ISSN 0956-9979 AUGUST 1998

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Flash, bang, wallop:** Win95/CIH has garnered a lot of attention during the last month. This month's editorial (p.2), first virus analysis (p.8) and feature on p.13 cover the virus and some of its ramifications.

• **Watching the detectives?** Peter Morley raises questions about testing procedures and the increasing importance of reviewers testing products' disinfection capabilities. Read his views on p.17.

• **Shimon's story:** The founder of *EliaShim* talks about his day-to-day business and the implications of setting up an anti-virus company in Israel on p.6.

CONTENTS

EDITORIAL
Anti-CIH-pating the Future 2

VIRUS PREVALENCE TABLE 3

NEWS
Marburg Follow-up 3

IBM PC VIRUSES (UPDATE) 4

INSIGHT
Super Gruper 6

VIRUS ANALYSES
1. Flash in the Pan? 8
2. The Worm has Turned 10

FEATURE
Flashpoint 13

TUTORIAL
Free Macro Anti-virus Techniques – Part 3 15

OPINION
Testing Times 17

PRODUCT REVIEWS
1. *Avast32 for Windows NT* 18
2. *Norton AntiVirus for Windows 95* 21

END NOTES AND NEWS 24

EDITORIAL

Anti-CIH-pating the Future

In the future everyone will be famous for fifteen minutes – Andy Warhol.

Win95/CIH has had its fifteen minutes of fame, but what will change as a result? A lot, I hope. I don't *expect* things will change, but many disturbing aspects of the world of software development and distribution have come to the fore following CIH's rapid distribution around the globe.

"Where have you been before *today?"*

The *PC Gamer* incident late in June started me wondering. What precautions do such organizations take? 'Not many' seems to sum it up. Of course, the compilers of these CDs do not say this. They 'use the most up-to-date versions of the best anti-virus programs'. Some also employ the services of external testing agencies, such as the *UKCVCC* (see the Marburg article on p.3), where the 'authority' of an important-sounding, bought-in name is supposed to allay purchaser concerns.

Back when *Dr Solomon's Software* was called *S&S International*, the company provided a cover disk certification service. *S&S* gave this up. It saw the capacity of cover disks soaring inexorably to 1.44 MB (and then on to *multiple diskettes*) and, at the same time, more compression formats and layering of packaging methods were appearing. Eventually, *S&S* decided the risks posed by these factors were unacceptable and the service was withdrawn. If the producer of an anti-virus package, deservedly held in high repute, pulled out years ago due to concerns about capacity and packaging complexity, can a licensee of that and other software genuinely expect to do any better with today's multiple-CD cover disks containing multi-layered compression and distribution packages?

I feel that, as with many other commercial certification schemes, the *UKCVCC* is 'selling a name'. It sounds impressive for *Future Publishing* (and others) to claim their CDs are 'certified virus free' by the *UK Computer Virus Certification Centre*. Certainly more impressive than to say they scan their Internet downloads in-house before burning a gold master CD. Given that these CDs are compiled thus, an event such as struck *PC Gamer* was all but inevitable. Perhaps we should be pleased that it was 'only' Marburg and not CIH!

In fact, *PC Gamer* got off lightly. *Wireplay*, a popular UK dial-up gaming system, shipped a CIH-infected update of their client software for over 27 hours from 3 pm, 20 July. The nature of the *Wireplay* subscription-based system means that most downloaders of the affected software can be tracked by the service provider. What would have happened, however, had this been a typical Internet site with essentially wide-open access and limited abilities to track downloaders?

Another company I spoke with (on condition of maintaining its anonymity) had also discovered CIH. In the last phases of product testing, a self-check routine in the application reported the main program file had changed. Backtracking through the program's development process, a programmer's machine was found to be the initial point of introduction. Unfortunately for the company, that machine had been turned on by a colleague on 26 June (while its user was on vacation) and has since had its motherboard replaced.

Then, on 24 July, CIH-infected drivers for a *Yamaha* CD-R drive were found on a *Yamaha* FTP server. Advocates of scanning archive files should note – the only infected file was compressed as part of an *InstallShield* package that was itself archived into a *WinZip* self-extractor. The latter automatically ran the (non-infected) *InstallShield* setup program. Four or five layers of packing and compression are increasingly common, and an acquaintance with a particular interest in such things has seen as many as *eight* layers in a 'real world' example.

All these examples have been web- or net-oriented and CIH had its initial lucky break via web distribution of infected 'warez'. However, too many people have focused on this aspect of the virus – 'Which sites should I avoid?' is a question I've been asked way too often the last few days. The question you should ask of new software is 'Where have you been *before* today'.

CIH has had its break – we have to hope we can prevent it reaching critical mass.

NEWS

Marburg Follow-up

Following the warning included with last month's issue, the Finnish anti-virus developer *Data Fellows* informed *VB* that the Win95/Marburg virus was also found on cover CDs of the Slovenian and Swedish editions of *PC Gamer*. The former carried the same infected files as the UK issue, but on the Swedish CD, the infected files were:

\patchar\quake2\q2-315~8.exe
\spel\kknd2\directx\ddhelp.exe

This suggests that the compilers of the Swedish CD, after previewing the UK CD and infecting themselves, chose a different selection of software to include on their cover CD.

Data Fellows also informed *VB* that the Italian July/August issue carried no Marburg-infected files.

Virus Bulletin questioned the UK *PC Gamer* CD compiler, Trevor Witt, about the magazine's anti-virus precautions. Apart from running in-house checks with 'up-to-date' anti-virus software, *Future Publishing* contracts virus testing to the self-styled *UK Computer Virus Certification Centre* (*UKCVCC*). Longtime readers with good memories may recall mention of this organization appearing on these pages before (see *VB*, July 1992, p.3).

On 3 July, *Future Publishing* posted a reassurance on their web server that the *UKCVCC* had, on re-testing, been unable to detect any viruses on the July cover CD. A few days earlier, the publishers had received a complaint that some files were infected. Although *Future Publishing* was told the names of the suspect files, this information was not passed on to Dr Simon Shepherd, who *is* the *UKCVCC*.

Questioned about this, Dr Shepherd was not concerned that the file names were not forwarded to him, despite that being a good starting point to investigate claims of the presence of an unknown virus in such a large collection. Instead, the *UKCVCC* was content to run its battery of (reputedly) fourteen commercial scanners over the CD again. When asked which scanners are used, Dr Shepherd indicated *Dr Solomon's AVTK*, *Symantec's Norton AntiVirus* and 'a few others I cannot recall'. [*Dr Shepherd's memory does not appeared to have improved over time. Ed.*]

He seemed particularly enamoured with these products, though unaware of several other top-ranking scanners (according to recent *VB* detection tests). Some of these had recently added detection of Marburg and their use might have saved the embarrassing 3 July denial letter.

Dr Shepherd suggested that it was unreasonable to expect the *UKCVCC* to do better than the most up-to-date scanners. 'It's a belt and braces operation', he said, with the CD compilers, himself and the duplicators all running various virus scanners along the production line ▮

Prevalence Table – June 1998

Virus	Type	Incidents	Reports
Cap	Macro	55	11.0%
CopyCap	Macro	45	9.0%
Laroux	Macro	37	7.4%
Autostart.9805	File	27	5.4%
Parity_Boot	Boot	25	5.0%
AntiEXE	Boot	24	4.8%
Mental	Macro	20	4.0%
Concept	Macro	19	3.8%
Form	Boot	18	3.6%
Empire.Monkey	Boot	11	2.2%
AntiCMOS	Boot	10	2.0%
DelCMOS	Boot	9	1.8%
Junkie	Multi-partite	9	1.8%
Npad	Macro	9	1.8%
NYB	Boot	9	1.8%
Dodgy	Boot	8	1.6%
Mentes	Macro	7	1.4%
Ripper	Boot	7	1.4%
Schumann	Macro	7	1.4%
USTC.7680	Multi-partite	6	1.2%
Extras	Macro	5	1.0%
Sampo	Boot	5	1.0%
Spirit	Boot	5	1.0%
Wazzu	Macro	5	1.0%
Win95/Marburg	File	5	1.0%
Appder	Macro	4	0.8%
Eco	Boot	4	0.8%
Edwin	Boot	4	0.8%
Quandary	Boot	4	0.8%
ShowOff	Macro	4	0.8%
V-sign	Boot	4	0.8%
WelcomB	Boot	4	0.8%
Alien	Macro	3	0.6%
Moloch	Boot	3	0.6%
Temple	Macro	3	0.6%
Others [1]		75	15.0%
Total		499	100%

[1] The Prevalence Table includes two reports of each of: Alien, Moloch and Temple; two each of: Beryllium, Bleah, Counter, CSV.5536, ExeBug, Gest, Groov, Hark, Jerusalem.1363, MDMA, Minimal, Muck, Rapi, Stealth_Boot and Win95/CIH.1003; and a further 45 viruses responsible for a single incident each. Readers are reminded that a fully detailed listing is posted at http://www.virusbtn.com/Prevalence/.

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 16 July 1998. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

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

Baba

CR: Two variants of an appending virus containing the texts '=>COMMAND.COM<=' and 'Hi! hello from MSS!'. The 700-byte variant also contains the text 'HELLO FROM OVER1'.

```
Baba.470        8ED8 B440 33D2 B9D6 01CD 2133 C933 D2B8 0042 CD21 B440 BAC2
Baba.700        8ED8 B440 33D2 B9BC 02CD 2133 C933 D2B8 0042 CD21 B440 BAA7
```

Bachkhoa.3687

CER: An encrypted, stealth, 3687-byte appender containing the texts 'FILE_ID.DIZ','CHKLIST.MS', 'CHKLIST.CPS', 'FILESIGN.SAV', and 'Ha Noi University of technologyYour PC was infected by BACH KHOA virus version 1.22'. Infected files have their time-stamps set to 62 seconds.

```
Bachkhoa.3687   B9D3 03D1 E983 E910 2E31 0783 C302 E2F8 C32E A368 0F2E 891E
```

Blind.346

CR: An appending, 346-byte virus which installs itself in the Interrupt Vector Table. It contains the text '[Blind_Guardian] by Int13h * PARAGUAY'.

```
Blind.346       B440 BA00 02B9 5A01 CD21 E826 00B4 40B9 0400 BA67 02CD 21B4
```

Falopa.548

ER: An appending, 548-byte virus containing the text 'No se metan en el camino de la droga que somos muchos y hay poca. Falopa Virus Parana'. This message is displayed on 25 December. Infected files have the word 4656h ('VF') at offset 0012h.

```
Falopa.548      8BEC C746 02BA BA5D 58CD 213D CDAB 7451 B820 35FE C0CD 212E
```

Hypervisor.3141

CER: A stealth, appending, 3141-byte virus containing the encrypted texts 'GROUPS_I'M_IN', 'IDENTIFICATION', 'SYS:SYSTEM/SYS:LOGIN/NET$BIND.SYSNET$BVAL.SYS', 'COMEXE', 'SECURITY_EQUALS', 'NET$OBJ.SYS', 'NET$PROP.SYS', 'NET$VAL.SYS', 'SUPERVISOR', 'PASSWORD' and 'LOGIN_CONTROL'. Infected files have their time-stamps set to 62 seconds.

```
Hypervisor.3141 3E8B A68C FEFB 061F 2EFF AE8E FE33 C08E D881 2E13 0404 0058
```

Illusion.1328

CR: A stealth, 1328-byte virus containing the texts 'ANTI-VIR.DAT', 'AVP.CRC', 'CHKLIST.MS', 'SMARTCHK.CPS', 'IVB.NTZ', 'CHKLIST.TAV', '->#ThE_WiZArD' and '– – IlluSioN viRus coded by ThE_WiZArD in Spain (1997) – – When you know that your time is close at handMaybe then you will begin to understandLife down there is just a strange Illusion ...'. In order to infect a file, the full file-name has to be specified on the command line (e.g. 'SAMPLE.COM').

```
Illusion.1328   3D66 3075 04B8 5505 CF93 80FF 1174 9A80 FF12 7495 80FF 4E74
```

Kerplunk.3059

CER: A stealth, encrypted, appending, 3059-byte virus containing the texts 'Permanent system error. Please hit the computer NOW!GUESTSUPERVISORSECURITY_EQUALS' and 'Kerplunk coded by Virtual Daemon/SLAM'. Infected files have their time-stamps set to 32 seconds.

```
Kerplunk.3059   6E04 81ED 0300 0E0E 1F07 8DBE 2000 8D96 A60B FFD2 8DB6 2D00
```

Light.1010

CER: A stealth, 1010-byte appender containing the texts 'LIGHT in the DARK' and 'A long time ago,in very remute institut ...'. Infected files have the word 494Ch ('LI') at offset 0003h (COM) and at offset 0012h (EXE).

```
Light.1010      B440 B9F2 03CD D726 8955 1526 8955 17C3 0653 57E8 9400 2EF6
```

