ISSN 0956-9979 OCTOBER 1998

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Trawling for trouble:** This month's feature examines the significance of the Internet in the growing problem of self-distributing viruses and malware, starting on p.13.

• **Cold calling:** Dr Igor Muttik outlines Shiver, the first effective cross-application macro infector, in the first of our virus analyses on p.9.

• **Here endeth the lesson:** Jimmy Kuo's occasional series of tutorials ends this month with the final instalment of self-help macro tactics, beginning on p.7.

CONTENTS

EDITORIAL

CD Coverage

Our readers may think that this is something of a scratched record, but cover CDs and network downloads infected with CIH or Marburg have been in the news again this past month.

The September issues of the UK's *Ultimate PC* gaming magazine and the Hungarian *PC Guru* were loaded with 20 and 23 infections of CIH respectively – or is that disrespectfully? It is interesting to compare the two publishers' approaches however.

Ultimate PC was immediately withdrawn from sale once the infection was discovered. This was enforced so thoroughly, in fact, that it took quite some string-pulling to obtain a copy! It would have been on the shelves for five days at the most. The *PC Guru* publishers discovered the infection after pressing the CDs but before distribution started. They decided the cost of pulling the issue or just the CD (not to mention the possibility of replacing the CD) was too high, so they distributed it with a note, in a very small typeface, warning of the virus' presence.

"The success of scanners to date may be mainly due to luck"

The August issue of *Ultimate PC* was rushed out ahead of schedule and included a two-page apology. The publishers set up a help line to deal with CIH-related calls and they are, as always happens in such cases, promising to improve their procedures. *PC Guru's* minimal note included many untruths and misinformation (such as a claim that the CIH virus is essentially harmless).

Marburg shipped on two cover CDs – the August *PC Powerplay* in Australia had two infected files and the September issue of the Scandanavian *PC Player* carried eight Marburg infections. Other software known to have shipped CIH-infected during the last month or so includes a Year 2000 'toolkit' from a UK developer (discovered after approximately seventy copies were shipped, it was promptly recalled and replaced with clean, re-pressed CDs) and the copy of *Wing Commander: Secret Ops* game at one of the 'official' download sites.

Most of these incidents 'started' in late July. *Ultimate PC*, for example, claims the master for its CD was made on 28 July. *PC Guru* is unclear about this, but stated that on 12 August the virus was undetectable. Ultimate PC made a similar claim in their apology, but said 2 August was the date.

The gaming community has been badly afflicted with CIH since mid June. How such 'connected' members of that community could be unaware of both their increased risk and of the fact that most anti-virus developers had made special updates available by the end of the first week of July is beyond me. It certainly does *not* reflect the past claims of these magazines and developers that they pay due care to the integrity of the software they distribute. And I am not just referring to those who have been unfortunate enough to be caught out!

Ultimate PC's response to its CIH incident has several parts. CD mastering has moved in-house, more scanners are being licensed, and *Symantec's NAV* is to be kept as up-to-date as possible (guess where the 2 August date came from?) and demonstration copies will ship on all future CDs. A nice marketing coup for *Symantec*, but does it really help the magazine purchaser? Had such a deal been been made with *PC Gamer* a few months back, then at least two cover CDs would have shipped versions of *NAV* unable to detect Marburg, the 'bonus' on *PC Gamer's* July cover CD.

I'm *not* getting at *Symantec* here. Others have exploited such marketing opportunities before. One day one of them will be kicked by it. As I said of the *PC Gamer* incident – where *fourteen* scanners failed – depending on scanners alone in such a setting is very risky; inadequate even. Scanning is good at detecting known viruses. Indeed, it is excellent technology for finding such things.

The trouble with *new* viruses is exactly that. As you do not know when you will encounter a new virus, depending on scanning technology alone is a recipe for eventual failure. The success of scanners to date may be mainly due to luck – there have been very few incidents approaching the magnitude of the recent Marburg and CIH outbreaks. If we have more, expect more infected cover CDs and commercial software.

NEWS

Gathering Evidence

Romanian anti-virus developer, *GeCAD*, has implemented a 'bot' to monitor selected IRC channels. With the increase in distribution of malware on IRC (see p.13 of this issue), *GeCAD* is interested in gauging the level of threat. It works in a similar way to a web spider program.

