three percent. He also made that modification take place only once a day, no matter how often the template was called.

‘It was a ‘good’ virus which did not re-infect any template; that would give it away! After ten modifications to data, the program destroyed itself so there was no trace left in the template – well, almost no trace; there was one blank line where the original code was written.

‘When I got all the components, I tried putting them together, but they did not work too well. I then got another person whom I knew to answer my questions about how to overcome the faulty interface of the components without ever telling him what I was really doing.

‘Anyhow, around that time, I was going to speak to the *ADPA Auditor’s Association*, and I took the thing along, and just let it loose. It was obvious that the numbers had changed: the original values were all 00, and the changed values appeared with cents, not 00, so they stood out. After the first two demos and one other at a security meeting, I never took it out or demo’d it again.’

Certifying the Certifiers

A recent innovation in the anti-virus world is the certification just set up by the *NCSA* (*National Computer Security Association*). Highland believes strongly that it is extremely difficult to certify a product as good simply on the basis of its detection ratings: ‘Certification doesn’t come from this,’ he stated. ‘People sitting in a lab are never going to certify a thing. That it passes the virus-stopping tests – that’s the least important thing I consider when I look for a client.

‘The first thing I would look for is a product that can be installed without user intervention, so it won’t cost a fortune to install, especially in a large commercial environment.

‘Number two is the test against viruses. With over 6000 viruses out there, is it really necessary for a product to detect 100% to be certified? Some of those viruses have never been and will never be in the wild, and some won’t infect a flea.

‘Corporate users, even individual users, should have some assurance in the certification process: if a vendor promises to remove a virus from an infected program, the virus will be gone. Also, the program must still work. The vendor should not guarantee removal unless the program can work after being cleaned: if he is uncertain, he should say so!

‘Maybe there should be a stiff penalty related to this; not to allow the vendor to say: “It must be something in your system”, when they know there not much more than a snowball’s chance in Hell of having the program work after the virus is cleaned out.’

Outlook on Life

Highland believes that it is futile, if indeed not impossible, to look more than a year or two into the future as far as viruses are concerned.

‘There is very little cooperation in the industry as a whole – the only real cooperation I see is in virus exchange. However, the anti-virus industry will be held loosely together, because all the vendors want to collect as many viruses as possible. This collection is the lifeblood for their scanners.’

He foresees little more than marginal changes to anti-virus products in general, with vendors still pushing scanners, each trying to get the fastest to ‘win’ first place.

‘Several will attempt to monitor “normal” operations,’ he asserted, ‘whatever they are, and I believe customers will still have to live with false alerts. They’ll do this under the name of heuristic monitoring or some equally good marketing term. How about “cognitive analysis”?’

Some vendors, he says, will realize that they can produce an anti-virus product for a network which can be installed on all machines directly from the server, and without the need for human intervention: ‘The ability to get that information can be built into the software. Unfortunately, too many vendors make cosmetic changes to sell updates rather than valuable overhauls.’

Macro-type viruses he sees becoming an even greater threat than all the other viruses known today: ‘These viruses will *not* be confined only to *Microsoft* products,’ Highland stated, ‘but it was appropriate that the first macro virus in the real world was brought to us through the courtesy of the wonderful people who brought us *Windows 95*.’

More laboratory viruses will, he thinks, escape into the wild: this assertion is based on the numbers of people who already have access to such files, and on the fact that effective controls on distribution do not seem to be in place.

Highland has of late taken more of a ‘back seat’ than an active role in the computer virus world; he finds it interesting simply to sit back and watch what is happening, and to evaluate events.

Whether he will resume an active role remains to be seen; however, what *is* certain is that his contributions, both in academia and in computing as a whole, have left influences which will be felt for many years to come.

VIRUS ANALYSIS 1

FITW – Polymorphic down to the Boot

Dmitry O. Gryaznov
S&S International PLC

Recently, reports have been received of a new virus in the wild in Slovenia. It has been given the name FITW, short for ‘Fart in the wind’ – a text string found within the virus. FITW is what is sometimes called a ‘SPAM’ virus – that is to say, **S**tealth, **P**olymorphic **A**nd **M**ulti-partite. On top of this, it is a fast infector.

Infected File Execution

When an infected program is run on a clean PC, it checks to see if the virus is already resident. If the ‘Are you there?’ call (Int 21h, AX=FE23h) gets the reply AX=000Dh, the infected program assumes that FITW is resident. If this is not the case, the virus infects the hard disk’s Master Boot Record (MBR) and stays resident. It allocates memory by decreasing the size of the final Memory Control Block (MCB) in the DOS MCB chain – a technique often referred to as TWIXT, as it usually results in a virus residing between the end of the MCB chain and the BIOS top of memory.

Unlike most viruses which use this technique, FITW can detect the presence of DOS Upper Memory Block (UMB) MCBs. If it determines that these are present, the virus will put itself in memory allocated from the last DOS UMB. This results in the virus residing above the conventional 640K boundary, which could mean that more primitive scanners may have trouble detecting it in memory.

The virus then intercepts Int 21h (DOS Services), using a technique known as tunnelling: it intercepts Int 01h (Single Step), and traces Int 21h until it reaches DOS code. Thus, it can bypass the more primitive TSR behaviour blockers.

The virus intercepts several Int 21h functions: 36h (Get Disk Info), 4Ch (Program Exit), 4B00h (Load and Execute), 11h and 12h (FindFirst/FindNext using FCB), 3Dh (Open File Handle), 3Eh (Close File Handle) and FE23h (normally unused; utilised by FITW for self-recognition in memory).

Int 21h functions 11h, 12h and 36h are used for stealth: the virus tries to conceal visible changes to infected file length as reported by the DIR command (which uses File Control Block [FCB] calls). FITW also attempts to conceal the decrease in the amount of free space on the disk. The other intercepted DOS functions are used for replication.

Whenever a COM or EXE file is executed or opened, FITW will infect it unless its file name is COMMAND.COM, TB*.*, F-*.*, IV*.*, or contains the letter ‘V’. It is the fact that it infects on File Open that makes FITW a fast infector. When a file is infected, the variably-encrypted virus body is appended to the end of the file and the entry point is then adjusted appropriately. The size of the infected program thus increases by a variable amount, in the approximate range 7950 to 7990 bytes.

FITW determines that a file is already infected by checking the seconds field of the time stamp – if it is set to 34 seconds, it treats the file as already infected.

The virus also intercepts Int 13h (BIOS Disk Services) and infects hard disks and diskettes as they are accessed. Thus, it propagates both via infected files and infected floppies. Calls to read infected boot sectors are also stealthed by the virus: that is, whilst the virus is active in memory, none of the changes it has made to the system are visible, either in the infected files or in the infected MBR or boot sector.

In spite of its attempts, the stealth capabilities of this virus as regards infected files are far from perfect: the DIR command does reveal some changes to the length of infected files, although the visible changes are very different from the actual increase in file size.

Code Storage

On an infected disk, be it a hard disk or a diskette, the virus stores most of its code at the end of the physical disk. This is where it also keeps a copy of the original (uninfected) boot sector – in total, this remaining virus code occupies nine consecutive sectors.

On floppy disks, there is an extra twist – the virus actually uses nine sectors beyond the area normally formatted. On 40-track diskettes, such as 5.25-inch 360Kb, it adds a 42nd track (called track 41, since track numbers start from 0). On 80-track diskettes (5.25-inch 1.2Mb, 3.5-inch 720Kb and 1.44Mb), an 82nd track is added (track 81).

This technique has advantages and disadvantages – on the plus side, it makes overwriting of the virus body almost impossible; however, it makes infection of floppy disks more obvious due to the painful noise the drive makes when accessing extra tracks. It works on most drives used today: even some commercial copy-protection packages use it.

Gaining Control

When booting a computer from an infected floppy or from an infected hard disk, FITW receives control, loads the rest of its body to memory and passes control to its own code. The virus then allocates 5 Kilobytes of conventional memory through the standard method of decreasing the BIOS Top Of Memory value (stored in the word at offset 0413h in segment 0) by 5, and copies itself there.

Before passing control to the original boot sector/MBR code, the virus intercepts Int 13h (BIOS Disk Services) and then waits until the Int 21h (DOS Services) vector has changed. In this way it is able to detect that DOS has loaded, and then intercepts Int 21h itself to infect files.

Trigger and Payload

On Mondays, provided the date is the first, third, fifth, seventh, or ninth of the month, FITW will trigger when an infected program is run with the virus already active in memory. It then intercepts Int 09h (the Hardware Keyboard interrupt), making a reboot via Ctrl-Alt-Del impossible, and trashes all data on the local hard disks by writing random garbage across all the disks.

While this is happening, the virus responds to every keystroke by a user desperately trying to stop the trashing with the next character in a particular sequence. That is, whatever key the user presses, the virus displays the next letter of its internal message:

```
The “FartStorm” coded by Demon Emperor >Big
hello to CmosKiller&Xorboot< Are you wild? Hey
don’t... wait... let us talk... No? Be off
lame fuckhead!
```

Polymorphism

FITW is highly polymorphic, generating a new random encryption/decryption routine on every infection. Here we come to the most special feature of the virus: FITW is polymorphic not only in files, but in the MBR and boot sectors as well.