Miny.396

CEN: An encrypted, overwriting, 396-byte direct infector containing the texts 'Miny1 Virus - Bug Fix Version 97/08/14', ' (c) Copyleft 1997 by Osiris of CVC,Corea', 'Abnormal Program Termination.', '*.c?m' and '*.e?e'. Infected files have their time-stamps set to 62 seconds.

```
Miny.396        BE03 018B FE8A 2678 02B9 6401 AC32 C4AA E2FA C3
```

Nessy.502

CR: A prepending, 502-byte virus which resides in the Interrupt Vector Table. The payload changes the case of all characters on the screen. Infected files have the word 4E53h ('SN') at offset 000Ch.

```
Nessy.502       BA00 02B9 F601 B440 CD81 F8B4 3ECD 81F8 FA07 1F5A 595E 5F5B
```

Occt.900

CN: A prepending, 900-byte virus containing the texts '????????COM', '*.com', 'c:\command.com', 'OCTT1.0' and 'OCTT V 1.0 (c) by Crazy Wizard'.

```
Occt.900        BAFF FFB9 8403 B440 CD21 1F72 19B8 0042 B900 00BA 0000 CD21
```

Ol.281

CR: A 281-byte appender which resides in the Interrupt Vector Table. The signature in memory consists of the word 6C4Fh ('Ol') at address 0000:03A0h. Infected files have the word E6FFh at offset 0003h.

```
Ol.281          BA00 02B9 1901 8A26 1003 CD21 721D 81F9 1901 7517 2AC0 E81D
```

Opic.727

EN: An encrypted, appending, 727-byte, direct infector containing the texts '*.exe', 'Odessa!', 'Opic', 'config.sys' and 'REM-Odessa Virus (c) Opic [CodeBreakers 98]'. Infected files have the byte 58h ('X') at offset 0012h.

```
Opic.727        B94E 02E8 0300 EB60 90AC D0C0 D0C0 D0C0 D0C0 F6D0 F6D8 D0C8
```

Opic.1712

CN: A polymorphic, appending, 1712-byte direct infector which infects eight files at a time. It contains the texts '**********PROSPERO!********** There is a path to the trancendece of the dollar: Embark rich beggars! Does magic bring prosperos to his knees? Reading pretty twilight, making grass uncertain? Oh,all that christmas snow shouldered by one birthday suit! The fate of the world under his armpit like a thermometer? Rejoice Villains! Your time has come. **********(C) Opic [CodeBreakers,98]**********', '*.c*' and 'Prospero Virus(C) Opic [CodeBreakers '98]'. The virus' polymorphism is table driven and it uses one of seven encryption schemes. The following templates cover all possible variations.

```
Opic.1712       1790 ??90 ACF6 D051 90B1 04D2 C890 5990 F6D0 90AA 90E2 ECC3
Opic.1712       00EB 1790 ??AC 5190 B104 D2C0 90F6 D8D2 C090 59AA 90E2 EEC3
Opic.1712       EB17 90?? 90AC 90F6 D851 B104 D2C8 5990 F6D8 90AA 90E2 EDC3
Opic.1712       1790 ??90 90AC F6D0 903E 3286 2601 90F6 D090 90AA E2ED 90C3
Opic.1712       EB17 90?? AC51 9090 B104 D2C8 F6D0 90D2 C890 59AA E2EE 90C3
Opic.1712       1790 ??AC 51B1 043E 3286 2601 D2C0 3E32 8626 0159 AAE2 ECC3
Opic.1712       1790 ??AC 51B1 04D2 C890 3E32 8626 01D2 C890 59AA 90E2 ECC3
```

Overdoze

CR: A family of appending, viruses containing the texts '[Overdoze] (c) 1994 The Unforgiven/Immortal Riot' and 'Dorked with by the EVG/Executioner'. Infected files have the byte 56h ('V') at offset 0003h.

```
Overdoze.568    2EA3 DD01 B440 80CC 00B9 3802 CD21 B442 B000 538B D92B CB5B
Overdoze.572    2EA3 DD01 B400 80CC 40B9 3C02 CD21 6800 4258 528B D12B CA5A
Overdoze.582    2EA3 EB01 B440 B946 02CD 2132 E424 0080 CC42 0C00 C1E9 10CD
Overdoze.585    2EA3 EC01 B440 B949 02CD 21B8 0042 2BC9 CD21 B440 BAEB 01B5
Overdoze.588    2EA3 F501 B440 80CC 00B9 4C02 CD21 6800 4258 83E1 00CD 21BA
Overdoze.591    2EA3 F801 C0E4 08B4 40B9 4F02 CD21 578B F82B F88B C75F 0D00
Overdoze.596    2EA3 F101 B437 80C4 0880 C401 B954 02CD 21C1 E810 80CC 420C
Overdoze.606    2EA3 FF01 B440 B95E 02CD 21B8 0042 33C9 CD21 BAFE 01B4 0080
```

PKZ.325

CEN: An encrypted, overwriting, 325-byte direct infector containing the texts 'XPEH.COM', '*.Dom' and '[LeV.PKZ.OW.Var]'.

```
PKZ.325         482E 434F 4D5E C606 1D01 00BE 3901 B924 00B0 ??30 0446 E2FB
```

Replicator.472

CR: An appending, 472-byte virus containing the texts 'V1.01' and two other messages written in Russian. Infected files have the byte 56h ('V') at the end of their code.

```
Replicator.472  8916 CB02 890E CD02 31D2 B9D8 01B4 40CD 21B8 0042 31D2 31C9
```

Replicator.767

CER: An appending, 767-byte virus containing the partially Russian text that could be read as '*V3.01 14/02/1996 from PTI of StGTU*'. Infected files have the byte 5Ah ('Z') at the end of their code.

```
Replicator.767  B9FF 02B4 40CD 21B8 0042 31D2 31C9 CD21 5A58 813E DA02 4D5A
```

Replicator.815

CER: An appending, 815-byte virus containing the texts ' High Hopes ' and '*V3.04 23/02/1996 from PTI of StGTU*'. Infected files have the byte 5Ah ('Z') at the end of their code.

```
Replicator.815  B92F 03B4 40CD 21B8 0042 31D2 31C9 CD21 5A58 813E 0A03 4D5A
```

Replicator.888

CER: An 888-byte appender containing the texts 'LIFE IS SHIT !', '*V2.04 7/02/1996 from PTI of StGTU*', 'FUCKEN LIFE..' and 'Nave a NICE day !'. Infected files end with the byte 5Ah ('Z').

```
Replicator.888  B978 03B4 40CD 21B8 0042 31D2 31C9 CD21 5A58 813E 5303 4D5A
```

Rift.467

CR: An appending, 467-byte virus containing the texts '*.COM' and 'Rift Villy v.3.0'.

```
Rift.467        B906 00F3 A4B4 20CD 2181 FBBE 4074 4A8C C848 8ED8 BB03 003E
```

SRX.2304

CER: A prepending (COM) and appending (EXE) 2304-byte virus containing the texts '*.COM', '*.EXE', 'CHKLIST.CPS', 'CHKLIST.MS' and 'Bad command or file name'. The payload, which triggers on 12 December, displays the usually encrypted text '25 WAYS TO PREVENT A VIRUS ATTACK.... No.2 ALWAYS USE CONDOMS !! No.25 SELL YOUR COMPUTER !'. Infected COM files have the string 'SRX' at offset 0003h.

```
SRX.2304        B440 B919 000E 1F8D 166C 019C FA2E FF1E 0801 2E8B 0EBF 012E
```

Usher.553

CR: A 553-byte appender containing the ecrypted text ' -= PERSIAN GULF =- Hey you! Jump to the FIRE! This is a VIRUS, By:Aref Karimi (Wizard) The Last days of 1375. Be happy, he he .'.

```
Usher.553       80FC FD75 0332 E4CF 80FC 4B74 03E9 7801 1E50 5351 528B DA43
```

XBM.823

CR: An encrypted, appending, 823-byte virus containing the text '[XyeBo_MHe], (c)Midnigh+Pr0wler'. Infected COM files start with the word E940h and infected EXE files have the byte 34h at offset 000Ah.

```
XBM.823         BE68 00E8 C3FF B440 33D2 B937 03E8 0700 BE68 00E8 B3FF C39C
```

INSIGHT

Super Gruper

Shimon Gruper and his good friend Eli Gamush served together in the Israeli airforce during their four years of compulsory military service. In 1984 they realized their dream of setting up *EliaShim*, a company which provided computer programming services. Shimon recalls, 'Initially, we developed various applications for large companies. 1985 saw the beginning of downsizing from mainframes to PCs. Little by little, we got involved in security issues when our clients became concerned by the fact that PCs had no built-in security.'

In 1985, *EliaShim* developed one of the first PC security and access control products – a combination of hardware and software. The PC's resources were inaccessible to users who did not follow correct login procedures, complete with identification and password. It marked a turning point for the company – 'Since then, *EliaShim* has dealt only with security issues. Since I decided that viruses form a security threat, adding anti-virus technology to our security line was a natural thing.'

The Promised Land

Despite the tragedy of losing his young business partner in 1990 – Eli (29) passed away while on a trip to open *EliaShim's* US branch – Shimon remains optimistic and confident about the company's future. He admits that the Israeli market is currently suffering from a slowdown, mainly due to stagnation in the political arena.

On the other hand, he says 'Israel has more hi-tech startups than any other place apart from Silicon Valley, and foreign investments are flowing into them. There is a very high number of computers per capita here, and the majority of them are connected to the Internet. Overall, the market for software, and especially security software, is good.'

In fact, he is quick to point out the benefits of starting an anti-virus company in Israel. Given the access to very high quality engineers and programmers, salaries are significantly lower than in the US. One of the biggest sources of recruitment is from Russian immigration. One million Russians settled in Israel in the last six years, adding almost 20% to the country's overall population of 4.5 million, and making up at least half of the IT sector. Shimon himself was born in the former USSR, now the Ukraine, and moved to Israel at the age of twelve. He reckons that the biggest problem *EliaShim* has is proofreading the English text written by its Russian programmers!

Today, as well as maintaining his position as President and CEO of the company, Shimon is also the Chief Technology Officer – a role he obviously relishes. He sees himself as the organization's visionary, 'I speak to industry analysts, the press and at conferences. My favourite part of the business is learning about new technologies and trends in the industry and thinking how we can use and implement them in our products. I also look for strategic partners to see who can benefit from our technology.'

First Encounters

At college, he studied 'Instrumentation and Control', switching to Electronics in his second year, but it was at high school in the 1970s that his experience with computers began. 'The first computer I worked on was a Data General NOVA minicomputer with a 32K memory and 512K hard disk. The first language I learnt was BASIC. During high school I designed and built my first microcomputer based on a *Motorola 6800* CPU. It had 4K of memory and I succeeded in installing a 3K BASIC interpreter on it, which left me 1K to write programs!'

In college, he built his second computer, based on a Z-80 CPU and S-100 bus. It had eight-inch floppy drives and a 32 KB memory, running the CP/M operating system and introduced him to assembly: 'I did a lot of programming in assembly language, writing utilities and enhancements for CP/M.' Even during his time in the airforce, he kept up his knowledge of computers, teaching the design and programming of computer-based equipment.

The first computer virus Shimon saw was brought to him in August 1987 by the head of a university computer laboratory, who described the program as invasive and went on to explain how it attached itself to other programs. Immediately interested in the concept of replication and spread, Shimon began disassembling and debugging what was later known as the Jerusalem virus to find out how it worked.

He remembers the media frenzy which followed all too well. 'I forgot about it for a while, as I could not see any useful implementation for it. Suddenly, on a Friday in October 1987, there were headlines in all the major Israeli newspapers and on national TV about the new threat discovered by Jerusalem University. That was the first time I heard it referred to as a virus.'

Having studied its behaviour, Shimon started to consider protection against the Jerusalem virus. Later that week he wrote a resident program (TSR) that prevented viruses from attaching themselves to other programs. The idea was to provide a generic solution rather than a specific one. He was sure that this virus was just the beginning, and that hackers would write and spread more like it.

'I pondered the idea of generic protection for a long time, and it forms part of our *ViruSafe* anti-virus software to this day. Unfortunately, the entire anti-virus industry has

focused on scanning. I cannot educate the market single-handedly, which is why, in 1988, I added signature scanning to *ViruSafe* as well.'

His experience stood him in good stead when the Michelangelo virus came along. Convinced that the panic was exaggerated and that the news was hype, he suspected that the newspapers would accuse anti-virus vendors of crying wolf when no disaster occurred on 6 March. Fortunately, anti-virus vendors were not blamed, and Shimon now recognizes that the Michelangelo experience has been beneficial, 'everybody now understands what a computer virus is and why they need anti-virus software.'