The project, named *The Gatherer*, uses a Java bot called jIRC to collect files sent via the DCC protocol. jIRC is passive to avoid intrusions, spam and similar things. The additional support scripts are also written in Java. Currently any files received are scanned by *GeCAD's* scanner, *RAV*, but other developers interested in having their product added should contact Costin Raiu (craiu@gecad.ro). Such products must be able to detect IRC malware.

Weekly statistics are generated and posted on the Web at http://www.gecad.ro/jirc/. In early trials, jIRC has mostly collected IRC malware, however, some DMSetup Trojans are also virus-infected. Thus, *The Gatherer* is also able to produce listings of the viruses most commonly found on IRC. To date, Spanska.4250.A tops this list, followed by Die_Hard2.4000. Of the IRC malware, DMSetup.C is the most often received, with Whacked.388.A leading the *mIRC* script viruses.

The Gatherer currently only targets DalNet, but there are plans to extend it to Undernet and IrcNet. During a month of beta tests, over 8000 files were collected ▮

No Extra Help?

The UK building society, *Halifax* – as is increasingly fashionable – runs an insurance business. Apart from offering preferential rates to its customers, its operations seem neither unduly newsworthy nor of particular interest to *VB*. However, on the morning of 10 September 1998, a regular reader of (and occasional contributor to) *VB* supplied us with a copy of a missive he received from this august financial institution in his mail (postal, not e-).

Significantly entitled 'Date Change and Computer Viruses Policy Exclusion: Important changes to your cover', it immediately piqued our interest. [*… we were compiling the Prevalence Table, after all! Ed.*] The document details an 'additional exclusion' to the *Halifax's* standard household insurance policy. Unsurprisingly, it excepts coverage of any loss or damage caused by equipment 'failing correctly to recognise [sic] data representing year 2000 or any other date in such a way that it does not work properly'.

A comprehensive exclusion, but understandable. In fact, the *Association of British Insurers* (*ABI*) issued a press release early in August stating that, in general, members of the *ABI* would not be covering problems surrounding the millenium date change 'since it is a predictable and foreseeable event'. However, the document, also excluded coverage of damage or loss due to computer viruses. This is very surprising given the foregoing justification for excluding claims based on millenium date issues.

The real surprise was in the *Halifax's* definition of the terms. For the purposes of this exclusion:

Prevalence Table – August 1998

Virus	Type	Incidents	Reports
Cap	Macro	31	15.0%
Laroux	Macro	30	14.6%
CIH	File	18	8.7%
Form	Boot	12	5.8%
Concept	Macro	10	4.9%
CopyCap	Macro	7	3.4%
Marburg	File	7	3.4%
Mental	Macro	7	3.4%
AntiExe	Boot	6	2.9%
Wazzu	Macro	6	2.9%
Angelina	Boot	5	2.4%
Showoff	Macro	5	2.4%
AntiCMOS	Boot	4	1.9%
Extras	Macro	4	1.9%
Helper	Macro	4	1.9%
Parity_Boot	Boot	3	1.5%
Ripper	Boot	3	1.5%
Appder	Macro	2	1.0%
Baboon	Boot	2	1.0%
Dyslexia	File	2	1.0%
HLLP.DeTroie	File	2	1.0%
Johnny	Macro	2	1.0%
MTE	File	2	1.0%
NightShade	Macro	2	1.0%
Npad	Macro	2	1.0%
NYB	Boot	2	1.0%
Paix	Macro	2	1.0%
Stealth_Boot	Boot	2	1.0%
TPE	File	2	1.0%
Others [1]		20	9.7%
Total		206	100%

[1] The Prevalence Table also includes one report each of: 10_Past_3.748, Chack, DelCMOS, Eco, Generic_Boot, Gollum.7167, HLLO.4608, Judge.390, Leandro, Monkey, Nina-256, NoINT, One_Half, Phantom, RP, SemiSoft.59904, Telefonica, Temple, Tequila and WelcomB.

> Computer viruses include any program or software which prevents any operating system, computer program or software working properly or at all.