This is not the first virus which features polymorphism in infected boot sectors or MBRs, but all those previous to this used oligomorphic rather than polymorphic code in the boot sector/MBR. That is, they could be detected with a set of scan strings, possibly using wildcards. Unlike them, FITW enjoys true polymorphism wherever it is found.

Most of the code of an infected Master Boot Record is encrypted, with the decryptor being very much variable from infection to infection. In boot sectors, however, the code remains unencrypted. A handful of meaningful instructions (which simply load the body of the virus into memory) are randomly interspersed with randomly chosen ‘do-nothing’ instructions. This constitutes an instance of a relatively rare type of polymorphism – that in which the code is not encrypted.

In any case, no simple scan strings are available. Therefore, scanners which employ simple techniques of searching for scan strings to detect boot sector/MBR viruses might face difficulties detecting FITW. As this virus is both stealthy and dangerous, it is important for manufacturers to be able to detect it. The time has come to apply the full power of well-developed modern file scanning techniques (for example, generic decryption) to boot sector viruses as well as to file infectors.

Conclusions

FITW’s polymorphism in particular will present difficulties for reviewers and testers who want to compare different anti-virus software detection of boot and MBR infectors. Such viruses are responsible for at least 80 per cent of all the real life computer virus incidents.

It is no longer enough to have one or two sample diskettes or disk images of a particular boot sector virus. In cases such as that of FITW, it becomes necessary to have hundreds, if not thousands, of samples of infected disks, very much as it is with polymorphic file infectors – and we definitely shall see more polymorphic boot/MBR infectors in the future.

Bearing in mind that testing scanners against dozens of floppies infected with different ‘normal’, non-polymorphic, boot sector viruses is already a very tiresome and time-consuming process, I am extremely glad that I am not a reviewer! *[A rare privilege, that… Ed.]*

FITW

Aliases: In_the_Wind.

Type: Highly polymorphic (both in files and in boot sectors), multi-partite fast infector with stealth capabilities. Most of the code is encrypted in the infected MBR, but not in floppy disk boot sectors.

Infection: COM and EXE files, MBR of hard drive, boot sector of floppy disks.

Self-recognition in Memory:

Int 21h, AX=FE23h. The interrupt handler returns AX=000Dh.

Self-recognition in Files:

The seconds field of the time stamp is set to 34 seconds.

Hex Pattern in Memory:

```
9C3D 23FE 74F3 80FC 3675 03E9
6AFF 80FC 4C74 2C3D 004B 7503
```

(No simple pattern possible in files or boot sectors.)

Intercepts: Int 21h, functions 11h, 12h and 36h for stealthing purposes; functions 4Ch, 4B00h, 3Dh and 3Eh for virus replication; function FE23h for self-recognition in memory.

Trigger: On Mondays (third, fifth, seventh, ninth of month), it trashes all data on the hard disks, overwriting with random garbage.

Removal: Files – under clean system conditions, identify and replace. Hard disk – use FDISK /MBR. Diskettes – use SYS command, or reformat diskette.

VIRUS ANALYSIS 2

Unashamed and Naked

Kevin Powis
Precise Publishing Ltd

Unashamed is an in-the-wild boot sector virus which infects floppy and fixed disks. It contains a pseudo-random trigger, designed to release a text message payload.

Installation

Like all boot sector viruses, Unashamed is first executed when a computer boots from an infected disk. This causes the firmware to load the boot sector of the disk (the very first sector) into memory and pass control to it.

Once loaded and invoked in this way, Unashamed immediately relieves the PC of 1KB from the top of conventional memory, using the standard method of amending the low memory word which controls the number of kilobytes DOS thinks the PC has. Decrementing this word will thus convert a 640K PC into a 639K PC, leaving a nice hole at the top of memory for the virus.

However, there is a twist: the virus author seems to have designed the method he uses to fool some anti-virus software. Rather than simply using the normal address of the memory word, which could be recognised as potentially hostile, the author takes advantage of the 16-bit capacity of the 80x86 chip’s registers and loads an over-large value into one of these. This causes the register to overflow, and it is invisibly and automatically converted to the correct value by the CPU without intervention from the programmer.

Whilst this technique does nothing to prevent the virus being detected as a ‘known’ virus, it would certainly help the virus to escape heuristic tests conducted by scanners which do not specifically know about Unashamed.

Interrupt Hooking

With its 1KB safely reserved, Unashamed copies itself to the top of memory and continues execution from there. Next, it takes control of the required interrupt vectors.

In what appears to be another attempt to avoid certain anti-virus products (in this case, behaviour blockers), Unashamed uses a small sub-routine to handle interrupt hooking. This routine relies on certain registers having preset values on entry – the values set determine the interrupt to be hooked.

The end result is that interrupts are hooked as normal but, because the code doing this is effectively split in two, there is nothing immediately obvious as suspicious code on view. The author appears to have given this a lot of thought and to have considered that the extra work involved in doing this would give his creation a head start against the installed base of anti-virus software.

Unashamed hooks two interrupts, the BIOS disk services interrupt (Int 13h) and the keyboard interrupt (Int 09h). The former is used to allow the virus to replicate, and the latter, to control the trigger and the payload. The original vectors are stored by the virus for later use.

Once its interrupt handlers are in place, Unashamed needs to allow the original boot sector to run so that the computer can boot. This is achieved simply by sending an Int 19h, a seldom-used interrupt which will reboot the system without clearing memory or resetting interrupt vectors, allowing Unashamed to stay in memory and active.

The Disk Handler

The PC now reboots, this time with the virus already in memory, monitoring all disk and keyboard activity. Remember that, if this is the initial infection (if the PC has just booted from an infected floppy disk), the hard disk is currently still uninfected. However, that state of play only lasts for a moment, as the boot process will cause disk activity which is now intercepted by the virus.

> *“Unashamed resets the counter prior to infection, which voids the action of incrementing it after the infection has taken place”*

Every disk access (floppy or fixed) causes at least one Int 13h to occur, which is now intercepted by the virus. Unashamed allows all disk activity as normal, with the exception of attempts to read from or write to the Master Boot Sector (MBS) of any disk. This interception is needed to enable future infection, but it also allows the virus to provide itself with stealth capabilities.

If a program attempts to write to the boot sector – which would obviously destroy the virus – Unashamed simply stops the write from taking place and returns a success code; essentially ‘fooling’ the calling process into thinking that the write has worked correctly.

When Unashamed intercepts a read request destined for a boot sector, it uses the far pointer to the original disk handler which it obtained earlier to bypass its own stealth functionality and to read the target sector into memory. It then checks this sector to see if it is already infected. This is deemed to be the case if the bytes 051Ah are found at offset 1BBh inside the sector.

If the sector is infected, Unashamed locates the disk's original boot sector, which is hidden (see below), and returns the contents of this sector instead of those of the infected sector. This will make the PC appear uninfected when standard disk reads are used, and completes the virus' stealth capabilities.

However, if the disk is not infected, Unashamed uses a sub-routine to calculate the best place to hide the original boot sector before infection takes place. This routine determines first whether the target disk is fixed or floppy.

In the case of a fixed disk, sector 14, head 0, cylinder 0 is always used. This will usually fall on an unused sector before the first partition, but this is not guaranteed. The virus takes no steps to ensure that this is the case, so accidental damage could occur.

If the target is a floppy, Unashamed picks up a value from the boot sector which allows it to identify the disk type. From this it selects the sector in which it will hide the original boot sector as follows: for double density 3.5-inch diskettes (720KB) it chooses sector 5, and for double density 5.25-inch diskettes (360KB) it chooses sector 2. For all high density diskette types (5.25-inch 1.2MByte or 3.5-inch 1.44MByte), sector 14 is selected.

Once the hiding place has been decided, Unashamed writes the original clean boot sector to the chosen sector on the disk. It then resets an internal infection counter (see below) and updates its own image in memory with details from the target disk's BIOS Parameter Block (this details the disks structure). It now has an image that can be used to overwrite the target boot sector and thus infect the disk.

Unashamed completes the infection process by writing the boot sector to the disk and incrementing the same infection counter mentioned above. This counter is at offset 01BDh within the virus body, and is used by the other virus interrupt handler.

Trigger and Payload

The virus' keyboard interrupt handler controls the trigger and payload. Every time a key is pressed or released, the virus handler will take control. Immediately, it compares the infection counter to see if it is greater than one. If not, control is then passed down through the interrupt chain to the next keyboard handler.

Should the infection counter reach two or more, Unashamed relies on a pseudo-random trigger based on the PC timer. This gives a 50% chance, one hour in each 10-day period, of the payload being released. If this happens, the virus resets the video display to 40-column mode and unencrypts and displays an embedded message which reads: 'the UNashamed Naked!'. After a short delay, the PC reboots.

The release of the payload is dependent on the infection counter reaching two or more. However, in the copy of the virus used for this disassembly, this would never happen. As has already been noted above, Unashamed resets the counter prior to infection, which voids the action of incrementing it after infection has taken place. This cannot be a programming error: it is surely a conscious decision on the part of the author to give his creation a greater chance of spreading further without detection.

Conclusion

Overall, Unashamed employs interesting variations on standard techniques in an attempt to avoid generic detection. It is in the wild, but is easily detected by signature and not difficult to remove. The virus carries no in-built intent to cause damage.