A day in the life of *EliaShim's* President shows a man who is dedicated to keeping up with the constant evolution and development of his chosen field. After grabbing some coffee and answering what emails he can, he spends until noon in meetings with his senior management team, and on the telephone to *EliaShim's* offices in Japan and Holland.

In the afternoons, he concentrates on the United States branch of the company. Once a week, he hosts a meeting for all the employees, updating them on company business. Between 6pm and 9pm he enjoys time with his family – his wife Emily and their two children. Then it is back to work, holding conference calls with the US office and finishing off any outstanding business.

Looking Ahead

Shimon has very definite views on the future of the anti-virus industry. 'I think it will evolve towards more comprehensive anti-vandal solutions that protect users from all kinds of malicious programs and not only those that replicate. These days, users cannot distinguish between different types of malicious program. They do not care if they replicate or not and demand full protection. Soon, scanning for strings of known viruses will not be enough any more. Other technologies will be involved to provide comprehensive protection.'

Currently, *EliaShim* is focusing on Internet security, providing protection against malicious executables like Vandals, malicious programs that are automatically downloaded and executed from the Internet. Shimon explains, 'Vandals can be in Java Applets, ActiveX controls, plug-ins, email attachments and so on. They are designed to cause damage to computers, steal information and even money. Vandals, unlike viruses, are hit-and-run. They come into the computer, do the damage and then disappear. Thus, it is impossible to scan for Vandals, since we cannot catch them. I recognized this trend more than a year ago, which is why we have pioneered the anti-Vandal concept.'

When asked about his views of the ethics of virus writing he replied 'Obviously, I'm against writing viruses and I think that there should be a law against writing malicious code in every country'. However, he agrees that most viruses are not written with malicious intent, and that most virus writers 'are in it for the fun and the challenge'. He believes that a hefty fine would reduce numbers. Like many other anti-virus developers, Shimon is aware of the 'opportunity' that the Internet now offers virus writers.

'Unfortunately, along with all the wonderful things, the Net also allows the writing of Vandal applications that can be specifically targeted and used to generate profit. One example is a plug-in program on a web site in Canada that offered free porno movies. After a user had downloaded this plug-in into his Netscape, it silently shut down the modem and dialled a 1-900 number [pay per call] charging $5 a minute. *AT&T* admitted that it had refunded $2.74 million to 38,000 customers. This means that someone succeeded in putting this sum into his pocket. Knowing the mentality of virus writers, I do not think that they would actually be capable of such things. However, there will always be "entrepreneurs" who employ virus writers to write Vandals for them'.

Home Sweet Home

Shimon Gruper is a surprisingly down-to-earth and compassionate man considering his powerful position in a competitive industry. He is woken up by the family's cat, which has taken to jumping on his bed in the mornings. Perhaps it knows that Shimon was against pets in the house until he found the one-day-old kitten in his back yard. 'It was so small, and still blind, I could not resist.'

Despite not finishing work until the small hours, his favourite hobby is reading. He tries to read at least one book a week. Frequent travel to the US and Europe means that much time in the air is spent with a book in his hands.

Shimon's constant awareness of the current political situation in Israel, while touched with a personal urgency, is also mixed with characteristic humour and optimism. 'I am strongly in favour of peace with our neighbours and as soon as possible. I would hate it if my son had to go into the army and actually fight a war. I think that the current situation is hurting both our economy and our relationship with other countries. I would really like to see the day when Israel does not make the headlines on CNN, and when the first item on Israeli news is not a terrorist bus bomb but a story about our Prime Minister's love-life!'

VIRUS ANALYSIS 1

Flash in the Pan?

Richard Wang
Sophos Plc

Since the introduction of flash BIOS technology in desktop PCs, the possibility of a malicious attack on the BIOS has been recognized within the computer security industry. The first virus to succeed in targeting a PC this way is now known to be in the wild and spreading across the globe.

In June of this year, a file received from a customer was found to be infected with a new Portable Executable (PE) infector. Called CIH, it works under *Windows 9x* and at least four variants are now known. These are internally labelled 'v1.2', and 'v1.4' with two self-identified as 'v1.3'. Although *Windows NT* also uses PE files, CIH uses mechanisms which do not work under *NT*.

The variants have identical payloads, and differ by only a few bytes. There are also different trigger dates – 26 April, 26 June and the 26th of any month. Once the virus had been identified, it was found to be widespread in Taiwan, where it is believed to have originated, and in the wild in several other countries. [*Confirmed reports have been received from Australia, Chile, France, Germany, Korea, Norway, Russia, the UK and the US. Ed.*]

Initially, CIH attracted attention because it used a new file infection mechanism – the fragmented cavity attack. However, it quickly became apparent that the potential threat posed by its payload made it of more than just technical interest. The payload was thought to consist of a routine to write garbage to each hard disk but, just days before the 26 June trigger date of two of the variants, the full extent of the payload started to become clear.

Any machine using one of several popular *Pentium* chipsets and a flash BIOS chip from one of at least two different manufacturers is vulnerable to the attack on the BIOS itself. In vulnerable machines, a small but critical section of the BIOS is overwritten by CIH, leaving the PC unbootable.

This was no longer a run-of-the-mill disk-trasher!

Payload

Although the technical details of CIH's infection mechanism are intriguing for the virus researcher, its payload is what sets it apart from other viruses. The payload consists of two parts, both of which are triggered when the right conditions are met. As the payload is part of the infection mechanism, it is not triggered until the virus is resident in memory. The trigger condition is met when a file which has an EXE extension, but which is not a suitable host, is opened on the trigger date.

The second part of the payload is common. It overwrites the first 2048 sectors (1 MB) of each hard disk in the system with random data from memory. Anything overwritten in such a manner will be difficult or impossible to recover. The virus looks for further disks indefinitely and the machine – despite running the hard disk continuously – is unresponsive to user input.

The first part of the payload code to trigger is what has given CIH the world's sudden attention. Flash ROM technology has existed for several years. Having the BIOS 'flashable', by storing it in such a chip, has allowed the basic bootstrap procedure and I/O routines of the PC to be rewritten by software. Earlier EPROM technologies allowed reprogramming the BIOS, but required the chip to be removed, erased under ultraviolet light and reprogrammed in dedicated hardware.

Flash ROM technology meant that BIOS upgrades and bug-fixes could be implemented more cheaply and easily – in fact, by the user – rather than requiring special skills and equipment. The dangers of allowing software to rewrite critical sections of a machine's internal bootstrapping code have been discussed (Jakub Kaminski, *VB'95 Conference Proceedings*), but until now, no virus was known to have exploited this successfully to cause damage.

The most likely reason for the previous lack of such a payload is the complexity of the task. A detailed knowledge of both the motherboard chipset and the BIOS Flash ROM itself is required in order to write directly, and successfully, to the BIOS. The Flash ROM used to store the BIOS will normally have a built-in mechanism to prevent accidental writes to it by electrical 'noise' during power up or down, or through instabilities in the power supply.

Thus to write directly to the BIOS, a program must gain write access to the Flash ROM via the motherboard chipset and then disable the chip's own protection. The information needed to do this is readily available from the chip manufacturers. To be able to rewrite Flash ROM, a programming voltage, or V_{pp}, typically of 5 V or 12 V is also necessary. This voltage is often set by a jumper on the motherboard. However, in order to facilitate BIOS upgrades, many machines are shipped with the write voltage enabled, leaving the BIOS vulnerable (see p.13).

The routine CIH uses to gain direct access to the BIOS chip is known to work with the *Intel 430TX* chipset, but it is likely it works with most, if not all, Pentium chipsets. Some late-model 486 chipsets may also be vulnerable. The programming sequences sent to the BIOS itself allow writing to work for at least two ROMs from different manufacturers. The damage is done by writing one byte to the BIOS boot block – the area containing code for initial hardware tests and bootstrapping the machine.

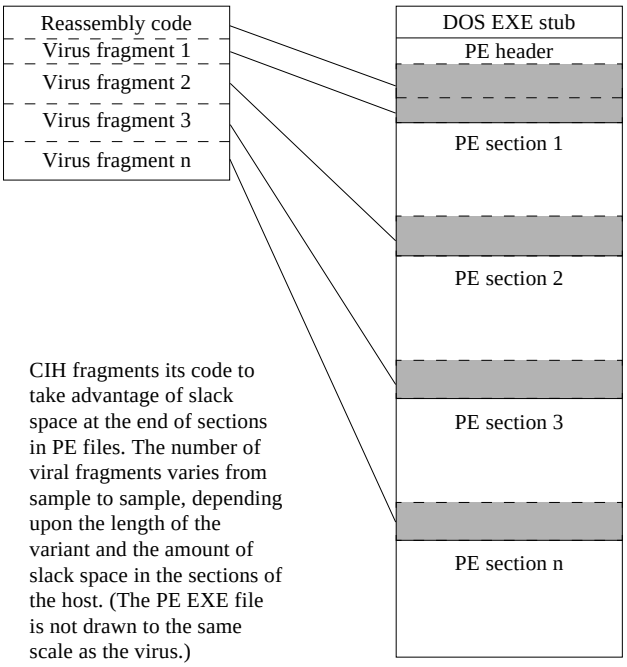
Unlike most RAM, Flash ROMs cannot be written to in arbitrary-sized chunks. Many ROMs have page sizes of 128 bytes for their boot blocks. Thus, although CIH only writes a single byte, the 127 bytes surrounding it are reset to FFh. When the machine next boots, the initial BIOS boot block will contain invalid code. This effectively kills the machine until a new BIOS is installed. In cases where the chip is soldered onto the motherboard, rather than plugged in a socket, a change of motherboard will probably be required.

Infection Mechanism

PE files are executables used by *Windows 9x* and *Windows NT*. A PE file consists of a DOS executable, usually just a stub that indicates the program should be run under *Windows*, a PE header section and several data objects. These objects can contain executable code, information on imported and exported functions, data or relocation information. Each object following the PE header must be aligned within the file to start on a boundary that is an even power of two, between 512 bytes and 64 KB.

Any difference between the length of useful code or data in an object and the chosen section alignment is normally padded with nulls by the linker. Information on the alignment of the objects and the size of each object is stored in the PE header and in a series of object tables just after the header. Typical PE files contain five or six objects, all of which have some space that is effectively wasted. It is in these areas that CIH stores its code, thus infecting a file without increasing its length.

Cavity infectors, which seek a 'hole' within a file large enough to hold the entire virus, are not new – in fact Lehigh, one of the first file infectors, used exactly this technique. However, the approach CIH takes means a greater number of files are potentially infectible.



CIH fragments its code to take advantage of slack space at the end of sections in PE files. The number of viral fragments varies from sample to sample, depending upon the length of the variant and the amount of slack space in the sections of the host. (The PE EXE file is not drawn to the same scale as the virus.)

CIH breaks its code into chunks that it uses to fill 'slack space' at the end of the sections in its hosts. It checks for files with insufficient slack space, refusing to infect them. However, this is unlikely, since all known variants are just under 1 KB long and most PEs will have at least that much free space. A peculiarity of the *Borland* linker means files produced by it are uninfectible.

When infecting a file, CIH builds a table of data about the length and location of its code fragments. This, and the minimal code to allocate memory for itself and to piece its code fragments back together, is stored between the PE header and the first object of the host. If there is insufficient space in the header to take this crucial data and code, the file is also deemed uninfectible.

Running an Infected Executable

Executing an infected file on an uninfected machine causes the virus to hook Int 03h via the interrupt descriptor table and install its own interrupt handler. This makes debugging more difficult and allows the virus to run code at ring zero (with CPU supervisor privileges). It then calls the hooked interrupt and checks the value in Debug Register 0 (DR0). CIH assumes it is already resident if this is non-zero, transferring control back to its host program.

Otherwise, it places a non-zero value in DR0 and allocates memory in the VxD area, in which to reassemble itself. The use of debug registers for data storage (DR1 is also used during file infection) seems to be an attempt to further complicate tracing and analysis. As some debuggers use DR0 and DR1, tracing CIH with them will result in a misleading view of how the virus works. After allocating memory, the virus pieces its own code back together from the data in its fragment table. Once reassembled, CIH calls its own Int 03h handler again and, using the accompanying privilege level, installs a hook trapping calls to the file system. Control is then returned to the host program.

Protected mode interrupts are allocated a CPU privilege level at which their code runs. For Int 03h this is usually ring zero. The pointer to the interrupt handler can, however, be altered from ring three, thus CIH can gain ring zero access to the file system from an infected user application.