In cases where the *Halifax* policy includes legal expenses cover, the *Halifax* only extended their standard exclusions to cover date problems. From the all-encompassing definition of computer viruses above, it seems to *VB* that a better title for the exclusion would be 'Date Change and Computer Software Error Policy Exclusion: Important changes to your cover'.

Excluding cover of computer virus incidents seems harsh to *VB*. Afterall, as the *ABI* says insurance 'is designed to cover the unpredictable and unforeseen'. As the *Halifax* admits to never having dealt with a computer virus claim, *Virus Bulletin* is at a loss to see how they classify such events as 'predictable' ▮

Effective CyberMediately

With all the attention paid to its purchase of *Dr Solomon's*, *NAI's* announcement of an offer to acquire *CyberMedia Inc* went almost unnoticed. In fact, it did not rate as interesting enough to warrant column inches in *VB* at the time. On reflection, this was something of an oversight.

The deal closed on 10 September, giving *NAI* a range of successful *Windows* utility programs to complement its growing line of PC maintenance and administration products. The *CyberMedia* stable includes the crash protection and recovery program *First Aid*, the automatic update locater and downloader *Oil Change*, and the highly regarded *Uninstaller Deluxe*. This last is perhaps the most interesting of the posse.

The week before the acquisition was finalized, *CyberMedia* won an injunction against *NAI's* arch-rival *Symantec*. The judge in that case found that *CyberMedia* was likely to succeed in proving *Symantec's Norton Uninstall Deluxe* product (and related technologies) infringed *CyberMedia's* copyright on *Uninstaller*. The court ordered *Symantec* to issue a 'Notice of Recall' on *Norton Uninstall Deluxe* and other products bundling it.

Symantec had just released version 5.0 of *NAV* and its *SystemWorks* bundle. *SystemWorks*, which included *Norton Uninstall*, is *Symantec's* response to the suites of anti-virus, desktop security and administration software, such as *Total Virus Defense*, that *NAI* is now attempting to popularize. This claim of *CyberMedia's* might be more than just icing on the utilities cake given the ongoing copyright infringement claims *Symantec* has against *NAI*…

Apart from the *CyberMedia* deal, the purchasing goblins at *NAI* have been busy in Europe during the last few months. Aside from acquiring *Dr Solomon's*, *NAI* has recently purchased of the Spanish producers of *AnyWare AntiVirus*, and the former distributors of *Dr Solomon's* products in Sweden, *QA Information Security*.

Both deals strengthen *NAI's* distributor network. The former is, presumably, also expected to increase *NAI's* distribution by the eventual replacement of *AnyWare's* product with *NAI's*. *VB* wonders to what extent this move into Spain is a response to the push into North America by the other major Spanish anti-virus company, *Panda Software*? ▮

Watch that Backdoor

Following last month's report about the much-heralded network backdoor, *Back Orifice* (*BO*), *VB* received several enquiries involving *BO* 'infections' and noted some interesting related phenomena.

The media coverage *BO* garnered meant that many people were talking about it on-line. Some were very worried, mostly quite unnecessarily, that they may have been 'infected'. Is it, therefore, surprising that someone posted a 'detector' for *BO* that was in fact just the thing it claimed to protect against? The poor spelling, grammar and lack of any indication of the identity of the author in the readme file accompanying *BOSniffer* should have been more than enough to warn people off running it. Those who missed such clues installed *BO* with the *SpeakEasy* plugin (which transmits the compromised machine's IP address on IRC).

VB also received a number of reports of *NetBus* incidents. Similar to *BO* in many ways, *NetBus* has been around longer but does not seem to have attracted much interest, nor caused many problems, before *BO* became 'popular'. As with *BO*, *NetBus* installs a DLL to hook various low-level system processes (such as keyboard input and mouse positioning). The *NetBus* server also installs itself so as to run at every startup and normally runs as a hidden process. *NetBus* has most of *BO's* functions, plus some twists of its own (e.g. open/close CD drawer).

The Swedish author insists it is a joke: 'With *NetBus* you fool around your friends across your local network, or even over the global internet! The purpose of this program is just to have fun, and not to systematic irritate people.' He seems to believe it will be further developed into a network and/or remote administration tool. As it appears that *NetBus* is being distributed by people with less than laudable goals, *VB* feels that the author has failed to grasp a rather fundamental aspect of human nature… ▮

NAI Hasn't a Clue?