[*Editor's Note: The first sample of this virus was received at Virus Bulletin from a correspondent in Rwanda (Central Africa) approximately one year ago. Since then, more reports of this virus have come in from this part of the world. It has now spread to both Europe and North America, but still appears to be particularly prevalent in Central Africa.*]

Unashamed

Aliases:	Unashamed_Naked.
Type:	Boot sector infector.
Infection:	Boot sector on floppy disks, MBS on hard disks.
Self recognition:	
	The bytes at offset 01BBh in any boot sector are equal to 051Ah.
Hex Pattern:	

	`D3E0 B900 018E C0FC ADAB E2FC`
	`B809 00BE 8101 BF4C 01E8 2001`

	(This pattern will locate the virus on hard and floppy disks, and in memory.)
Intercepts:	Int 13h (BIOS Disk handler) and Int 9h (Keyboard handler).
Trigger*:	Infection counter and random timer algorithm.
Payload*:	Displays the message 'the UNashamed Naked!' in 40-column mode.
Removal:	The standard method of running FDISK /MBR is sufficient to remove the virus from a hard disk. Removal from floppies can be achieved by salvaging any required files and then reformatting the floppy.

*Although the virus does contain the trigger and payload as described, these are effectively disabled in the copy of the virus analysed.

COMPARATIVE REVIEW

Jingle bells, jingle bells

Merry Christmas, said Santa, as he deposited a sackful of 25 anti-virus products on the testing desk in the *VB* office. Yes, it is that time of year when our thoughts turn to trees, turkey, and detection ratios. Once again, we at *VB* risk life, limb, and sanity by pitting the best the scanner manufacturers have to offer against our test-set.

The main chunk of tests is over four test-sets – In the Wild, Standard, Polymorphic, and Boot Sector. The In the Wild test-set uses Joe Wells' *WildList* – the set is made up of file viruses on that list which have been reported by two or more anti-virus specialists. It contains 286 samples of 97 viruses.

The Polymorphic set has been revamped to update it and make it easier to handle. It now contains 500 samples each of 11 viruses. With the Standard test-set, this makes a total of 6051 samples of 270 file viruses, occupying 78,060,778 bytes. The Boot-Sector set has been expanded from 15 samples to 68: all samples are held on 3.5-inch diskettes, and SIMBOOT [*see 'Revisiting the DOS Scanner Testing Protocol', VB, November 1995 p.14*] is used where possible to test against this set.

Scan Times

All scan times are quoted for products running on the same machine – a *Compaq Deskpro 386/20e* with 4MB memory and a 540MB *Conner Peripherals* hard drive (*Norton* info: 936.6KB/s data transfer rate; 12.5 ms average seek; 2.71 ms track-to-track seek). Clean floppy scan timings use a 1.44MB, 3.5-inch diskette with 1,408,310 bytes in 50 executable files (20 COM, 30 EXE). Infected floppy timings use a similar diskette with 100 EXE and 50 COM files infected with Groove or Coffeeshop, total 1,413,319 bytes. Scan times for a clean hard drive were produced by running the products across one partition (1253 files in 42 directories; 82,270,423 bytes) on the test machine's hard drive.

Each product was run, as far as possible, in its standard configuration, with minor changes. Options to prevent user interaction and to write a log file were used to allow most products to be run unattended in a batch file.

Percentages

The scoring system is fully described in the Protocol article mentioned above, and in a previous one [*VB, February 1995 p.12*]. The only significant change was in the Standard and In the Wild test-sets, now scored on a per virus, not per sample, basis. If a product identified three out of four samples of virus A, two of two of virus B, and three of five of C, it would score 100*((3/4)+(2/2)+(3/5))/3 = 78.3%: previously scoring was 100*((3+2+3)/(4+2+5)) = 72.7%.

The polymorphic set is weighted in the same way as in previous comparatives [*for a description of the methods used, see VB July 1995 p.14*].

The percentage for the boot sector test-set is still calculated as a simple percentage: 100*((number detected)/(number of samples in set)). This is because there is only one sample of each virus in this set [*something which may change in the future. Ed.*]

Extra Tests

In addition to the tests carried out in previous comparatives, *VB* has introduced testing of memory detection, and boot sector and file disinfection (where available). In this review, the following extra tests were carried out:

- Disinfection of AntiCMOS.A, Empire.Monkey.B and Form on 1.44MByte floppy and hard disk, and detection in memory.
- Disinfection of Cascade.1701.A, Green.Caterpillar.1575, Natas.4774, Nomenklatura, and SatanBug.5000.A.

AVAST! 7.50 (23/10/95)

In the Wild	99.0%	Boot Sector	92.6%
Standard	100%	Polymorphic	88.6%
Overall	95.1%		

This product's detection remains excellent, especially in the In the Wild and Standard test-sets. Polymorphic rates have dropped slightly: this time it missed two samples of Satanbug.5000.A, and all replicants of DSCE.Demo.

AVAST! correctly detected all the boot viruses in memory, and was able to remove such viruses from diskettes by using a standard boot sector. To remove them from the hard drive requires the user to have previously created a recovery disk. This product has no file disinfection capabilities.

AVP 2.2 (24/10/95)

In the Wild	100%	Boot Sector	97.1%
Standard	100%	Polymorphic	100%
Overall	99.3%		

Once again, *AVP* nearly cleans up – only two samples (Crazy_Nine and Intruder_Boot) stood between it and a perfect score. There are drawbacks: it suffered two false positives, and takes an age to scan. Only *DrWeb* was slower on the clean hard disk: taking almost 51 minutes to scan just over 80 MBytes of files is not what most people want.

AVP's primary use is as a tool for helpdesk personnel, or those who clear up virus attacks. Where an infection is suspected (perhaps because another product says so), *AVP* is exactly the sort of product required – in these circumstances, speed (or lack of same) is not so much of a drawback. Its demonstration capabilities are also noteworthy – *AVP* can reproduce the visual and audio effects of many viruses.

All the boot viruses were found and cured in memory, and correctly removed from hard and floppy disks. There was one oddity – when presented with a Form-infected hard disk, it found the virus in the MBR (which it then, unsurprisingly, failed to disinfect) before finding it in and removing it from the PBR. Curious… File disinfection was equally trouble-free – all disinfected files were identical to the uninfected originals, apart from the Green.Caterpillar samples, both of which had sixteen unused bytes of the virus left behind at the end of the file.

AVScan 2.49a (24/10/95)

In the Wild	89.7%	Boot Sector	88.2%
Standard	98.1%	Polymorphic	88.6%
Overall	91.2%		

Overall, a slight drop in *AVScan*'s scores since the July comparative; however, they are very much above average.

Boot sector viruses in memory were detected with no problems, but the product requires a separate program for disinfection of boot and file viruses, which was not supplied.

CPAV 2.2 (09/09/95)

In the Wild	81.8%	Boot Sector	82.4%
Standard	92.0%	Polymorphic	34.1%
Overall	72.6%		

This comparative is notable for one thing perhaps above all others – it is the first time since June 1992 that *CPAV* has survived without crashing. Regrettably, this appears to be due more to the new ways in which *VB* runs scanners across the collection (they encounter fewer viruses in the course of one run) than to any change on the part of the scanner.

Whatever the reason, for the first time in over three years we get a look at *CPAV's* polymorphic scores. These scores are symptomatic of a product which seems to have undergone no primary development for many months. Signature updates help, but all scanners need modifications to their engine from time to time to enable them to deal with new techniques. It is clear these changes have been lacking.

CPAV detected and cured the boot sector viruses in memory and on hard and floppy disks. File viruses were less of a success – it failed to detect both Natas samples, and could not repair SatanBug. However, the files containing Cascade and Nomenklatura came out identical to the uninfected originals, and the Green.Caterpillars merely had some unused bytes of the virus left at the end of the file.

Doctor Lite 95.10b

In the Wild	86.0%	Boot Sector	88.2%
Standard	92.0%	Polymorphic	39.1%
Overall	76.3%		

It is a shame that the Polymorphic detection rate of this package from *Thompson Network Software* is so low, because the scores in the other sets are respectable enough.

All the boot sector viruses were correctly found in memory, but the product submitted for testing does not offer removal of boot sector or file viruses, so this could not be tested.

Dr Solomon's AVTK 7.53

In the Wild	100%	Boot Sector	95.6%
Standard	99.7%	Polymorphic	90.7%
Overall	96.5%		

As in July, the *AVTK* is let down in the Polymorphic test-set by incomplete detection of groups. Despite missing only thirteen (of 5,500!) polymorphic samples, it drops just over 9%. In the other test-sets, performance is equally admirable, as expected from this highly-regarded scanner. Of particular note is the faultless performance on the In the Wild test-set.

No problems were encountered with detection of viruses in memory (a reboot is advised afterwards), or with boot sector virus removal from floppy or hard disks – all three viruses were correctly removed. Equally trouble free was file virus disinfection: all files were identical to their uninfected originals, apart from samples of Green.Caterpillar, both of which had 16 unused virus bytes left behind at the end.