CIH sits in the file system API chain waiting for EXE files to be opened. On receiving such a call, it checks whether the file is a PE that is not already infected. Files with a non-zero value in the byte immediately before the PE signature are considered infected. The virus itself writes the first byte from the Ring0_FileIO routine into this location when infecting files. This is usually 55h ('U') – the PUSH EBP opcode. When a potential target is found, its header and object table are processed to determine how much of the virus can be placed at the end of each object. There is no lower limit to the code fragment size CIH will place at the end of each object. In testing, it readily inserted a one-byte fragment into a specially modified PE, using eight bytes of its fragment table space to record the fact!

However, there must be sufficient space between the header and the first object to hold all of the code necessary to reassemble the virus, allocate memory and install the file system hook. PE files with insufficient space are marked as infected – presumably to save reprocessing the header in future – but otherwise left unmodified. If there is space in the file, the virus code sections and an amended PE header are written and the file system call passed on.

Defending Yourself

Obviously, the first line of defence is regularly updated anti-virus software. Beyond that, removal of the jumper supplying the programming voltage to the Flash ROM will protect the BIOS itself from attack, but will not prevent the destruction of hard disk data. Unfortunately, as designs increasingly conform to *de facto* standards, and with downward pressure on prices, V_{pp} is often hardwired to save costs. [*A more complete discussion of these issues is included in the accompanying article on p.13. Ed.*]

Conclusion

Although, technically, the virus only damages the firmware of the BIOS, the end result is that fixing its damage requires some form of hardware replacement. CIH can therefore be said to have a similar effect to that of a hardware-damaging virus. It is to be hoped that it is not the first of many.

CIH

Aliases:	Win95/CIH, CIH.Spacefiller, PE_CIH. Variants of 1003- and 1019-bytes, and two of 1010-bytes are known.
Type:	Fragmented cavity infector utilizing PE section slack space.
Infection:	Hooks the *Windows 9x* file system API.
Self-recognition in files:	A non-zero byte immediately before the PE header.
Self-recognition in memory:	A non-zero value in DR0.
Hex pattern:	All variants.

```
E800 0000 005B 8D4B 4251 5050
0F01 4C24 FE5B 83C3 1CFA 8B2B
```

Payload:	Overwrites 2048 sectors at beginning of each hard disk and overwrites part of the Flash BIOS boot block.
Trigger:	Opening an uninfectable EXE file on 26 April (CIH.1003, CIH.1010.A), 26 June (CIH.1010.B) or on the 26th of any month (CIH.1019).
Removal:	Boot from a clean floppy disk, delete infected files and restore from backups.

VIRUS ANALYSIS 2

The Worm has Turned

Sarah Gordon
IBM Research

One of the most interesting developments in computing over the last few years has been that of the now-ubiquitous Internet. Twenty five years ago, when the late John Brunner told a tale of a computer program which could crawl through an Internet-like network of computers, the Internet was largely the stuff of science fiction. *Shockwave Rider*[1] is now 'required reading' for any fledgling hacker, cyberpunk wannabe, or serious student of Internet history and culture.

By 1982, ARPANET (the beginnings of the Internet) was being implemented. Real experiments with worm programs for beneficial tasks were being conducted at the Xerox Palo Alto Research Center.[2] The concept of a tapeworm program crawling through networked systems outside the research labs saw a hint of reality when, in 1988, Robert Morris Jr brought parts of the infant Internet to, if not a grinding halt, a slow crawl. He lost control of one of his projects, known ever after as 'The Morris Worm' or 'The Internet Worm'.

The main condition that prevented complete chaos as a by-product of that worm was the non-standardization of Internet-connected machines – the diversity of operating systems and architecture. While ten years ago the idea of one's desktop machine participating in a worldwide network of computers was just beginning to take flight, today modern operating systems almost invariably possess some form of Internet connectivity. Another interesting trend has been an increase in the homogeneity of the global network. Whereas the Internet once consisted of many different flavours of machines loosely grouped together as 'Unix', the Net has grown in such a way that there are now large populations of very similar machines.

While this similarity allows easy creation of software that can be run on a large number of machines, it also allows the homogenization of some attack strategies. That there are so many 'like-minded' machines on the Internet allows an attacker to penetrate many machines using the same technique, attack, or program should he find a security hole.

The Morris Worm was designed to spread from machine to machine, utilizing security holes of which most hackers and many administrators were already aware. Due to the large number of machines that were vulnerable to the worm's penetration technique, it was an overnight sensation. Estimates are that hundreds, or perhaps thousands of machines were compromised within hours of its release[3].

Since then, worms have come and gone without much fuss. There have been some isolated outbreaks, but generally the machine population susceptible to each worm has been

sufficiently small and diverse to diffuse the problem. However, the conditions have become increasingly favourable to the creation of a new, viable worm. Enter Admw0rm, a worm capable of spreading from one *Linux* to another, utilizing a hole in the Berkeley Internet Name Domain server, BIND.

The Ties that Bind

BIND is an implementation of Domain Name Service (DNS), written primarily for Unix Systems. It consists of three parts – client, server and tools. BIND is integral to the Internet, but has had some glitches. There have been vulnerabilities that result in the saving of incorrect (sometimes maliciously selected) data by a remote name server (cache poisoning), giving the possibility of corrupt or compromised information; denial of service attacks; and problems resulting in remote crashing of systems.

While problems in BIND are found and fixed, unpatched versions of the package remain on many machines. It is this fact that makes the discovery of the Admw0rm so unsettling. This program demonstrates yet another BIND vulnerability/compromise. BIND 4.9 versions prior to 4.9.7 and versions 8 prior to 8.1.2 have a vulnerability which, if properly exploited, can result in an intruder gaining access to the computer. (The worm takes this one step further, and does this compromise/access automatically.)

These vulnerable versions of BIND do not carry out proper bounds checking when they respond to an inverse query request. The worm begins its attack by making precisely that request, which, if successful, results in the creation of an SUID (root) shell on the newly compromised machine.

Packaged to Go

The version of Admw0rm that has been distributed widely amongst the 'hacker' community is designed to attack the popular *Linux* operating system. The worm comes in a tar archive, complete with a README and a copy of the CERT advisory alerting system administrators of the BIND vulnerability, as well as full source code. In fact, the worm package is comprised entirely of C and shell script – all binaries are built by the handy install script included.

The worm starts with the execution of an install script, called, creatively enough, 'startup', written in standard Unix Shell command language. It builds several binaries and creates a zipped tar file of them for later use. It then adds itself to the password file of the new machine, creating an account with no password, before launching the main worm script, Admw0rm.

As diddling with the password file requires root access (if not, your system is in so much trouble, this worm will be a drop in the ocean) most users will not be able to launch the worm. It would, however, be relatively easy to bypass this limitation; in many cases, it is also relatively easy to obtain root access on many systems, but that is another topic.

The Body

The body of the Admw0rm, apparently the product of the 'ADM Crew' (claiming, somewhat optimistically, that the 'THE CREW WILL NEVER DIE'), is also a simple shell script. This script calls in turn: the previously compiled binary GimmeRAND which creates a random IP address, each octet in the range 1 to 231; Incremental – a simple shell script which, when given a starting IP address, creates a list of IP addresses by incrementing the starting value; and Scanco – a binary which attempts to connect to a remote machine via UDP port 53, on which BIND is normally located.

The result of the preceding contortions is to create a loop, into which IP addresses of machines which appear to be running named [*'name dee' – the name server daemon. Ed.*] are passed. Inside this loop, a test is first made to see if the remote machine is vulnerable to attack. In order to do this, the worm executes another compiled binary which tests whether 'named' is running with 'iquery' enabled. If it is, the replication routine is called.

Replication

When a vulnerable machine is found, the exploit executable is called. This utilizes the aforementioned named buffer overflow problem to perform the initial compromise, and then executes a string of shell commands. These commands create a new user called worm on the target machine, with a null password. Furthermore, /bin/sh (the usual location of the standard Unix shell) is copied to /tmp/.worm, and made SUID root. Finally, any restrictions upon those hosts which are allowed to connect to the machine are lifted by the deletion of the hosts.deny file located in /etc.

With this damage successfully carried out, the worm runs a utility that allows for remote submission of commands to a machine via telnet. This is called with instructions that cause the following operations to be attempted. The named daemon is restarted, as the previously used buffer overflow usually causes it to crash. A file /tmp/.X11x is created, which by default is an html page containing the text 'The ADM Inet worm is here !'. A directory /tmp/.worm0r is then created. The worm checks if it is already installed on this machine by looking for the existence of the file /tmp./wormishere. If found, the worm deletes all files in /var/log (the default location of the system log files) and the file /tmp/.worm. If the worm is not already installed, it uses the ftp protocol to connect to the source machine, and copies the tgz file (created above) from it.

The tgz file is decompressed in the directory /tmp/.worm0r and the self-infection check file /tmp/wormishere is created. Admw0rm's main body is executed as a background task while the IP address of the newly compromised machine is emailed using /bin/mail to a definable email address. Finally, all files named index.html are replaced with the same html page described above but there is a trivial error in this routine that prevents this operation from completing.

The Good News

The good news is that although the Admw0rm is a viable worm for the *Linux* operating system, it contains a number of assumptions and limitations. The most obvious is that the exploit used to travel from machine to machine is only present in some versions of BIND, and even then, only if the inverse query feature is turned on – it is off by default. The vulnerable versions of BIND have been superseded by ones that are not open to this particular attack. However, as was seen during the time of the Morris Worm, the availability of a patch or upgrade does not necessarily correlate with its being installed on vulnerable systems. The worm also assumes that the source machine has both telnet and ftp services enabled. Although these services are often on by default, many machines do not have them.

These limitations are responsible for restricting potential damage by the worm. Had the problem been manifested in the default configuration of BIND, the situation would most likely have been worse. That said, some administrators are seriously concerned about the number of compromised systems and Admw0rm's general availability. The fact that so many machines have been compromised using buffer overflow exploits should merit the attention of anyone responsible for the security of systems running BIND.

In fact, the CERT advisory in which this vulnerability and exploit was mentioned stated 'The CERT Coordination Center has received reports of new kinds of intruder activity indicating that intruders are targeting machines running vulnerable versions of "named" (domain name server software that is part of BIND). Thousands of sites running unpatched, vulnerable versions of "named" are known to have been compromised through exploit methods discussed here and in CS-98.04.'

According to CERT[4], you can tell if your version of BIND is vulnerable to this particular buffer overflow exploit. For BIND 8, if your fake iquery parameter is set to 'yes' (the usual file name where you will find this is /etc/named.conf), you are vulnerable. You should remove that option, or set it to 'no'. For BIND 4.9, look for the line containing 'Fake-iquery' in the /etc/named.boot file. If it is there, you are vulnerable – remove it. Make sure that the line defining INVQ is commented out in your conf/options.h source; otherwise, inverse query support will be added at compile time. (If your nslookup is old, disabling inverse query support may cause it to fail. If this happens, replace nslookup with a more current version.)

These same vulnerable versions of BIND have other problems which are not fixed by the suggestions above. According to CERT[5], the best option is to upgrade to BIND 4.9.7 or BIND 8.1.2 as appropriate.

The Bad News

The bad news is that this is just one of several which have been observed in varying stages of 'development'. The original warning, issued on 28 May stated 'While intruders appear to be using tools that exploit this vulnerability on Intel-based machines, it would not be difficult for intruders to adapt existing tools to exploit the vulnerability on other architectures.'.

In fact, we have observed ports of the worm to other operating systems, as well as learning recently of several variations on the general theme. Some of these worms are even capable of performing system trojanization[6], making their presence invisible to the casual user as well as the reasonably experienced administrator.

While I have gone to great lengths to explain why the Admw0rm is not the end of the Internet-As-We-Know-It, it represents a disturbing trend which some have been predicting for a long time. As the number of machines connected to the Internet increases and the software they run becomes more homogenized, the environment becomes less and less hostile to worms. Indeed, should any one platform and operating system (e.g. *Windows NT* running on an *Intel* processor) even approach the *de facto* standard, the conditions become downright favourable should a worm capable of penetrating such a system be developed.

The logic behind this is clear – for a worm to be 'successful' it must be able to spread to new hosts faster than its replicating segments are discovered and removed from compromised hosts. To obtain such a foothold requires a reasonable number of machines that can be compromised in a limited time. The simplest way is to take advantage of a huge system of homogeneous networked computers. If any one platform/OS combination began to dominate the network (a position we are rapidly approaching), this task becomes much easier for the prospective worm.

The conditions now are conducive to worm development. If you are not sure, look around. Have you seen any such homogeneous networked systems lately? Ask yourself 'Am I ready to defend my system from such an attack? Would I even recognize it if it crawled into my network?' In the case of Admw0rm, you can easily defend against an attack with the methods discussed here.