On 25 September, *Virus Bulletin* became aware of Graham Cluley's resignation from *NAI*. Long known for his public representation of *Dr Solomon's*, as one of its Senior Technology Consultants he was regularly found championing *Dr Solomon's* product in print and broadcast media whenever a quotable expert was required. He was also a habitual participant in on-line discussion and news fora. At press time, despite rumours to the contrary, Graham has not accepted a job with any other anti-virus company and is temporarily unemployed, considering his options ▮

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 15 September 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

Acurev

CN: Two encrypted, overwriting, direct infectors with the texts 'Acurev v1.8 coded by KilJaeden of the Codebreakers on 05/29/98', '—> How Can You Think Freely In The Shadow Of A Church? <— —> You Cannot Sedate, All The Things You Hate <— —> Your Infected <—', '*.com' and 'Happy Birthday Christine Moore *kiss* I'll be home In less then a month now... June29th, Can't wait!!'. The last message is displayed on 16 July.

```
Acurev.666          E2F5 BE26 018B FEB9 7402 E803 00EB 0B90 AC32 0625 01AA E2F8
Acurev.670          E2F5 BE26 0189 F7B9 7802 E803 00E9 0A00 AC32 0625 01AA E2F8
```

Debe.1229

CEN: An encrypted, appending, 1229-byte, fast, direct infector containing the texts 'ECHO BitBoy present THE 0DeadBeefh VIRUS or Hi to Admiral Bailey! ECHO Don`t let school interfere with your education. @ECHO Y|forMAT c:>NuL*.com', '*.exe', '*.bat' and more messages in Russian.

```
Debe.1229           2E8A B611 010A F65A 7402 32F2 32E6 2E88 2743 FEC2 E2E6 C358
```

Chris.597

CN: An appending, 597-byte fast direct infector containing the texts 'Need you, Dream you Find you, Taste you Fuck you, Use you Scar you, Break you Lose me, Hate me Smash me, Erase me', 'Happy Birthday Christine! Your As Beautiful As Ever', '*.c*' and '*.tx*'. The virus overwrites the first 116 bytes of files matching the '*.tx*' mask, and sends a message to the printer on 16 July.

```
Chris.597           B440 8D96 0301 B955 02E8 AD00 B801 573E 8B96 3C03 3E8B 8E40
```

Cowa.2193

CER: An encrypted, 2193-byte virus containing the texts 'COWA-BUNGA VIRUS (C) 1994 by Turbo Power *** Claudia Schieffer Lives !!!', 'OMSPEC=', '*.COM', '*.EXE', 'SMART*.*', 'CHK*.*', '.S' 'ANTI-VIR.DAT', '*.VIR', 'NAV_._*', 'SCAN', 'F-', 'VIR', 'VSH', 'AV', 'BMB', 'BMD' and 'SOP'.

```
Cowa.2193           8905 8C4D 020E 1F0E 07FD B9DD 078D B690 098B FECC CD01 E2FB
```

Cyrus.186

CN: An encrypted, overwriting, 186-byte, fast, direct infector with the text '*.com'. As there are only two *short* potential virus templates, other means of detection should be considered.

```
Cyrus.186           B9A2 008D 360C 018B FEE8 A200
Cyrus.186           8A16 B901 AC32 C2AA E2FA C3??
```

DG.378

CN: An appending, 378-byte, direct infector which infects one file at a time. It contains the word 4742h ('DG') seven bytes from the end of the code.

```
DG.378              B440 BA05 0103 D68B 9C54 028B 8C7D 02CD 21CC CD20 2E81 3E00
```

Fayte.494

CN: An encrypted, prepending, 494-byte, fast, direct infector containing the texts '-=Fayte=-*.com' and 'C:\windows\command'. The payload modifies the CMOS data.

```
Fayte.494           2BF6 019C 1B01 4646 86FB 81FE 7E01 75F2 803E 0401 000F 85D7
```

Gwar

DMR: An encrypted, boot sector virus containing the text 'Gwar virus v1.3, (c) 1998 by T-2000 / Invaders SKLSUX!Winsuck95'. The virus infects the MBR on hard disks and DOS Boot Sector on floppies. A destructive payload that overwrites the contents of the hard disk triggers on 2 May.

```
Gwar                3DC2 C375 03F7 D0CF 80FC 0272 1880 FC03 7713 0AF6 750F 83F9
```