DrWeb 3.06a (26/10/95)

In the Wild	95.9%	Boot Sector	66.2%
Standard	79.5%	Polymorphic	95.2%
Overall	84.2%		

A fairly impressive *VB* comparative debut for this Russian product. Like its compatriot, *AVP*, it suffers from speed problems – it was the slowest product tested, taking over one hour to complete the clean hard disk scan test. The heuristics which allow this product to function so well without specific knowledge of very many viruses (the product states it knows about 1390) also exact something of a penalty – the product suffered three false positives.

All boot sector viruses were detected and cured in memory, and disinfected from hard and floppy disks. The product suffered a lapse of reason when removing AntiCMOS.A from a diskette: a dialogue box appeared containing only ‘Yes’, ‘No’, and ‘Cancel’ buttons – no message. Blind optimism led this reviewer to choose ‘Yes’: fortunately the virus was removed correctly. *DrWeb* could not disinfect either Nomenklatura sample, and left 16 unused bytes of virus code behind when disinfecting the Green.Caterpillars. All other files were identical to uninfected originals.

F-Prot Professional 2.20a

In the Wild	98.7%	Boot Sector	95.6%
Standard	98.1%	Polymorphic	64.0%
Overall	89.1%		

Time and tide wait for no man, the cliché says – they do not wait for polymorphic scanning engines either. A bitty performance against that set dragged the final score down, but a little work here will see the product back at the top once more. Curiously, despite recently having been certified by the *NCSA* (a process during which a product has to detect every virus in the wild at the time of the test) it missed seven samples from the In the Wild test-set; not simply by not knowing about that particular virus, but, in all cases, by missing some of a group of samples whilst detecting others.

The other tests went well – the boot viruses were found in memory (the product advises a clean boot), and correctly removed from hard and floppy disks. File virus removal was also excellent – all files were disinfected correctly except for the Green.Caterpillar samples, where 16 bytes were left.

Intel LANDesk Virus Protect 3.0 r07

In the Wild	84.3%	Boot Sector	89.7%
Standard	91.7%	Polymorphic	51.0%
Overall	79.2%		

This product has the honour of being the first to arrive on CD-ROM for a *VB* comparative review – because it is designed for use in a networked environment, it comes with all sorts of software not tested in this review. The detection offered by the DOS scanner component is perhaps best described as mediocre, especially in the Polymorphic set, where getting just over half marks is not really good enough.

The boot viruses were all found in memory (after which a reboot is advised) and correctly disinfected, apart from Form, which could only be removed from the diskette. Of file viruses, the Green.Caterpillar and SatanBug samples could not be disinfected. Both files containing Natas samples were 30 bytes longer than the originals, and the EXE sample of Nomenklatura differed by one byte from the original: *LANDesk* had repaired the EXE header incorrectly, leaving the ‘number of pages’ field incremented by one, a change which should not affect functionality. Remaining samples were disinfected correctly.

IBM AntiVirus 2.3 (25/09/95)

In the Wild	99.5%	Boot Sector	95.6%
Standard	97.5%	Polymorphic	79.4%
Overall	93.0%		

Figures not astoundingly different from those in the last comparative. As in many other cases, it is the Polymorphic test-set which caused the most problems, but of note is the In the Wild test-set, where it missed only one sample.

The boot sector tests went well, with all viruses found in memory at the appropriate juncture, and a clean boot then advised. Viruses were disinfected correctly from both hard and floppy disks. The file virus tests were rather less successful – only Cascade and the two Green.Caterpillars could be disinfected, and 16 unused bytes of the latter two were left.

InocuLAN 4.0 (19/10/95)

In the Wild	96.9%	Boot Sector	92.6%
Standard	95.1%	Polymorphic	72.3%
Overall	89.2%		

Another quantum leap in polymorphic detection rate: just as the July result was far better than that last January, the result this time around is much improved over July. If this continues, the product will soon be very good indeed.

Also a good performance on other tests: the product detected and cured all viruses tested in memory, and disinfected hard and floppy disks. The only file viruses which, when disinfected, were identical to their uninfected originals were Cascade and Nomenklatura. Of the others, most were longer than, but functionally identical to, the originals, but the sample of SatanBug was left 80 bytes *shorter* than the original. This had no effect here, but it could well cause problems with other infected executables.

Iris AntiVirus Plus 21.09

In the Wild	96.9%	Boot Sector	94.1%
Standard	95.1%	Polymorphic	72.3%
Overall	89.6%		

Overall, good results for this Israeli product; however, like so many others, it is the Polymorphic test-set which lets *Iris* down. Having said that, 72.3% is far from terminal, and the product could, with a little work, become a real contender.

Product Name	In the Wild (286)		Boot Sector (68)		Standard (265)		Polymorphic (5500)		Overall
	Number	Percent	Number	Percent	Number	Percent	Number	Percent	Percent
AVAST!	283	99.0%	63	93.0%	265	100%	4998	88.6%	95.1%
AVP	286	100%	66	97.1%	265	100%	5500	100%	99.3%
AVScan	251	89.7%	60	88.2%	260	98.1%	4995	88.6%	91.2%
CPAV	234	81.8%	56	82.4%	246	92.0%	2003	34.1%	72.6%
Doctor Lite	250	86.0%	60	88.2%	246	92.0%	2532	39.1%	76.3%
Dr Solomon's AVTK	286	100%	65	95.6%	264	99.7%	5487	90.7%	96.5%
Dr Web	275	95.9%	45	66.2%	219	79.5%	5484	95.2%	84.2%
F-Prot Professional	279	98.7%	65	95.6%	260	98.1%	4194	64.0%	89.1%
IBM AntiVirus	285	99.5%	65	95.6%	259	97.5%	4826	79.4%	93.0%
InocuLAN	277	96.9%	63	92.6%	254	95.1%	4299	72.3%	89.2%
Intel LANDesk	241	84.3%	61	89.7%	248	91.7%	3405	51.0%	79.2%
Iris AntiVirus	277	96.9%	64	94.1%	254	95.1%	4299	72.3%	89.6%
McAfee Scan	278	97.1%	65	95.6%	256	96.3%	4426	76.3%	91.3%
MSAV	100	35.1%	11	16.2%	216	78.2%	466	6.4%	34.0%
Norton AntiVirus	273	96.6%	64	94.1%	256	96.3%	3499	61.2%	87.1%
Norman	279	97.4%	66	97.1%	261	98.8%	5473	95.1%	97.1%
PCVP Lite	190	68.0%	31	45.6%	249	93.0%	3126	51.7%	64.6%
ScanVakzin	186	68.0%	50	73.5%	234	86.7%	987	15.7%	61.0%
Sweep	286	100%	68	100%	263	99.4%	5000	90.9%	97.6%
ThunderBYTE	286	100%	59	86.8%	260	98.1%	4228	69.0%	88.5%
VET	264	94.3%	58	85.3%	258	96.9%	4194	64.0%	85.1%
Virus ALERT!	283	99.0%	63	92.6%	265	100%	4998	88.6%	95.1%
Virus Buster	198	73.3%	49	72.1%	235	88.0%	1990	34.0%	66.8%
ViruSafe	251	90.0%	65	95.6%	236	89.8%	2920	44.4%	79.9%
Vi-Spy	274	97.0%	63	92.6%	261	98.5%	4343	68.3 %	89.1%

Disinfection and in-memory detection, however, proved problematic. The product detected and cured Form in memory, but missed Empire.Monkey.B and AntiCMOS.A – a dangerous mistake.

As for disinfection, only AntiCMOS.A caused problems: the message ‘The boot virus, Anti Cmos, was found on the boot sector of drive a and has destroyed your file’ [*sic*] was displayed, and the user invited to select ‘Remove’ to clean the virus. When this was done, the product continued. Checking the floppy by hand revealed that the virus was still there. Returning to the product, a close look at the log information revealed by pressing the ‘More info’ button at the end of a scan informed me that no viruses had been cleaned. The user is not told (unless he looks really hard!) that disinfection failed, and thus he is likely to assume that it was successful.

The file virus tests went more smoothly. Cascade and the Nomenklaturas were disinfected correctly; and the four of Green.Caterpillar and Natas were left slightly longer but functionally identical. However, the sample of SatanBug, whilst also functionally identical, was 80 bytes shorter than the original file. This could produce problems on other files.