References:

1. Brunner, John. *The Shockwave Rider.* Ballantine Del Rey 0-345-32431-5.
2. *The Worm Programs – Early Experience with a Distributed Computation.* Communications of the ACM 25: 1982, pp.172-180.
3. Kephart, Jeffrey O. *A Biologically Inspired Immune System for Computers.* Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, Rodney A. Brooks and Patty Maes, ed., MIT Press, Cambridge, MA., 1994, pp.130-139.
4. CERT Advisory CA-98.05 *Inverse Query Buffer Overrun in BIND 4.9 and BIND 8.*
5. CERT Advisory CA-98.05 *Denial-of-Service Vulnerabilities in BIND 4.9 and BIND 8*.
6. Gordon, Sarah and Chess, David. *Where There's Smoke, There's Mirrors: Tracking Trojans on the Internet.* Proceedings of the Virus Bulletin Conference, 1998. [Pre-print.]

FEATURE

Flashpoint

The appearance of Win95/CIH (see p.8) raises several concerns for users (and in some cases, developers) of anti-virus software. Although this article focuses specifically on CIH, many of the considerations in the first part are relevant to the more general question of how to deal with an outbreak of a new virus in a Win32 environment.

The virus is indisputably in the wild. In fact, since first being reported less than two months ago, one variant made it onto the July WildList. Several anti-virus developers that *Virus Bulletin* staff are in regular contact with have confirmed receiving samples of at least two variants from customers and most have received field samples of at least one variant. Across those contacts and from samples sent to *VB*, we have confirmed field reports of all four currently known variants that have the BIOS-flashing payload.

If You Discover an Infection…

It is, perhaps, comforting that as this is being written, most infections reported to *VB* are of the 1003-byte variant (also referred to as 'v1.2'), despite the 1019-byte variant being on the WildList. The former's payload triggers on 26 April, so the damage it can cause is somewhat distant. If you find your organization is already riddled with this variant, you are 'safe' in the sense that further damage (beyond the initial infestation of your site) is some way off.

If you find you are infected with a CIH variant, as always, the advice on first discovering a virus is 'Don't panic!'. In this case, it is important to determine the exact variant so the relative urgency of its next trigger date can be judged. This is slightly complicated at present, as few scanners accurately identify the different variants. Also, the plethora of names which different developers have used does not ease cross-referencing of reports.

Fortunately, the trigger dates for three of the four known variants have passed for 1998, leaving the 1019-byte ('v1.4') variant, with its 26th of any month trigger, to be the greatest cause for concern. As noted above, it is widespread enough to have made the WildList, but is not commonly reported to *Virus Bulletin*.

Should you discover an infection of CIH, it is also important to determine, quickly but *safely*, how widely the infection has spread through your organization. Safely? Remember, this is a fast infector – the act of *opening* a clean file on a machine with an active infection will cause it to be infected if you have write-access to the file. When planning a network-wide hunt, it is most important to consider files on network shares very carefully. From an infected *Windows 9x* machine, CIH 'sees' files on attached network shares as if they were local. [*That is, after all, the point of networking! Ed.*] Thus, you should not scan remote drives from a *Windows 9x* machine unless you are sure you are scanning from a clean environment. If scanning from an infected *Windows 9x* machine, the OS of the 'server' is irrelevant – an infected workstation will infect files its user has write access to, regardless of file server platform.

However, scanning from a clean machine does not mean that you can ignore the OS on the 'server'. *Microsoft* has made networking pervasively easy under Win32. Its success means you have to be sure not to scan a network share exported from a *Windows 9x* machine unless you are sure that machine is clean. Remember that 'remote' files as far as your clean scanning machine is concerned, are 'local' to a copy of CIH resident on the remote machine. All this talk of network shares being the source of possible grief may seem quite gloomy, but the default rights on *Microsoft Networking* shares are 'full access', and far too many people never change them…

Troublesome Resident

Normally, anti-virus software goes out of its way to detect resident viruses in memory, before it opens any files. This is precisely to prevent the spread of fast infectors across your hard drive. Unfortunately, memory scanning in the Win32 environment is far less straightforward than it is in DOS. In fact, a poorly written scanning procedure could page your machine to death! To date, few products have any effective memory scanning under *Windows 95* or *NT*, with each OS presenting its own, separate, implementation headaches to anti-virus developers.

Do not be misled by the 'Scanning memory' message many Win32 scanners display at startup. They typically scan the master VMM and/or their own image. This should detect pre-GUI and scanner infections, but little else. As CIH can only go resident under *Windows 9x*, where you probably do not have reliable memory scanning, check these machines with a DOS scanner *in DOS-only mode*.

That is *not* the same as running a scanner at a DOS Prompt. To be sure, select 'Restart the computer in MS-DOS mode' from the shut down menu. If starting from power-up, press F8 when the 'Starting Windows…' message first displays and select 'Command prompt only' from the menu. Alternatively, you may boot from your own, or an anti-virus developer's emergency boot disk. If using your own boot disk, ensure it is from an appropriate version of *Windows 9x* to avoid possible FAT32 problems.

If you have *NT* servers or workstations, it is safe to check them with native *NT* scanners. CIH cannot go resident under *NT*, so cannot infect nor trigger its payload on such

systems. Thus, few of the concerns that must be considered when testing a *Windows 9x* machine apply. However, note that you should not scan a network share exported from a *Windows 9x* machine – in that case, an active infection on the machine exporting the share can spread further, as described above. In a similar vein, to check servers running non-Win32 operating systems, either run updated scanners native to that OS, or scan them across the network from something other than a *Windows 9x* workstation.

Flash – Saviour of the Universe?

Flash ROM technology has been used in computers for some time. In the last few years, it has taken over the PC BIOS market. This partly reflects the move to Plug and Play, with its requirement for retaining current device configuration information, while still being able to update that information from time to time (usually when devices are added to or removed from the system).

Apart from the 'benefits' of Plug and Play (still widely referred to as 'Plug and Pray' by those who actually use it), a side-effect of PC BIOSes being stored in Flash ROM is that 'feature upgrades' and bug-fixes can be deployed easily. The cynical would say that this has encouraged, or at least done nothing to discourage, an attitude of 'if this release is not right, we can fix it with an update' – certainly an approach not readily available to BIOS developers and system designers of yore.

The very cynical would say that this ability to 'fix the firmware' has allowed the standards-setters more room to change the 'standard', reducing any pressure there may have been to 'do it once, do it right'. In concert with a software development culture where such fundamental OS features as system security are shipped as 'patches' and 'hot-fixes', this may be less surprising than it is disturbing.

The typical counter to this is that, given errors are inevitable and that everything is getting so much bigger and more complex, the number of things that 'slip through' must increase. That is a worrying defence. As computer systems come to play ever more crucial roles in our lives, it seems reasonable to expect (even demand) that they become more reliable, more dependable, not less so.

Is it really that 'bad'? Look at a few PC vendor web sites. Most carry recommendations to download their latest BIOS updates and install them before installing *Windows 98* and it seems likely the same will happen again when *NT v5.0* ships. The PC world appears saddled with upgrade mania driven by 'ship it or bust' developers.

Protection Measures

Last month *VB* advised readers that some mainboards had jumpers or switches that could prevent the Flash ROM holding the BIOS from being written to.This may have produced some undue optimism. Having looked more closely into current trends in Flash ROM and mainboard design, it seems most vendors see lower voltages, fewer jumpers and more soldered-on components as the way forward. Combined, these directions also lead to lowered costs, and in light of CIH's payload, seriously reduced security for the purchaser.

The voltages at which Flash ROM can be written has fallen to 5 V and lower ('2.7 V flashable' has been noted on some parts from some manufacturers). Thus, from a chip design viewpoint, the need for a separate, dedicated write-enable voltage has disappeared. Many parts with 5 V or lower write circuitry have hardwired the write-enable voltage from their standard logic circuitry supply rail. This means that the mainboard designer *cannot* provide a point at which the owner of the board can disable all flash write operations *if* the BIOS is on such a Flash ROM.

Given CIH's payload, this last option probably seems crucial to most readers. Not having the option to disable flash write operations means the mainboard will always be open to attacks similar to that in CIH. A chilling prospect!

Mainboards that claim, or appear, to have a flash disable feature can easily be tested. Locate a Flash BIOS *update* from your vendor or BIOS manufacturer. Most 're-flasher' programs have an option to save the current BIOS image to a file – use that to backup the BIOS. Now set the 'protection' feature, start the machine and flash the update. If the flash operation seems to proceed, reboot and make another backup of the BIOS. Compare this file to the earlier one. If there are no differences, the protection feature works.

However…

If the Flash BIOS is able to be protected (most will not be), enabling that protection may not be a good idea. As mentioned above, Plug and Play requires a writable but non-volatile data storage area to ease its configuration burden at each startup (the ESCD that is so cryptically noted as having been updated on some machines at startup). This area is not, as far as *VB* has been able to tell, located at a fixed memory address.

That gives BIOS writers flexibility in situating the ESCD conveniently. It also complicates the implementation of having a small area of writable Flash ROM, and a larger area that can be separately write-disabled. A two-level writable flash option such as this, requiring some deliberate external action to enable flashing of the BIOS part of the ROM, would seem to be a desirable middle-ground approach compared to typical, current designs.

All is probably not quite as black as this may sound. Some mainboard designers have implemented a BIOS recovery option, in which a very small part of BIOS code is stored in non-flashable ROM. Setting a jumper or switch to recovery mode activates that code at boot-up and a bare minimum configuration will bring the machine up to a point where the ROM can be flashed as if for a normal BIOS upgrade. Fortunate CIH victims will have such a mainboard.

TUTORIAL

Free Macro Anti-virus Techniques Part 3

Jimmy Kuo
Network Associates Inc

[*In this instalment of his self-help series, Jimmy continues with more options for protecting your* Word *environment. Remember that unless otherwise stated, file locations are the* Word 95 *defaults and may vary for* Word 6 *and/or* Word 97 *users. Ed.*]

Word Viewer

To recap, the purpose of this series is to present topics which will help prevent or reduce macro virus infections. The title of this section is Word Viewer because, if a program other than *Word* is used to read DOC files, one which does not support macros, macro viruses are less capable of spreading. *WordView* or WordPad are such programs. WordPad does not support macros, while *WordView* has restricted support.

One of the commonest viral infection methods is via an emailed document. Double-clicking on the document causes *Word* both to open automatically and, in the absence of other protections, to become infected. This is governed by one of two setups – either the email itself program is programmed to activate *Word*, based on whatever criteria it uses, or it makes use of the registry.

The steps necessary to change the default association of DOC and DOT files to something other than *Word* are covered in this section. These changes affect such activities as double-clicking, drag-and-drop and various things in Explorer or File Manager. These procedures involve modifying the *Windows* Registry. Please consider *Microsoft's* warning in Fig. 1 (from the *Windows 95* help) before performing these activities.

Fig. 1 *Windows 3.1x* users should also heed this warning, but note that recovery details are not provided in your online help.

Locate WORDPAD.EXE, WORDVIEW.EXE or whichever alternate viewer you plan to use. Note the full pathname for the program. Find REGEDIT.EXE or REGEDT32.EXE and run it. Under *Windows 3.1x* you must launch RegEdit from the File, Run menu in File Manager with the '/v' switch.

Under *Windows 9x* or *NT*, locate the .DOC file type entry in HKEY_CLASSES_ROOT. Look for the highest numbered Word.Document type listed. Now traverse the structure to locate an entry matching that Word.Document type. Expand the tree to find the Shell, Open, Command key (Fig. 2).

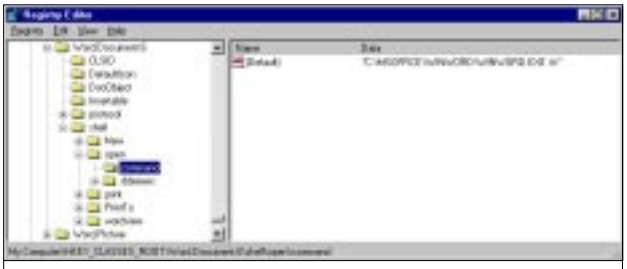


Fig. 2 Locating the matching commands to change.

Double-click on the 'Default' value in the righthand panel and change the string to the command (full pathname plus any required switches) of your chosen alternate viewer.

Users of *Windows 3.1x* (Fig. 3) should search for the Word.Document.6 key and change the command as above.

Fig. 3 Things look slightly different in *Windows 3.1x*.

Note that regardless of the platform or version of *Office*, all the commands have to be changed – Open, New, and so on.

Pro: Does not support macros.

Con: WordPad and *WordView* are just not *Word*.
Some email programs disregard the registry!
Avoids macros, but does not tell you they are there.

[Microsoft's *own free viewers can be retrieved, from http://www.microsoft.com/office/office/viewers.asp. Ed.*]

Replace NORMAL.DOT Every Time

In normal DOS usage, there is a regular suggestion to clean boot before running anti-virus programs. While there is no need to enforce a clean *Word* environment in this situation, it is still worthwhile to know that your environment is clean before each new day's use. One way to do it is to delete and replace your NORMAL.DOT file every day. We use AUTOEXEC.BAT to force this each time the machine is

booted. Unfortunately, this does not make NORMAL.DOT a useful mechanism for holding changes. If changes are not moved to the archived copy, they are lost. Furthermore, this method does not inform the user of an infection. The infection is simply destroyed. Since that is the normal course, I have added a read-only attribute to the file.

To set this up, do the following:

```
cd \msoffice\templates
md archive
copy normal.dot archive\normal.goo
```

(Goo is short for 'good' and presumably will not conflict with any other extensions in use.) Now insert this at an appropriate point in your AUTOEXEC.BAT:

```
attrib +r archive\normal.goo
Add the following to AUTOEXEC.BAT:
pushd \msoffice\templates
erase /f normal.dot
copy archive\normal.goo normal.dot
attrib +r normal.dot
popd
endbat
```

Pushd/Popd utilities save the use of full pathnames. [*A subsequent installment will explain ENDBAT.BAT. Ed.*]

Pro: Boots *Word* clean every day.

Con: Hassle to change any macros.

Does not tell you if anything happened.

Check for Changes to NORMAL.DOT

Another way to ensure a clean *Word* startup, instead of forcing a new NORMAL.DOT each day, is to check the current version against the known clean one which has been archived. This method differs from the previous one in its ability to alert you to changes. To set this up:

```
cd \msoffice\templates
md archive
copy normal.dot archive\normal.goo
attrib +r archive\normal.goo
```

Then add the following to AUTOEXEC.BAT:

```
diff \msoffice\templates\archive\normal.goo
    \msoffice\templates\normal.dot > NUL
if errorlevel 1 goto :changed
:: you may have other code here
goto :end
:changed
echo normal.dot changed^G
pause
:end
endbat
```

Diff is similar to DOS' FC, but returns the errorlevel necessary for our use. Ask the local guru for a copy and ensure it handles binary files! The command 'diff… >nul' must be on one line. '^G' represents Ctrl-G – it causes a beep. It can be tricky to create, but Ctrl-P, Ctrl-G in DOS EDIT works.

Pro: Informs of any change.

Ensures clean NORMAL.DOT each day.

Generally silent, no problems.

Con: No notification until next bootup.

Requires an expert to set up.

Requires an expert to understand its use.

Check the Startup Directory

After creating NOAUTO.DOT (see *VB*, April 1998, p.9), it was suggested that the file be placed in *Word's* Startup directory. Any template file stored there is automatically installed into *Word's* environment when it starts up. That also makes this directory the target of virus attacks by placing infected template files there. Thus, it is important to monitor this directory to make sure the contents do not change. Several viruses already use this 'trick'.

To ensure no viruses are added, you need a listing of the directory in a known, clean state. Then, each time the machine is started, check the current contents do not differ from the list. The code needed for this is presented below.

But first, start *Word* and locate the default Startup directory. Click on Tools and go to Options. Choose the File Locations tab, then look for the entry related to Startup. Note this entry and use it wherever '\msoffice\winword\startup' appears in the following instuctions. In DOS, continue by entering the following command (all on one line):

```
dir /b/o/a \msoffice\Winword\Startup >
    %TEMP%\wws.lst
```

This creates the list of the current contents of the Startup directory, storing it in your defined 'temporary' directory. Note that versions of MS-DOS prior to v5.0 do not support any of the '/b/o/a' parameters to DIR. Also, only *NT* and *Windows 9x* command prompts support '%TEMP%' substitution, but that works in batch files for all relevant versions of DOS. Add the following to AUTOEXEC.BAT:

```
dir /b/o/a \msoffice\Winword\Startup >
    %TEMP%\wws.chk
diff %TEMP%\wws.lst %TEMP%\wws.chk > NUL
if errorlevel 1 goto :diff_startup
:: you may have other code here
goto end
:diff_startup
echo Word startup directory changed^G
pause
:end
endbat
```

Pro: Informs of any change.

Ensures clean boot up each day.

Generally silent, no problems.

Con: No warning until next boot up.

Requires an expert to set up.

Requires an expert to understand its use.

Acknowledgments:

Vesselin Bontchev, *FRISK Software International*; Ray Glath, *RG Software Systems*; and Stefan Geisenheiner, Jivko Koltchev, Akihiko Muranaka and Francois Paget, *Network Associates*.

OPINION

Testing Times

Peter Morley
Dr Solomon's Software

[*Readers responses to this article are welcomed. Ed.*]

The February 1998 issue of *Virus Bulletin* included the annual review of DOS-based anti-virus products. For only the second time in history, a product – *Dr Solomon's Antivirus Toolkit* – detected 100% of all the test viruses. While this will obviously provide a useful commercial boost, I find it all quite amusing, particularly as the same basic detection capability goes into all products. Some comments are in order.

Since the phenomenon has only occurred once before, it follows that the testing procedures and/or specimens were changed immediately after that occasion. Such changes must be imminent again. If not, *Dr Solomon's* might continue to get 100% awards, which would annoy the other anti-virus vendors and ultimately bore *VB* readers.

The version of *AVTK* which received the award was 7.77, which was out of my hands in mid-September 1997, and into the market by late October 1997. Since then, there have been nine more releases – 7.78 to 7.86 inclusive – and by the time you read this, at least eight of them will be in the market. Each has added over 500 viruses, variants and Trojans to the detection capability. So, since scoring 100%, the product can handle an additional 4000 viruses.

These figures have at least two major implications; they cast doubt on the traditional three-monthly update philosophy, and they make it imperative that we make weekly updates available on the Web. We have done exactly that.

Incidentally, I do not believe that the figure of 500 new viruses per month will last, and said as much a few months ago. As often happens when predicting the future, I was wrong. I still do not believe it will last, and I think 300-400 will become more normal. If this does not happen, we still intend to process every virus that comes our way!

Testing Procedures

Virus Bulletin, along with most other testers, concentrates on detection capability, and adds useful information on scan speeds, false alarms, and user friendliness/administrative capability. Most of them duck unashamedly when it comes to testing repair capability. The reason is obvious – repair testing is difficult and time-consuming. The above testers used to point out that it is better to deal with file viruses by replacing the original executables than by repairing them. One can get away with this, if one avoids talking to IT managers who have had major server infections.

Now, however, macro viruses are with us, and replacement by the original is no longer an option. Repair is essential. So, the only viable method of ducking is to pretend the problem does not exist, and look the other way!

Suggestions

These are my recommendations for testing procedures. Always publish the list of what is being tested. *Virus Bulletin* is impeccable in this respect, but I have seen reviews which made no mention of it. I always regard such reviews as worthless, as they are probably based on a tiny sample, several years behind the times. You should, too.

Reduce the detection testing workload by removing the viruses everyone detects from the ones to be used next time. I believe readers of the reviews have no interest in what everyone detects. (Excluding *MSAV*, for obvious reasons.)

While the payoff is obvious, there are two snags to this proposal, one trivial, and one more serious.

The former is that if a product ceases to detect what was detected before, subsequent reviews will miss it. It does happen. Back in early 1993, a release of *FindVirus* failed to detect Cascade. All hell broke loose! The latter point is that results will show poor performers in a worse light than at present, and their developers may get stroppy about it. The reviewer must be prepared to provide samples of what has been missed, and it must not be arguable that the samples are corrupt, or that they should not be detected at all because they do not replicate properly. It may be necessary to implement this slowly, rather than suddenly.

Repair testing should be introduced, even if it is only repair testing of macro viruses. We are approaching the transition between most vendors dealing satisfactorily with most macro viruses, to the time when some will start to fall further and further behind. When this happens, readers of reviews ought to know it.

My final suggestion concerns updating the test samples, by adding new viruses. Do *not* have a sudden panic, followed by a big update. How about this? Invite each vendor submitting a product for review, to provide up to twelve virus samples for inclusion in the evaluation suite. They should say where they came from, particularly if they were received from the field. Make it clear that if the samples include any which are corrupt, *all* samples submitted will be excluded. Say, in the review, who provided what.

Use these samples in the evaluation suite after three months, so the 'We've never seen it' vendors do not have a leg to stand on. Be prepared to send specimens on request, to those vendors who are smarting over their failure to detect, or repair, when everyone else has done so.

PRODUCT REVIEW 1

Avast32 for Windows NT

A long established product which has seen many changes in the last year, *Avast32* is the 32-bit version of *AVAST!* by *Alwil Software*. As with *AVP* (see *VB*, May 1998, p.25) it is probably better known in Eastern Europe. Is this another undermarketed, quality product, or is its lack of exposure a good thing for users?

The Package

Avast32 is supplied on a CD, packaged in a gatefold cover including a small instruction pamphlet. Production standards are high, and the artwork used is one of *Alwil's* series of distinctive designs. The documentation in the pamphlet is not encyclopædic, covering the basics of installation for both *Avast32* and its DOS and *Windows 3.x* incarnation, *AVAST!*. Separate sections cover troubleshooting, contact addresses, upgrades and virus definition documentation. The last is necessary – one could not consider the 16-page pamphlet a major source of product information.

The CD contains both *AVAST!* and *Avast32*, the duplication of versions being perhaps the reason for the lack of autorun. The root of the CD contains one batch file for the setup and another for the upgrade of the program, a couple of associated executables, plus readme files. These are provided in French, Czech, German and English. They reiterate the installation and upgrade instructions in more detail, and contain a list of known bugs and their fixes, upgrades between versions and a mini-FAQ. The bug list is extensive but the almost universal solution is an upgrade to the current version of the software in which a fix has been applied. The readme also includes installation instructions for *Adobe Acrobat*, which facilitates the reading of the documentation on CD.

The other contents are neatly divided amongst folders. *AVAST!* and *Avast32* are worthy of a separate folder, as are *Acrobat* and the documentation. The virus definitions, different for the two versions of *Avast*, are included with

instructions on the use of PGP. This allows the updates to be tested for both authenticity and file corruption – a welcome feature, if somewhat less automated than in some other products. Finally, image files, allowing the production of 3.5-inch installation disk sets are provided.

Documentation

Electronically, the readme file includes the same variety of quickstart information, and what amount to addenda and corrigenda to the main manual. This manual is supplied in *Acrobat* PDF format, in the languages already mentioned, and stretches to some 140 pages. In an inspired move, the documentation is designed to be clearly legible a page at a time within *Acrobat*, at 800x600 resolution or better. This prevents the headache caused by larger format, multi-column A4 PDF documentation, where shuffling back and forth makes referring to the document awkward.

The documentation itself is copious and complete, though not without a certain whimsical charm of its own. This comes from a combination of slightly imperfect translation and an author who, clearly, has included his personality in the manual. Particular favourites include 'You had better be pessimistic, it will pay' and the very good advice 'If you are the type of character who is easily panicked, leave the computer, if you find a virus, and drink a cup of coffee'.

There are full hyperlinks within the manual, and an index, referred to, somewhat cryptically, as the register. Installation, setup, reaction to infection and, at most length, configuration are covered fully, yet the quirkiness of some of the text never detracts from its comprehensibility. This manual also acts as the general help file for the program.

Installation and Upgrades

Installation of *Avast32* is simple, with a batch file provided rather than the standard *Windows* setup program. This is because the batch file is also used in the installation of *AVAST!*, which needs to be installable in a DOS-only environment. Setup choices consist entirely of the language to be selected. After installation a reboot was needed, resulting in a folder of programs, and the addition to the task bar of a button to the on-access component.

Upgrades to the main program are also supported during the installation process, though the option is, understandably, only available for an existent version of *Avast32*. To check this feature an upgrade was performed from the November 1997 version of *Avast32*. Perhaps unsurprisingly, the behaviour blocker declared that the update procedure was performing some suspicious tasks. When these actions were approved, however, all progressed smoothly.

Virus signature updates are handled in either of two ways. Self-installing definition files are provided or, alternatively, an updating program on the CD may be used. Both per-

formed speedily and without problems when *Avast32* was installed in its default directory. The update files themselves are downloaded from *Alwil's* web site, and an update subscription service is available at extra cost.

The Program

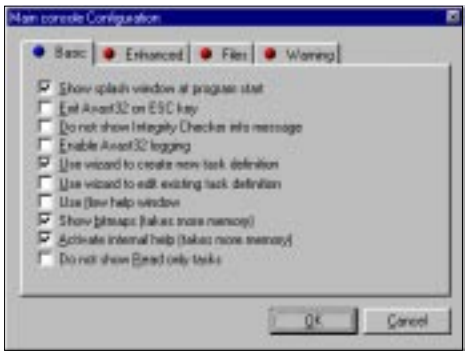
The on-access and on-demand components of *Avast32* are controlled for the most part from the same interface. This is available in two incarnations, toggled between the simple and enhanced styles. The documentation recommends the simple configuration to the typical user, with the enhanced reserved for administrators or more expert users. For the purposes of testing, the enhanced interface was used, as it gives both more information and more control.