Product Name	Clean Floppy Scan (min:sec)	Clean Floppy Read (KB/sec)	Clean Hard Drive Scan (min:sec)	Clean Hard Drive Read (KB/sec)	Infected Floppy Scan (min:sec)	Infected Floppy Read (KB/sec)
AVAST!	2:12	10.4	4:03	333	5:38	4.1
AVP	8:02	2.9	51:00	26.4	11:40	2
AVScan	1:48	12.8	4:51	277.8	2:41	8.6
CPAV	2:38	8.7	12:48	105.2	4:28	5.2
Doctor Lite	3:15	7	22:42	59.3	8:10	2.8
Dr Solomon's AVTK	1:24	16.3	4:20	310.6	19:49	1.16
Dr Web	6:38	3.5	1:01:11	22	5:09	4.5
F-Prot Professional	1:44	13.2	7:06	189.4	2:56	7.8
IBM AntiVirus	2:45	8.3	18:12	74	3:35	6.4
InocuLAN	2:57	7.8	13:10	102.2	15:53	1.5
Intel LANDesk	1:38	14	6:28	208.2	6:30	3.54
Iris AntiVirus	3:08	7.3	15:06	89.2	21:56	1.1
McAfee Scan	1:43	13.4	9:02	149	6:24	3.6
MSAV	1:33	14.7	6:10	218.2	4:58	4.5
Norton AntiVirus	1:40	13.8	2:19	580.1	6:58	3.3
Norman	2:37	8.8	6:09	218.7	4:15	5.4
PCVP Lite	0:42	32.8	3:18	406.2	0:54	25.3
ScanVakzin	1:13	18.8	3:13	417.8	3:16	7.1
Sweep	1:26	16.1	7:25	181.6	1:48	12.8
ThunderBYTE	0:30	46	0:38	2125	2:54	7.9
VET	0:54	25.7	2:41	502.1	1:40	13.8
Virus ALERT!	2:34	9	18:06	74.4	5:43	4
Virus Buster	1:18	17.5	4:06	328.25	10:03	2.3
ViruSafe	1:29	15.5	3:53	346	3:33	6.5
Vi-Spy	1:16	18	4:32	296.7	3:25	6.7

Norman Virus Control 3.48 (15/10/95)

In the Wild	97.4%	Boot Sector	97.1%
Standard	98.8%	Polymorphic	95.1%
Overall	97.1%		

This product was absent from the July comparative, but looking back twelve months, we see that *NVC*'s score in the Polymorphic test-set has risen by a factor of only fractionally less than three. This brings it firmly into the big league, resulting in a very sound product in terms of detection.

In the other tests, things were not quite so good. Both AntiCMOS.A and Form were missed in memory, and I could not persuade *NVC* even to detect Empire.Monkey.B on the infected hard disk. Apart from that, the boot viruses were disinfected correctly.

Of the file viruses, SatanBug could not be repaired, but Cascade and the COM sample of Nomenklatura were repaired correctly. The Natas' and the Green.Caterpillars were left 30 and 16 bytes (respectively) longer than originally, and the checksum of the EXE Nomenklatura was damaged – this does not affect execution.

McAfee Scan 2.2.7

In the Wild	97.1%	Boot Sector	95.6%
Standard	96.3%	Polymorphic	76.3%
Overall	91.3%		

Perhaps history does repeat itself. *McAfee's Scan*, once tops at detection, is on the up after a spell in the doldrums. While not yet at the top, it is in real danger of making it there.

All the boot sector viruses tested were detected correctly in memory (the product suggests a clean boot), and correctly disinfected from hard and floppy disks. Of file viruses, the COM infections of Cascade, Natas, and Nomenklatura were disinfected correctly; both Green.Caterpillars were 16 bytes longer; the EXE Nomenklatura sample had the initial CS:IP value in the header corrupted; the EXE Natas sample had the minimum memory field in the EXE header corrupted (to a value larger than the original), and the SatanBug sample was 294 bytes smaller than the original. All files were functionally intact, but SatanBug may cause problems.

MSAV 6.22

In the Wild	35.1%	Boot Sector	16.2%
Standard	78.2%	Polymorphic	6.4%
Overall	34.0%		

MSAV serves its purpose in the *VB* comparative – boundary conditions are very important in such things, and it now forms a baseline below which no other product should fall.

On the extra tests, of boot sector viruses, only Form was dealt with correctly – it was detected and cured in memory and on disks. The other two were detected neither in memory nor on disk. File disinfection was little better – Natas and SatanBug were missed, and the COM samples of Green.Caterpillar had four bytes of program code corrupted: this will cause problems. The Nomenklaturas, Cascade and the other Green.Caterpillar were disinfected correctly.

Norton AntiVirus 3.0 (09/10/95)

In the Wild	96.6%	Boot Sector	94.1%
Standard	96.3%	Polymorphic	61.4%
Overall	87.1%		

Detection is on the whole very good, excepting Polymorphics. The problem is not so much *incomplete* detection as the lack of knowledge of certain viruses. It ought to know about Uruguay.4 and Nightfall (for example) by now, though…

All the boot sector viruses were detected in memory (a clean boot is advised), and correctly disinfected from disk. Of the file viruses, SatanBug could not be repaired, but both the Nomenklaturas and the Cascade were left identical to the original samples. The Green.Caterpillar and Natas samples were functionally identical after disinfection, but varying numbers of bytes longer than the originals.

PCVP Lite 2.26 (01/09/95)

In the Wild	68.0%	Boot Sector	45.6%
Standard	93.0%	Polymorphic	51.7%
Overall	64.6%		

PCVP shows an interesting trait in all recent *VB* comparative reviews – this product always has more trouble with the Boot Sector test-set than with anything else. This is strange, because on the whole boot sector viruses are the easiest type to detect. However, *PCVP* also had problems against the polymorphic test-set. The Standard test-set gives it its best result, and of all the sets used, this has the highest percentage of old viruses…

PCVP Lite does not offer virus removal, so only the memory detection tests are relevant – the product failed to find Empire.Monkey.B in memory, but found the other two.

ScanVakzin 4.221

In the Wild	68.0%	Boot Sector	73.5%
Standard	86.7%	Polymorphic	15.7%
Overall	61.0%		

ScanVakzin's overall percentage is virtually unchanged since July, and only minor fluctuations have occurred in the individual figures – things are still not good.

Curiously, *ScanVakzin* appears to be able to cure boot sector viruses in memory, but not remove them from either hard or floppy disks. There is no disinfection of file viruses.

Sweep 2.79

In the Wild	100%	Boot Sector	100%
Standard	99.4%	Polymorphic	90.9%
Overall	97.6%		

It seems to be *VB's* destiny to fool *Sophos* only twice in each comparative review – this time with DSCE.Demo and Cruncher. If there is a quibble, it is with the slightly inconsistent naming and imprecise identification of some viruses, but with detection like this it seems perhaps churlish.

All three boot viruses were found in memory, and a clean boot encouraged. Empire.Monkey.B and Form could be disinfected from hard and floppy disks, but AntiCMOS.A could not be removed. No file disinfection is offered.

ThunderBYTE 6.50

In the Wild	100%	Boot Sector	86.8%
Standard	98.1%	Polymorphic	69.0%
Overall	88.5%		

Perhaps it is the fact that a redesigned heuristic engine has been incorporated into *ThunderBYTE* that accounts for its polymorphic detection drop – it should be capable of better here. However, 100% on the In the Wild test-set; and the Standard test-set score – both are excellent. And *ESaSS* still has the fastest scanner in the West… and, indeed, the East.

All boot viruses were found in memory (instructions were given to power off and boot clean), and removed from hard and floppy disks. Disinfection of file viruses fares better if the product's ANTI-VIR.DAT files are available – these record entry point and checksum information for executables, and the product uses this to help it reconstruct files. Repair is possible without this, but less likely to be successful.

Only the EXE Green.Caterpillar sample could not be repaired. The EXE samples of Natas and Nomenklatura came out different from the originals – most of the Natas virus body (4600 bytes) was still present, but the EXE header had been patched to prevent it being called, and with Nomenklatura, the initial value of SS:IP was corrupted. In both these cases, the programs were functionally fine.

Virus ALERT! 4.10

In the Wild	92.6%	Boot Sector	99.0%
Standard	100%	Polymorphic	88.6%
Overall	95.1%		

A change in scanning engine for *Look Software* results in a change in the appearance of the percentage table – for the most part a change in the right direction. All the boot viruses were correctly found in memory and disinfected on floppy disks. The product is only capable of removing boot viruses from the hard disk if you have previously prepared a recovery disk – and if you have, it works. There is no facility to disinfect infected files.

Virus Buster 4.84.00

In the Wild	73.3%	Boot Sector	72.6%
Standard	88.0%	Polymorphic	34.0%
Overall	66.8%		

On average, *Leprechaun Software's* product's scores are fractionally down from last time, although the polymorphic detection rate has improved by a factor of two. All the boot viruses were successfully found in memory and removed from floppy disks. Only Empire.Monkey.B could not be removed from the hard disk. Of the file viruses, only the Green.Caterpillar (left 16 bytes longer than before) and Natas samples (31 bytes longer) were disinfected.

VET 8.353

In the Wild	94.3%	Boot Sector	85.3%
Standard	96.9%	Polymorphic	64.0%
Overall	85.1%		

In the Wild and Standard detection is slightly up on July, whilst Boot Sector and Polymorphic scores are down, as is overall percentage. The Polymorphic score would be vastly improved if there were more complete detection of the viruses the product knows – this is where it really loses out.

All boot viruses were detected in memory (Form and (Empire.Monkey.B were also cured therein), and correctly disinfected on floppy and hard disks. As to file viruses, neither sample of Natas was detected, and the SatanBug sample could not be repaired. Of the others, all were disinfected to programs functionally identical to uninfected originals, but varying numbers of bytes (1 to 721) longer.

ViruSafe 6.7

In the Wild	90.0%	Boot Sector	95.6%
Standard	89.8%	Polymorphic	44.4%
Overall	79.9%		

Very good Boot Sector and more than reasonable In the Wild and Standard detection are again offset by distinctly unexciting Polymorphic figures. The scores are higher than last time, but not dramatically so.