Avast32 also adds the option of scanning a file to the context menu in Explorer. Minimal control is also available, over the on-access component, from the task bar. This may be double-clicked to bring up a status dialog box giving details of which on-access protection components are running. This box is something of an oddity – it contains a miniscule window for viewing these options, forcing scrolling, where a larger box would seem appropriate. It also offers little in the way of control, 'on' and 'off' being the options available here.

The main user interface, however, cannot be accused of lacking control options or visibility. Control over the actions within the program is from drop down menus and tabs. The drop down menus are divided amongst View, Options and Help while the tabbed options are for Tasks, Results, Viruses and Help. The two Help menus contain entirely different options. The major difference between the simple and enhanced user interface is the absence in the former of all tabbed options save those for Tasks.

The Tasks tab is, not surprisingly, the main area where scans and checks are performed. In *Avast32* terminology, a scan is a test for viruses using patterns, and a check an integrity test using checksum files. Checksum files are stored locally rather than centrally. Scanning and checking may be performed either separately or concurrently. A selection of standard tasks cover the investigation of local floppy and hard drives, both completely or with the option for interactive folder selection. It is also possible to create new tasks from scratch, or edit those already present. The options available in this area are many and varied.

At the simplest level, there are two variables in creating a new task or modifying an existing one; whether the Wizard is used and whether the task will be available for all users or merely the creator. The latter is signified by colour coding within the Tasks listing, the supplied tasks being considered yet another subdivision. The manual suggests the use of the Wizard to all but experts, a recommendation with which this reviewer heartily agrees. The reason for this is the sheer number of parameters *Avast32* allows to be altered within each task – a selection quite possibly the largest in any recent offering to *VB*. Other programs allow the adjustment of as many parameters globally, but *Avast32* allows these to be set for each individual task.



The more interesting options include setting tasks to trigger on either start up of *Windows*, the *Avast32* interface or another task and closing the interface after tasks have been completed. Combinations of these parameters can be set so as to initiate start up scans without the need to exit manually. It is also here that task priority may be selected. The lowest allows scanning in the background without large overheads, whilst the highest is handy for reviewers who value their sanity.

Notification can be configured across a network or individually – the supplied warning including a spoken WAV message to the effect that an infection has been detected. Messaging for notification is fully customized to the text involved, the amount of information provided and the domains or users to whom notifications are provided. Logging is supported, though with somewhat fewer options than might be expected, with only log file name, size and location configurable options.

These variables are added to in the Options menu, where global parameters are set. Matters here are divided amongst Common parameters, Resident protection, the Main Control console and Command Line scanner. Options range from the purely cosmetic (whether splash screens are shown on startup), through the control of the Wizard function to the setting of passwords for disabling resident protection.

The ability to use a central network repository is a recent addition to the program and allows for updates to be performed from this single location. With the complexity of the interface and password options, this provides areas where normal and administrative users are differentiated. This is still something of a weak spot for *Avast32*, since there are no specialist administrative tools provided and no dedicated server versions of the product.

If either creation of a log file or system logging is selected within a task, results are viewable from the Results tab once the task completes. This has an Explorer-like view of the computer and network environment. When a folder which has been the subject of a recent task is selected, information concerning detections and checksums is displayed. This includes clean results, as well as infection and checksum failure information.

The Viruses tab provides scant but useful details upon a large selection of viruses, though it would appear that not all viruses detected by *Avast32* are included in this listing.

Details include whether a virus is known to be in the wild, handy for checking against possible false positives, and whether the virus is considered particularly dangerous. This last classification seems reserved for viruses such as One_Half, where removal can be a bad thing if performed in too simplistic a fashion. The Help tab allows access to the PDF format help files, though the *Avast32* interface combined with the *Acrobat Reader* makes for a less easy read than when used standalone. The View drop down menu provides the toggle for the Simple and Enhanced interface, while the Help menu gives a direct link to the *Alwil* web site.

WWW Presence

Since the *Alwil* web site is used as the traditional method of obtaining definition upgrades and may be accessed directly from the program, it seemed appropriate to investigate the area. The website is available, as with the documentation, in both Czech and English.

The layout is uncluttered and locating and downloading a new set of virus definitions was a simple task. This update was provided in a ZIP file which could, when unzipped, be installed from the Options menu. Download time was five minutes using a 28800 bps dial-up connection and access to the site was not subject to significant delays.

Speed and False Alarms

The on-access component of *Avast32*, despite having two sub-components – the behaviour blocker and the resident guard – only exists in two states, 'on' or 'off'. The behaviour blocker triggered without fail when certain EXE and COM files were copied as part of the overhead testing. This problem was overcome by selecting to ignore this write operation for the purposes of behaviour blocking – a choice only available after the operation has been begun. This also highlighted the fact that the Ignore option was applicable to the entirety of an XCOPY command – ignoring the first copy effectively passes the rest of the copies as approved. The operation of the overhead scanner under these test conditions added a 30% overhead to the copying of files which comprised this test.

In the Clean test-set (520 MB of COM and EXE files), one file caused a false alarm in an on-demand scan. This file is part of an archaic anti-virus suite and was declared to be home to the Tequila virus. The test itself was in the 'arthritic sloth' speed category – a result of the program running the scanner thread at a very low priority. The program also includes checksumming techniques to spot alterations of files, but this was not tested.

Scanning of floppies proved leisurely too. A clean diskette containing 43 files was scanned in 65 seconds, whereas a disk containing the same files, though each infected with Natas.4744, was scanned in 100 seconds. This is a considerable increase on a reasonably slow baseline, and hints that the *Avast32* engine is working very hard to determine whether files are infected. (Speed tests were performed on a standalone machine, which remains constant in specification throughout reviews. The results here are thus directly comparable with other recent *VB* tests using *NT4 SP3*.)

Detection

Avast32's on-demand scanning component proved admirably efficient – none of the samples in the *Virus Bulletin* test-set based on the April WildList were missed. The same flawless record was achieved in the on-demand boot sector tests. Slight hiccups were caused by using the Return key, which should simply initiate a pause in scanning. Starting a scan, pausing and resuming resulted in no detection at all. On the other hand, a subsequent scan appeared to be more speedy and detected the virus in question.

As with many *NT* products, boot sector problems were apparent in the on-access tests. Although the on-demand scanner was able to scan 'odd' format disks while detecting their nature, the on-access scanner went no further than stating that the disk was strange. A likely argument is that an inaccessible disk is a safe disk, yet with boot sector infectors this is not as certain as it is with file viruses.

This problem was also complicated by poor disk change detection, another old favourite under *Win32*. It was impossible to detect a boot sector infection twice on the same disk, if that disk was scanned twice in a row. Interpolating another test sample usually resulted in detection on each disk, but not universally.

Conclusion

A product which cannot be faulted on detection, *Avast32* has continued a record of improvement not only in that area but also in the number of options available. Two weaknesses are to be seen, though neither impacts severely upon the usability of the program as a whole. The documentation is entirely usable and complete despite its idiosyncrasies, and the lack of industrial-strength administration tools may be a problem in large networks. Overall, a product which is unlikely to disappoint any but the technophobe – who might be scared off by the level of control available.

Technical Details

Product: *Avast32* for *NT*.

Developer/Vendors: *Alwil Software*, Prubezná 76, 100 00 Praha 10, Czech Republic, Tel +420 2 782 2547, fax +420 2 781 0548, email sales@alwil.cz, WWW http://www.anet.cz/alwil/.

Availability: *Windows NT v3.51* (or higher) with 10 MB of free disk space.

Version Evaluated: 2.0-722, June 1998.

Price: For 1–10 PCs $39/machine, with a sliding scale to 2501–3000 PCs $5/machine.

Hardware Used: 166MHz Pentium-MMX workstation with 64 MB RAM, 4 GB hard disk, CD-ROM drive and 3.5-inch floppy drive running *Windows NT4 SP3*.

Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NW/199807/test_sets.html.

PRODUCT REVIEW 2

Norton AntiVirus for Windows 95

Norton AntiVirus (*NAV*) is *Symantec's* offering in the anti-virus field, and claims to be not only global in scope, but in current advertising materials 'the only anti-virus products able to detect and remove all existing macro viruses'. Disregarding such bare-faced lies as this, the software was investigated in the usual *Virus Bulletin* fashion.

Confusion within *Symantec* about the submisison date for this review resulted in delays to the testing, allowing both the *VB* test-sets and the *NAV* virus definitions to be the most up-to-date possible. Thus, this review gives an even better picture than usual of how the product is performing this month. Of particular interest may be detection of flavour-of-the-month viruses Win95/CIH and Win95/Marburg. Good, bad or indifferent?

The Package

Norton AntiVirus is supplied in an apparently bullet-proof box, the lurid yellowness of which is tempered somewhat by the quizzical smile of the eponymous Peter. Opening the box is like reaching into a lucky dip – a huge collection of tat is included, together with the things you really want. Included is an advertisement and offer for a US ISP, a 'tell a friend' reward plan enabling customers to obtain free propellor-tipped caps, a registration card and more advertisements for other *Symantec* products. The manual, CD and emergency boot disk are hidden beneath this pile.

The CD and Installation

The *NAV* CD is also a regular box of tricks, some of which are unique in the anti-virus field. Many products now operate on autorun but *NAV* appears to be the only known one in which a lengthy animated graphic greets the user (and, irritatingly, one which cannot be shortened in any obvious way). *Symantec* informs us that this is a feature of the Deluxe retail product and not present in the package offered to corporate customers.

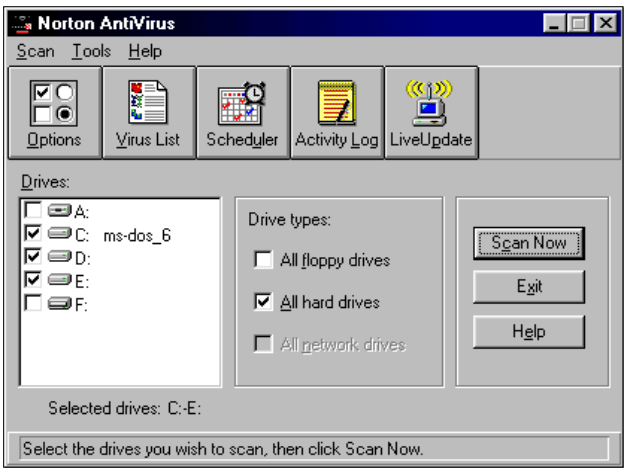
The graphics are presumably designed to appeal to someone who, raised upon all-singing, all-dancing games packages, will not settle for an ugly application. The graphical splendor that is *NAV* certainly does not disappoint on this front – as part of the opening menu there is the option to view a number of videos. These consist of short features upon viruses in general, *NAV* in particular, suggested reactions to a virus incident and a tour of the *Symantec Antivirus Research Centre* (*SARC*). The level of information provided is unlikely to tax any viewer.

Also included on the CD are evaluation copies of a plethora of *Symantec* products, and an ISP setup package, including *Internet Explorer*. The mysteriously undocumented *Norton Safe on the Web v1.0* also makes an appearance here, though it was not tested.

Selecting the installation option for *NAV* brings yet more choices. These were whether to install the *Windows 95* or *NT* version of the program, or *Adobe Acrobat* to allow reading of the PDF version of the user's guide. Selecting the *Windows 95* version, installation commenced with a standard *InstallShield* routine. The licence agreement was followed by options to schedule weekly scans, the on-access part of *NAV*, or to perform a scan at start-up. All of these were selected in the default settings of the installation. These choices having been made, file installation was a speedy process, finishing with a reminder of the web addresses for the purchase of support options and for *SARC*, together with the telephone number for virus enquiries.

This seemed a likely place for the installation to terminate, but *NAV* still had some tricks up its sleeve. Next on the list was a reminder to run LiveUpdate upon the completion of installation, a little odd considering the following. The choice was offered as to whether to run that self-same LiveUpdate, whether to create a rescue disk set and whether to scan upon installation – with the default settings again being yes to all. Last in the installation process came the option to integrate *NAV* with the default web browser on the test machine. Here it was possible to select from a lengthy collection of file and download types to scan, though *Microsoft Access* (MDB) files were notable by their absence. A reboot of the system completed the process.

Installation resulted in the addition of a scan task to the Explorer right-click menu, the *NAV* Auto-Protect icon in the system tray, and a program group off the Start/Programs menu. This last contains the main program, scheduler and uninstaller for *NAV*. Also present is an option to create



rescue disks and, if not selected during installation, the option to integrate *NAV* with installed browsers. The on-line Virus Encyclopædia may also be reached from here. A further sub-folder contains help files for connecting to the web and on-line support, together with links to the *NAV* support and *SARC* web sites.