All boot sector viruses were detected in memory and disinfected from floppy and hard disks. *ViruSafe* uses anti-stealth techniques which allow it to remove a virus from the disk whilst it is active in memory: the product should make it clearer that the virus is still active in memory, and that a simple reboot will complete the process. With file viruses, all samples apart from the Nomenklaturas were disinfected; but only SatanBug was removed correctly. The others had varying numbers of bytes (1 to 30) left at the end of the file.

Vi-Spy 12.0 (01/10/95)

In the Wild	97.0%	Boot Sector	92.6%
Standard	98.5%	Polymorphic	68.3%
Overall	89.1%		

Vi-Spy, as far as I can tell, is the only product in the comparative that, in its default mode, still attaches significant weight to a file time/date stamp which is set to a peculiar value. It is by this method that it detects DSCE.Demo – a more intelligent algorithm would be nice. Detection in all categories is up from six months ago, however.

Vi-Spy detected all boot sector viruses in memory, and Empire.Monkey.B and Form were disinfected from floppy and hard disks without problems. AntiCMOS.A was removed from the floppy, but I was advised to use a utility program, VSRECOVR, to repair the infected hard disk – it did the job. File viruses were dealt with well – Cascade, both Nomenklaturas, and SatanBug emerged identical to the uninfected files. The two Green.Caterpillars and the Natas COM sample were a variety of bytes longer but functionally identical – only the EXE sample of Natas was not repaired.

Conclusion

In terms of detection, there are no big surprises. *AVAST!* and *Virus ALERT!* tie for fifth place on 95.1%, *Dr Solomon's AVTK* is fourth with 96.5%, *NVC* next on 97.1%, and *Sweep* just above that with 97.6%. Out in front is *AVP*, with 99.3%. It is nice to see *Norman Virus Control* in the top five – I look forward to seeing more of this product in the future.

Lack of space prohibits, alas, analysis of the bulk of the tests. However, the relevant statistics for scan time tests are displayed in the table on page 17.

For the extra tests, one of the main difficulties is ranking the products, which has not been done this time, as evaluating a percentage for this type of test is non-trivial. For information on how a particular product performed in this area, refer to the text. It is gratifying to see that the important area of detection in memory was handled well by most products. In future comparatives, it is hoped that there will be space for more detailed analysis of the results.

TEST-SETS

In the Wild

286 genuine infections (one each, except where stated) of:
814 (3), Accept.3773 (5), Anticad.4096 (4), Anticad.4096.Mozart(4), Arianna.3375 (4), Avispa.D (2), Bad_Sectors.3428 (5), Barrotes.1310.A (2), BootEXE.451, Bosnia:TPE.1_4 (5), Byway.A, Byway.B, Cascade.1701.A (3), Cascade.1704.A (3), Cascade.1704.D (3), Cawber (3), Changsa.A (6), Chaos.1241 (6), Chill, Coffeeshop (2), CPW.1527 (4), Dark_Avenger.1800.A (3), Datalock.920.A (3), DelWin.1759 (3), Die_Hard (2), Dir-II.A, DR&ET.1710 (3), Fairz (6), Fichv.2.1, Finnish.357 (2), Flip.2153 (2), Flip.2343 (6), Freddy_Krueger (3), Frodo.Frodo.A (4), Ginger.2774 (2), GoldBug (4), Green_Caterpillar.1575.A (3), Helloween.1376 (6), Hidenowt, Jerusalem.1244 (6), Jerusalem.1808.Standard (2), Jerusalem.Sunday.A (2), Jerusalem.Zero_Time.Australian.A (3), Jos.100 (3), Junkie, Kaos4 (6), Keypress.1232.A (2), Lemming (2), Liberty.2857.A (2), Little_Brother.307, Little_Red (2), Macgyver.2803.B, Maltese_Amoeba (3), Markt.1533 (3), Mirea.1788 (2), Natas.4744 (5), Necros (2), Neuroquila, No_Frills.Dudley (2), No_Frills.No_Frills.843 (2), Nomenklatura (6), November_17th.768.A (2), November_17th.800.A (2), November_17th.855.A (2), Npox.963.A (2), One_Half.3544 (5), Ontario.1024 (3), Pathogen:SMEG (5), Phx.965 (3), Predator.2448 (2), Quicky, Sarampo (6), SatanBug.5000.A (2), Sayha (2), Screaming_Fist.II.696 (2), Sleep_Walker (3), Stardot.789.A (6), SVC.3103.A (2), Tai-Pan.438.A (3), Tai-Pan.666 (2), Tequila.A, Three_Tunes.1784 (6), Trakia.653, Tremor.A (6), Trojector.1463 (6), Vacsina.TP-05.A (2), Vacsina.TP-16.A, Vampiro, Vienna.648.Reboot.A, Vienna.Bua (3), Vinchuca (3), Virogen.Pinworm (6), VLamix, Xeram (3), Yankee_Doodle.TP.39 (5), Yankee_Doodle.TP.44.A, Yankee_Doodle.XPEH.4928 (2)

Boot Sector

68 genuine infections (one each) of:
AntiCMOS.A, AntiCMOS.B, AntiEXE, Are Three, Boot.437, BootEXE.451, Brasil, Bravo_Boot.B, Crazy Nine, Da_Boys.A, DiskWasher.A, DiskWasher.B, Empire.Int_10.B, Empire.Monkey.A, Empire.Monkey.B, EXEBug.A, Finnish_Sprayer, Flame, Form.A, Form.C, Form.D, Frankenstein, Ibex, IntAA, Intruder_Boot, Joshi.A, Jumper.A, Jumper.B, Junkie, Kampana.A, Leandro, Matteo, Natas.4744, NYB, Parity_Boot.A, Parity_Boot.B, Peanut, Peter, QRry, Quox, Rainbow, Ripper, RM.B, Russian_Flag.A, Sampo, She_Has, Stealth_Boot.B, Stealth_Boot.C, Stoned.16.A, Stoned.8.A, Stoned.Angelina, Stoned.Azusa.A, Stoned.Bunny.A, Stoned.Dinamo, Stoned.June_4th.A, Stoned.Kiev, Stoned.LZR, Stoned.No_Int.A, Stoned.NOP, Stoned.Standard, Stoned.Swedish.Disaster, Stoned.W-Boot.A, Swiss_Boot, Unashamed, Urkel, V-Sign, WelcomB, Wxyc.A

Polymorphic

5500 genuine infections (500 each), of:
DSCE.Demo, Girafe:TPE, Groove and Coffee_Shop, MTZ.4510, Neuroquila.A, Nightfall.4559.B, One_Half.3544, Pathogen:SMEG, SatanBug.5000.A, SMEG_v0.3, Uruguay.4

Standard

265 genuine infections (one each, except where stated) of:
405, 417, 492, 516, 600, 696, 707, 777, 800, 905, 948, 1049, 1260, 1600, 2100 (2), 2144 (2), 5120, 8888, 8_Tunes, AIDS, AIDS-II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax, Anti-Pascal (5), Argyle, Armagedon, Athens (2), Attention, Bebe, Big_Bang, Black_Monday (2), Blood, Burger (3), Butterfly.Butterfly, Captain_Trips (4), Casper, 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, Destructor, Diamond.1024.B, Dir, DiskJeb, DOS_Hunter, Dot_Killer, Durban, Eddie, Eddie_2.A (3), Fax_Free.Topo, Fellowship, Fish_1100, Fish_6 (2), Flash, Fu_Manchu (2), Genesis.226, Halley, Hallochen.A (3), HLLC.Even_Beeper.A, Hymn (2), Icelandic (3), Internal, Invisible_Man (2), Itavir, Jerusalem.PcVrsDs (4), Jo-Jo, Jocker, July_13th, Kamikaze, Kemerovo, Kennedy, Lamer's_Surprise, Lehigh, Liberty (5), Liberty.2857.D (2), Loren (2), LoveChild, Lozinsky, Macho (2), MIX1 (2), MLTI, Monxla, Murphy (2), Necropolis, Nina, Nothing, NukeHard, Number_of_the_Beast (5), Old_Yankee (2), Oropax, Parity, Peanut, Perfume, Phantom1 (2), Pitch, Piter (2), Poison, Polish-217, Power_Pump.1, Pretoria, Prudents, Rat, Revenge, Riihi, SBC, Screaming_Fist.927 (4), Semtex.1000, Shake, Sibel_Sheep (2), Spanz (2), Stardot.789.D (2), Starship (2), Subliminal, Sunday (2), Suomi, Suriv_1.01, Suriv_2.01, SVC.1689.A (2), Sverdlov (2), Svir, Sylvia, Syslock, Syslock.Macho (2), Syslock.Syslock.A, Taiwan (2), Telecom (4), Terror, Tiny (12), Todor (2), Traceback (2), TUQ, Turbo_488, Typo, V-1, V2P6, Vacsina.634, Vacsina.Penza.700 (2), Vacsina.TP.? (6), Vcomm (2), VFSI, Victor, Vienna.? (11), Virdem, Virdem.1336.English, Virus-101 (2), Virus-90, Voronezh.1600.A (2), VP, Warrier, Warrior, Whale, Willow, WinVir_14, Yankee_Doodle.TP.? (5), Zero_Bug.

PRODUCT REVIEW

Watchdog

Dr Keith Jackson

Watchdog is a multi-user security system which can be used to provide features such as user partitioning, subdirectory protection and data encryption on a PC. The bumph which accompanied *Watchdog* mentions special purpose hardware which can augment the software, but this review refers only to the software version of *Watchdog*.

Documentation

If security products were judged by weight or volume then *Watchdog* would be an outright winner. The documentation provided comprises no fewer than ten A5 manuals – placed on top of each other, they make a stack 75mm high. My life is too busy to read through each of these page by page, so the following comments are taken by dipping into the documentation at appropriate places.

The first thing to browse through is the manual entitled *Getting Started*, a slim volume (39 pages) which explains clearly and concisely how to install *Watchdog*. It also points users towards other manuals in a sensible sequence. *Getting Started*, followed by *Setup* and *Quick Reference* as necessary, *Concepts* for general information, and *Advanced Topics* for customisation information. This leaves manuals entitled *PC Data Security: User Guide*; *Secure Drive: User Guide*; *Producing Reports*; *SA Program Guide* and *Composite Index*. The acronym SA stands for System Administrator.

The content of each manual is well-written, and should prove useful at many levels, ranging from new users up to complicated SA usage. Despite the fact that the content is quite good, finding specific items within the documentation is non-trivial. Not only is it necessary to wade through the *Composite Index*, locate the correct volume(s), and then commence searching, but not all the manuals contain an individual index, including, curiously, the starter packs (*Getting Started*, *Setup Guide*, and *Quick Reference*).

I found it confusing having ten different manuals scattered across my desk; almost like reverting to childhood and having lots of small books. It should be possible to keep the number of manuals down – simply having a user and an SA guide would make the situation manageable. *Fischer* states that the number of manuals has now been reduced to three.

Installation

The installation process can either use Basic or Maximum Security (the latter is the default option). Basic Security, which requires 19 Kilobytes of RAM, offers only ID and password sign-on protection, system boot protection, format protection, virus protection, and limited audit reporting. Maximum Security requires 59 KB of RAM (if expanded memory is available, only 19 KB of base memory is used). In addition to all the Basic Security features, this option allows specific subdirectories to be protected on a user-by-user basis, the setting of file access permissions for these subdirectories, data encryption and detailed audit reporting.

I found *Watchdog* easy to install and to deinstall. After the installation program was executed from diskette, *Watchdog* went off for a long think (over one and a half minutes), and then accessed the floppy disk itself for about a minute.

After requesting entry of, and being given, a user ID and password, files are copied from the *Watchdog* diskette to the hard drive. Finally, changes are made to AUTOEXEC.BAT and CONFIG.SYS (the originals are safely backed up), and a reboot required before the changes made come into effect.

Installation offered the choice of ‘Express Install’ (only the SA password and a user ID and password need be specified) or ‘Custom Install’ (*Watchdog* prompts for all system parameters). I chose Express Install: this provided ‘a Maximum Security configuration with transparent interface’, i.e. except for entering their ID and password, users do not know that *Watchdog* is active, unless security features are triggered to forestall an undesirable user action.

One problem during installation was that the last line of my AUTOEXEC.BAT invokes a program called *Norton Commander*: this stays memory-resident and provides a simple front-end through which DOS can be used. *Watchdog* added its controlling program (by which the user logs on to the system) as the last line in AUTOEXEC.BAT. As control never leaves *Norton Commander*, the *Watchdog* control program was never executed. A swift session with a text editor put this right.

Also, my test computer uses a system of multiple boot paths allowing the machine to be booted into one of a number of different configurations. As a consequence, the file AUTOEXEC.BAT contains several separate PATH statements. This confused *Watchdog*: it only altered one. Thus, without changes with a text editor, programs within *Watchdog* were not readily available. *Fischer* has passed both problems to the development team for examination.

When installation was complete, 31 files, occupying 1.56 MB, had been placed in the main *Watchdog* directory (C:\SYSLIB), along with four further System Administrator files (164 Kbytes) in a directory called C:\SYSADM.

Security Administration

A setup program is provided with *Watchdog* which can be used to tailor the product’s operation in almost any way desired. It is to be used by the System Administrator, who has special privileges to configure the system. It is the System Administrator who is responsible for the creation of user accounts and the setting of their privileges.

New users can be set up, and their PC use constrained in almost any desired manner. This goes as far as assigning specific subdirectories to specific projects, and letting the project team be comprised of groups of people. My only criticism of the program is that the seemingly infinite number of choices may introduce too much reliance on built-in defaults. There are, however, centralised administration facilities which may ease the burden in large organizations.

I had minor problems with the password entry mechanism. *Watchdog* is fussy about what makes a valid password, and it refuses to permit passwords which are too short, or which contain any repeating characters. All this was my own fault: *Watchdog* is quite correct to insist on a secure password.

Watchdog Operation

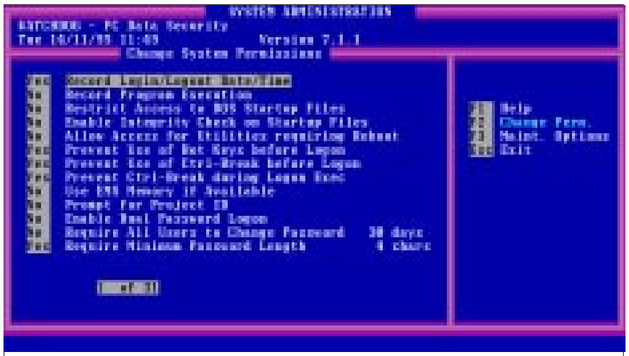
When *Watchdog* is active, it is completely transparent. In most circumstances, once the initial sign-on with a UserID and password has been carried out, the overhead imposed by *Watchdog* (discussed elsewhere in this review) is the only visible evidence that the product is present.

Attempts to access resources or files outside your privileges (these having previously been set by the System Administrator) cause DOS to return an appropriate error. Given that the *Watchdog* overhead is only noticeable when tested with a stopwatch, the adjective 'transparent' seems very apt indeed.

Watchdog did, however, exhibit problems when used with *Norton Commander*. Any attempt to move a file from one subdirectory to another caused the error 'You tried to copy more than one file to a file name, XXYYZZ must be a path name' to be returned by *Norton Commander*. Something is getting very confused, as this error occurred on *all* moves, including attempts to move just a single file!

Control Diskette Booting

When *Watchdog's* 'Control Diskette Booting' feature is activated (the installation of which took a whopping 1 minute 41 seconds), it is not possible to access the hard disk if the computer is booted from a floppy disk. This may be a useful protection were the PC ever to be booted from a bootable disk injected with a virus – whilst the virus would still infect the hard drive, at least it is prevented from making modifications to the hard drive at the file level at this stage. Any attempt to access the hard disk produces an 'Invalid Drive Specification' error from DOS. The drive simply is not there, as far as DOS is concerned.



```
                     SYSTEM ADMINISTRATION
WATCHDOG - PC Data Security
Tue 14/11/95 11:49           Version 7.1.1
         Change System Permissions

Yes  Record Login/Logout Date/Time
No   Record Program Execution                            F1  Help
No   Restrict Access to DOS Startup Files                F2  Change Perm.
No   Enable Integrity Check on Startup Files             F3  Maint. Options
No   Allow Access for Utilities requiring Reboot         Esc Exit
Yes  Prevent Use of Hot Keys before Logon
Yes  Prevent Use of Ctrl-Break before Logon
Yes  Prevent Ctrl-Break during Logon Exec
No   Use EMS Memory if Available
No   Prompt for Project ID
No   Enable Dual Password Logon
No   Require All Users to Change Password   30 days
Yes  Require Minimum Password Length         4 chars

        [ 1 of 3]
```

Watchdog offers a wide range of user-configurable options to set up protection on a PC.

The overhead introduced by Control Diskette Booting was measured by timing how long it took to copy 40 files (1.25 MB) from one subdirectory to another. Without *Watchdog*, these files could be copied in 20.6 seconds, a time which rose to 23.8 seconds when Control Diskette Booting was active, and 32.3 seconds when the encryption associated with Control Diskette Booting was also switched on.

Secure Drive

Supplied with *Watchdog* is a separate program called Secure Drive, an alternative to the Control Diskette Booting feature described above. This component requires that *Watchdog* has previously been installed before it will operate, and that the Control Diskette Booting feature is switched off.

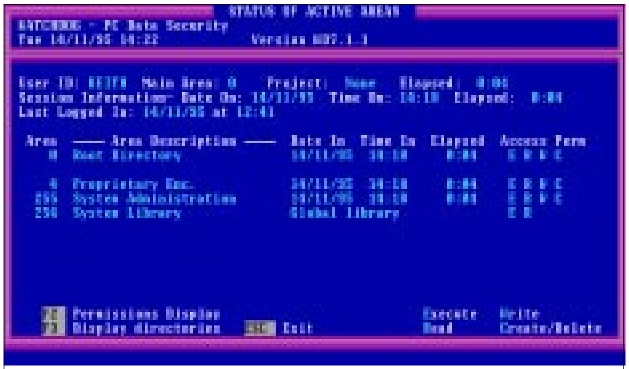
When Secure Drive is installed (which took an age to complete – see below), it encrypts the hard disk to provide a barrier against any attempt to gain unauthorised access to the data stored on the hard disk. A boot password may also be introduced when Secure Drive is installed.