LiveUpdate and Web Services

LiveUpdate is a much vaunted ability of the *NAV* program (indeed, of a whole range of *Symantec* products) and thus worthy of testing, but *NAV* proved to be too paranoid to allow any part of the program to activate on a machine with a non-standard Winsock installed. This problem seems not unexpected with LiveUpdate alone, but remarkably over-cautious in its aborting the entire program when such a problem was encountered.

Given this failure, the link to the *Symantec* support site was followed in an attempt to clarify the problem. The attempt proved fruitless though interesting. It provided a number of services including options to 'chat one-on-one' with a *Symantec* technician, subscribe to the *NAV* news bulletin or use the newsgroup-like *NAV* discussion groups. The technician option, however, proved only to be available in the US, and cost some $20 a shot, thus proving untestable due to the tester's locality.

The *SARC* site, on the other hand, provided information on such things as the CIH virus and other recent notable, and not so notable, viruses. It seemed a little odd to see the altogether mundane W97M/ZMK.J, or WorldCup, virus mentioned as important – an inclusion perhaps triggered more in reaction to other organizations' publicity machines than any real interest in the creature.

The on-line Virus Encylopædia was also available from the *SARC* page. Presented in both English and Korean, this gave the reviewer a chance to catch up on his rusty language skills. The virus descriptions themselves are vast in number, though this looks to be due to much of the text for tedious or rare viruses having been computer generated from a database.

Documentation

The manual supplied with *NAV* is an impressively chunky piece of literature, though this is something of an illusion since the instructions refer to all the *Windows* and DOS incarnations of the *NAV* range. The *Windows 95* documentation alone comes to a total of 32 pages, covering installation, use, updating virus signatures, dealing with virus infections and troubleshooting. The PDF version supplied upon the CD is only slightly different, in a purely cosmetic fashion, though it does benefit from hyperlinking.

Within the program, the more usual *Windows*-style help is available. This is a much more comprehensive and useful set of information than the manual. Context-sensitive help is also supplied, through the usual right-click operation, though of a nature best described as minimalist. The results, where tested, were universally a very brief description, together with a general link to the help function, though not a link related to the subject of the query. There are links, as a final part of the help process, repeated from those already mentioned, to the various *NAV* web sites.

The Program

The *NAV* program has the appearance of a rather smaller beast than might be expected. The main control panel boasts only five tools, available from either buttons or a drop down menu, to which are added the help function and a small section of scanning commands. The scanning commands cover the familiar choices of selecting files folders or drives for the attention of *NAV's* beady eye.

This diminutive face, however, is backed up by a larger set of options under the sub-menus Options, Virus List, Scheduler, Activity Log and LiveUpdate. LiveUpdate has been considered already and Virus List is much the same sort of information as provided by the tedious specimens at *SARC*, though here the characteristics of the viruses have not been processed into sentence form, which is, perhaps, a blessing. Activity Log is, as might be expected, a Log of *Norton AntiVirus* events.

This leaves the two more important selections, of which Options is the most complex. On-demand and on-access components may be configured here in a blanket fashion for each, together with inoculation. This last feature is not, as is the case with some products, a matter of appending checksums to files, but an external checksumming process to monitor files for changes. Checksums are, by default, only checked for boot records and system files. Exclusions from scanning and file types to scan may be configured here too, with, as yet, *Access* files being exempt from scanning in the default setting. Exempt files may be tagged by variety of scan rather than just overall, for example, WINWORD.EXE is flagged as not being checked for alteration of program files or unknown viruses.

Continuing the host of options are actions for *NAV* at startup and upon detection of viruses, the latter encompassing alerts and processes to be enacted upon the infected files. Lastly, logging is controlled from this area and passwords may be set for almost any changes which might be made to the *NAV* settings.

Last but not least comes the scheduler, also accessible more directly from the main *NAV* program group. The visibility, timing and scan types are all configurable, as is the appearance of the scheduler itself. Time intervals may be selected from a reasonably comprehensive list, though triggers, as for example a virus detection, are not supported.

Detection and Speed

For reviewers, a continuing problem with *NAV* is its insistence upon producing log files in a proprietary, binary format, rather than as text. A *Symantec*-supplied conversion utility wasted more time than it saved before failing, and thus several methods were used to determine detection ability. Deletion was the most effective of these, and provided some useful extra information. Among these was the discovery that 'deny access' options deny even *NAV* the ability to delete the file concerned. Boot virus detection on the other hand was performed as usual, log files being unnecessary for what is a relatively small set of samples.

On-demand and on-access scanning in *NAV* differ in that heuristics are, by default, disabled in the on-access scanner, while enabled in the on-demand. There are also two settings for heuristics, if enabled, in each case. The first of these detected the four samples of Omed.544 otherwise missed on-access in the Standard test-set, but no change in macro virus detection was noted regardless of this setting.

Polymorphic and, more importantly, In the Wild viruses were detected perfectly – a solid and reassuring result. When deletion was selected as a method of producing detection statistics three of the In the Wild file samples (Byway.A and .B, and DirII) were detected yet undeleted. These, however, are cluster infectors, which redirect file reads through a fixed disk cluster by alterating the file and directory information. Deletion of infected files in such cases tends to result in data loss through file system corruption – thus non-deletion is the correct choice.

The standard set proved trickier on account of Marburg, which was missed in all 18 samples presented to the scanner. This is somewhat alarming, since Marburg is a virus which has had the luck to become very widely distributed, yet has been known since late April. Three months really should be long enough to provide a good detection routine for a virus, even one which, as Marburg does, uses several odd (and new) techniques.

The macro virus set again showed some weaknesses, with two variants of both W97M/Class and A97M/AccessiV remaining undetected, as did W97M/Kitty.B. Class is a new form of *Word* macro virus, somewhat in the sense that XF/Paix was a new form of Excel virus a few months ago. AccessiV, on the other hand was undetected simply because MDB files are not in the standard list of scanned files. *NAV* users concerned about this virus would be well advised to alter this setting. Kitty's non-detection was surprising for 'the only anti-virus products able to detect and remove all existing macro viruses'.

Boot sector viruses, though all detected, did throw up some anomalies, though fewer than in many competing products. There was a degree of instability during disk changes in the on-access test, which managed to reach a state where soft reboots were impossible. This mode of detection also produced intermittent failures in the detection of disk changes. On-demand the scanner declared MISiS.A and Michelangelo.A to have been locked by a utility, but nevertheless detected the viruses without problem. The detection results are thus 100% for Overall In the Wild, 100% for Polymorphic, 96.7% for Macro viruses, and for the Standard set 97.8% on-access and 98.2% on-demand.

Scanning of the *Virus Bulletin* Clean test-set produced no false positives, and was performed in 340 seconds – a data throughput rate of 1.5 MB/s. Scanning clean and infected diskettes, took 30 seconds for the clean and 49 seconds for the infected disk. This quite large difference suggests the *NAV* scanner puts quite a degree of processing into identifying the virus.

The on-access scanner may be configured to test files when they are run (not tested), opened, copied or moved as one option, or created or downloaded as another. The last three, combined, incurred a 20% overhead on file copy operations. Tests of the constituent operations showed that most of this is attributable to the scan on file creation option.

Conclusion

A full-featured package, as expected, neither leaping to new heights nor plumbing new depths. The non-detection of the relatively new Class macro viruses hints at problems with the engine, or at least the much-heralded heuristics. Along with the inclusion of MDB files in the standard extension list, this should be addressed in the upcoming *NAV 5*.

More worrying for *NAV* users is the non-detection of Marburg, in definitions provided after this virus was known to be in the wild and a week or so after making the July WildList. *Symantec's* partnership with *IBM*, together with the implementation of *NAV 5* might well change speed and efficiency of scanning, but providing timely detection of new viruses known to be causing real-world incidents may be a more pressing necessity.

Technical Details

Product: *Norton AntiVirus for Windows 95/98*.

Developer: *Symantec Corporation*, 10201 Torre Avenue, Cupertino, CA 95014, USA, Tel +1 408 2539600, fax +1 408 2524694, WWW http://www.symantec.com/.

Availability: *Windows 95/98* with 12 MB of disk space and 16 MB RAM.

Version Evaluated: 4.0 Deluxe.

Price: $69.95 Deluxe, $49.95 Basic (*Norton AntiVirus* only).

Hardware Used: 166MHz Pentium-MMX workstation with 64 RAM, 4 GB hard disk, CD-ROM drive and 3.5-inch floppy drive running *Windows 95*.

Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Test_sets/199808.html.

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, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YP, England

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

US subscriptions only:

Virus Bulletin, 18 Commerce Way, Woburn, MA 01801, USA

Tel (781) 9377768, Fax (781) 9320251

END NOTES AND NEWS

The eighth annual *Virus Bulletin* conference is to be held at the Munich Park Hilton in Germany from 22–23 October 1998. A full exhibition will run concurrently. Features of the event include a welcome drinks reception, black tie gala dinner, partners program and speakers panel. For registration details contact Jo Peck at *Virus Bulletin*: Tel +44 1235 555139, fax +44 1235 531889, or email Joanne.Peck@virusbtn.com.

Compsec '98, the fifteenth World Conference on Computer Security, Audit and Control will take place from **11–13 November 1998, at the Queen Elizabeth II Conference Centre in London, UK.** The agenda includes an exhibition, a pre-conference workshop on 10 November and the Seventh Annual Directors' Briefing on 13 November. Early bird discounts are available for registrations received before 15 May. For details and a registration form, contact the conference secretary Amy Richardson; Tel +44 1865 843643, fax +44 1865 843958, email a.richardson@elsevier.co.uk, or visit the new Compsec '98 Web site http://www.elsevier.nl/locate/compsec98/.

From 8–9 September 1998, *Dr Solomon's* is running a two-day live virus workshop, priced £695 +VAT, at the Barns Hotel, Bedford, UK. For more information, contact Caroline Jordan; Tel +44 1296 318881 or email Caroline.Jordan@drsolomon.com.

Data Fellows* has released *F-Secure Anti-Virus Agent & Server to provide corporate system services with centrally managed, scalable protection against malicious code. In the first phase, *F-Secure Anti-Virus Exchange Agent* (for *MS Exchange*), *F-Secure Anti-Virus Notes Agent* (for *Lotus/Domino* server), and *F-Secure Anti-Virus CVP Agent* (for CVP-compliant firewalls) are available. *F-Secure Anti-Virus SMPT Agent* will follow. **Also announced was the August release of *F-Secure Workstation Suite 3.0***, claiming a world-first for the integration of anti-virus and high-security encryption. Unlike similar suites offered by competitors, this product incorporates an encryption key length of 128 bits, without the restrictions imposed by US export laws or the risks associated with key escrow features. The security manager can choose from encryption algorithms including RSA, 3DES and Blowfish. For further information, contact the Product Manager in Finland; Tel +358 9 859900, fax +358 9 85990599, or email Teemu.Lehtonen@DataFellows.com.

***Symantec Corporation* is relocating its European headquarters.** From 3 August 1998, the new address will be *Symantec Ltd*, Schipholweg 103, 2316 XC Leiden, The Netherlands; Tel +31 71 4083111, fax +31 71 4083150.

An intensive live virus workshop will be hosted by *Sophos* at its training suite in Abingdon in the UK **from 16–17 September 1998** . The course costs £595 +VAT. For registration details, contact Karen Richardson; Tel +44 1235 559933, fax +44 1235 559935, or visit the company's web site; http://www.sophos.com/.

The fifth international conference on computer security, audit and control, *COSAC'98* is to take place at the Slieve Donard Hotel, Newcastle, County Down, Northern Ireland, UK from 14–18 September 1998. Features include a pre-conference training day and full partners' programme. For more information about registering for ***COSAC'98*** contact Helen Hawkins; Tel +44 1232 738080 or email cosac@aka-associations.co.uk.

Trend Micro Inc* has released *OfficeScan Corporate Edition, which offers central control of virus protection, even from a remote site. A licence for *Trend Virus Control System* (*Trend VCS*) is included with every *OfficeScan Corporate Edition* licence. *OfficeScan* is available immediately from £400 for 50 users. For more details contact Steve White at *Trend's* UK distributors *Peapod*; Tel +44 181 606 9924, or email trend@peapod.co.uk.

The UK government has introduced a new computer security standard which enables organizations to demonstrate to clients and suppliers that their systems are secure. BS7799 is set to become a prerequisite, according to *Commslogic*, the company which is conducting a major survey into corporate concerns about computer vulnerability. For more information about the survey, contact Peter Cross; Tel +44 1252 776776.

***Secure Computing Magazine's International Conference on Network Security* will be held from 2–3 September 1998** at the Mount Royal Hotel, London. Two optional workshops – Securing Unix Networks and Remote Access Security – are being run on 1 September. For information on prices and registration, contact Debbie Rosen at MIS; Tel +44 171 7798944.