I encountered a fatal error whilst installing Secure Drive; 'Global Error # – 1724'. The documentation did not explain it, and it would not go away. Eventually, I traced the problem to a line in CONFIG.SYS which was a device driver for a Stacker drive, when no such drive was present. Removing this line removed the error message and let me install Secure Drive. *Fischer* states that the documentation is being changed, and the problem resolved.

One unexplained feature of Secure Drive installation is that it refuses to continue until the second diskette is write-enabled. Any program which requires its master disks to have write-protection removed should only be approached with a very long barge-pole. Secure Drive deinstallation also requires that the second floppy disk is left write-enabled. Another change to the manual is required, *Fischer* acknowledges, and says that the manual will in future also advise users to make backup copies of install diskettes before use.

As stated above, installation of Secure Drive takes quite a time to complete, ranging from just over two minutes for Partial Protection, up to over twelve minutes for Full Protection. These figures could be significant if the SA had to install *Watchdog* on a network of computers.

When Secure Drive installation was complete, CHKDSK reported that my hard disk had '415 lost allocation sectors in 31 chains, 849920 would be freed'. The reason for this is that *Watchdog* is preventing CHKDSK from seeing



```
                       STATUS OF ACTIVE AREAS
WATCHDOG - PC Data Security
Tue 14/11/95 14:22            Version WD7.1.1

User ID: KEITH  Main Area: 0    Project:  None   Elapsed:  0:04
Session Information: Date On: 14/11/95  Time On: 14:18  Elapsed:  0:04
Last Logged In: 14/11/95 at 12:41

 Area  ——— Area Description ———   Date In   Time In  Elapsed  Access Perm
   0   Root Directory             14/11/95  14:18     0:04     E R W C

   4   Proprietary Enc.           14/11/95  14:18     0:04     E R W C
 255   System Administration      14/11/95  14:18     0:04     E R W C
 256   System Library             Global Library               E R

    F2 Permissions Display                          Execute   Write
    F3 Display directories    ESC Exit              Read      Create/Delete
```

Watchdog is able to control access permissions on directories to allow different users different sets of rights.

directories to which I did not have access. It is possible to give such programs privileges to allow them to access the whole disk, at the discretion of the System Administrator. In this way it would also be possible to set up an anti-virus scanner's access so that it too could see the whole disk.

Two encryption algorithms are available with Secure Drive: a proprietary (secret) algorithm and DES (Data Encryption Standard). Unfortunately, *Fischer* may only supply DES to overseas countries once the US State Department has granted a licence, so this option was not available in my version. I had to rely on the proprietary encryption algorithm supplied, which *Fischer* informed us was written specifically for the PC and based on a 128-bit key.

Secure Drive can be configured either to encrypt only the parts of the hard disk that control access to the data, or to encrypt *all* data on the hard disk. Clearly the latter will introduce more overhead at installation time (existing data must be encrypted) and at run-time (data must be encrypted before writing and decrypted immediately after reading).

The overhead introduced by Secure Drive was measured in exactly the same way as described above for the Control Diskette Booting feature. Secure Drive proved to be as onerous as Control Diskette Booting when the data itself was unencrypted (23.8 seconds), but slightly slower (34.7 seconds) when the data was encrypted.

Virus Protection

Although *Watchdog* is a general purpose security product, it does offer some features which can be used to protect PCs against the effect of viruses.

Executable programs can reside in a subdirectory which has read-only or execute-only access permission. Therefore, a virus cannot alter such files without knowledge of *Watchdog*. This type of protection can be applied to all components of the operating system.

As the discussion of Secure Drive (see above) makes clear, encryption can be used in many different ways to make it impossible for viruses to access files when the PC is booted in a non-standard manner (e.g. from a floppy disk). Note that encryption does nothing to prevent viruses which propagate by infecting individual files.

All this assumes that viruses use normal paths through the operating system. Any virus that accesses resources at a lower level than that always stands a chance of circumventing any imposed access permissions. In addition, viruses may cause chaos on PCs protected by encryption if they make assumptions about where certain pieces of information are stored and then write blindly to the disk.

However, *Watchdog* does trap Int 13h – the option is called 'Prevent Direct Access to the Hard Disk'. A virus which accesses the system at a lower level than that by dealing directly with the hardware via ports will be much more difficult to deal with in software.

Conclusions

Nobody would use *Watchdog* only for its anti-virus features. There are other competing products available which provide such features without adding the general security baggage accompanying *Watchdog*. Having said that, I have no doubt that installing *Watchdog* will help in keeping viruses at bay. When a hard disk is carefully protected, many viruses will have trouble replicating. The virus may cause damage by writing to erroneous parts of the disk: nothing can stop that as far as boot sector viruses are concerned, but they will not be able to become resident on the PC.

The insistence of the Secure Drive installation program that one of its floppy disks must be left write-enabled is something that should be avoided at all costs. I can see no need for this feature, especially when a blank floppy disk *is* required during installation. This is little more than a method of introducing copy protection without uttering the words.

Fischer has targeted *Watchdog* mainly at large organizations and others wishing to secure multiple PCs on a network. In this regard, *Watchdog* should do the job very well. There are many environments in which it would be advantageous to allow the user only limited access to certain things, whilst still allowing an administrator full control.

Technical Details

Product: *Watchdog* (version 7.1.1, Secure Drive v2.10).

Price: Single-user licence £195; 100 users £105; 500 users £70; 1000 users £68. Other prices on application.

Developer/Vendor: *Fischer International*, 4073 Mercantile Avenue, Naples, Florida 33942, USA, Tel +1 941 643 4803, fax +1 941 643 3772.

UK Vendor: *Fischer International, Systems Corporation (UK) Ltd*, 6 Beaumont Gate, Radlett, Herts WD7 7AR, UK. Tel +44 1923 859119, fax +44 1923 859151.

Availability: Secure Drive requires one floppy disk drive, *MS-DOS v5.0* or higher, a hard disk, and a blank formatted floppy disk. The requirements for *Watchdog* are not stated.

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

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

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

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

Infosec, billing itself as the UK's first dedicated information security show [*what about Compsec – now in its eleventh year? Ed.*], will be held from 30 April to 2 May 1996 at the *London Olympia* (London, UK). It is planned that **the programme will include conferences and seminars on topical security issues**. *Reed Exhibitions*. Information on attending or exhibiting is available from *Infosec* on Tel +44 181 910 7821.

Australia's *Cybec Pty* has announced that its soon-to-be-released anti-virus NLM, *Vet_Net*, has received provisional **certification from *Novell* for *NetWare* 3.x and 4.x**. The product is currently undergoing beta tests in Australia: more information is available from *Cybec*; email info@cybec.com.au.

The next round of **anti-virus workshops presented by *Sophos Plc*** will be held on 17/18 January 1996, at their training suite in Abingdon, UK. Cost for the two-day seminar is £595 + VAT. Any one day (day one: Introduction to Computer Viruses; day two: Advanced Computer Viruses) can be attended at a cost of £325 + VAT. Contact Julia Line on Tel +44 1235 544028, fax +44 1235 559935, for details.

America Online Inc has sent out a letter warning subscribers to its services of **a virus, sent via email, which contains destructive code.** The email concerned comes with the file 'AOLGOLD' or 'INSTALL.EXE' attached, and when downloaded, damages information stored on hard drives. According to an *AOL* spokeswoman, the file is also to be found on the Internet.

Proceedings of the *Fifth Annual Virus Bulletin Conference*, *VB '95*, are now available from *VB* offices. The price is £50 + airmail p&p (England £7, Europe £15, elsewhere £25). To order, contact Petra Duffield at the *VB* Conference Department; Tel +44 1235 555139, fax +44 1235 531889.

On 8/9 January, 5/6 February, and 4/5 March 1996, *S&S International* is presenting further **Live Virus Workshops** at the *Hilton National* in Milton Keynes, Buckinghamshire, UK. The two-day courses cost £680 + VAT, and offer the opportunity to gain experience with viruses within a secure environment. Contact the company for details: Tel +44 1296 318700, fax +44 1296 318777.

McAfee Associates has launched new **anti-virus software for several platforms:** DOS (advanced virus protection, according to a company press release), *Windows 3.x*, *Windows 95*, *Windows NT* desktops, and *Novell/Windows NT* network servers. Its first anti-virus package for the *Macintosh* was also released in December. The company has also launched *WebScan*, which it classifies as the first anti-virus software utility to prevent Internet computer users from downloading virus-infected files and email. *WebScan* can also detect macro viruses such as Winword.Concept. For more information on any of these products, readers can access *McAfee's* Web page at http://www.mcafee.com.

***Eurosec 96* will be held in Paris, France on 26/27 March 1996**. Details from Isabelle Hachin, *XP Conseil*. Tel +33 1 42 89 65 65, fax +33 1 42 89 65 66.

The construction of a new **£1.5 million high-security headquarters** has been announced by *Sophos Plc*. The building, already under way, is expected to be completed by July 1996, and has been designed to meet government standards to carry out even Top Secret work. For more information, access the company's web page, http://www.sophos.com, or email sales@sophos.com.

Reflex Magnetics' Live Virus Experiences are scheduled for 6/7 March, 12/13 June, and 9/10 October 1996. Information on the two-day courses is available from Rae Sutton: Tel +44 171 372 6666, fax +44 171 372 2507.