Hepatitis.1270

CR: A stealth, encrypted, 1270-byte, virus with the text 'Aitd magyar virus Copyright 1997-98.Na mi van csikfejüek!!! Hepatitis B.CMOS cica haj hovü lett a setupom.Sajnálom a Wincsestert.Tesztelve F-PROT 2.28'. Infected files have their time-stamps set to 60 or 62 seconds. The payload, which triggers on the 12th day of any month, overwrites the MBR of the first hard disk.

```
Hepatitis.1270      EB01 90F6 15EB 0190 8005 79EB 0190 802D 6247 47E2 925C C000
```

Hysteria.1852

CER: An appending, 1852-byte virus containing the texts 'This virus is created by Virus Generator On-Line' and '(c) 1998 Mad Daemon / maddaemon@hysteria.sk'.

```
Hysteria.1852       B440 B93C 07E8 F0FC 3D3C 0775 2880 3E3C 074D 740A BAD6 01B4
```

Messev.3158

CER: A stealth, encrypted, appending, 3158-byte virus containing the texts '[Messev v2.10, (c) 1998 by T-2000 / Invaders]', 'C:\WINDOWS\SYSTEM\IOSUBSYS\HDFLOP.PDR', 'MeSSeV LiVeS!', 'TBSCAN.PKZIP.EARJ.EXELHA.EXERAR.EXECHKDSK.', 'Daddy-K-tit 2 Gallyon van Vessem',

'This is a pretty lame virus, I only released it coz I wanted to infect some ppl.Messev - Screwed version', 'My gun will be your angel of mercy![DEMANUFACTURE - FEAR FACTORY]', 'If I don't pass... fuck it!' and 'SKLSUX!'. The virus drops another boot sector virus called Gwar (described above). Infected files have their time-stamps set to 60 seconds.

```
Messev.3158          56E4 210C 02E6 210E 1F8B DEB9 0F0C 8037 ??43 B409 BA0F 0C03
```

PB.742

CER: An appending, 742-byte virus. Infected files have the word 4250h ('PB') at offset 0012h.

```
PB.742               B91C 00B4 40CD 21C3 3DCA BE75 04B8 554A CF3D 004B 741D 2EFF
```

Priviet.1931

CER: An appending, encrypted, 1931-byte virus containing several Russian texts and the message 'Press any key to continue...'. The payload displays the text on-screen.

```
Priviet.1931         B98B 0790 B440 E8D0 FF33 DBC6 0630 0000 E809 00FA E421 24FE
```

Punky.543

CN: An encrypted, appending, 543-byte, direct infector with the texts 'Funky Punky written by Spooky. Austria 1996.', '????????COM' and '*.com'.

```
Punky.543            E800 005D 81ED 0601 80BE 2103 0074 0A8D B61C 01B9 FC00 E8F8 01??
```

Rame.965

CN: An encrypted, 965-byte overwriter containing the texts 'RAME v.01 by RedArc' and '*.com'.

```
Rame.965             ACBD 0000 8AD8 45AC E82D 0075 1653 8AF8 80E7 0F8A D8C0 EB04
```

Reu.1397

CER: An appending, 1397-byte virus containing the texts 'WOLF3D.EXE', 'Written in the city of Istanbul (c)1994 by REUIUKRGT.', 'CIV.EXE', 'CìV.EXE', 'RETAL.01', 'DISCOVR1.PAL', 'ICONPG1.PAL', 'YEAGER.EXE', 'SUP.EXE', 'FOOTBALL.EXE', 'TIM.EXE', 'TìM.EXE', 'PRE2.EXE', 'BIRTH0.PIC', 'RETAL.00', 'LHX.EXE', 'JF.EXE', 'GEOSCAPE.EXE' and 'X.EXE'. The virus can be detected with the template published in May 1997 for the Reu.1367 variant.

Ruby.1055

ER: An encrypted, appending, 1055-byte virus containing the texts '[Ruby_Tuesday]' and 'Sea4, CodeBreakers'. Infected files have the word C402h at offset 0012h.

```
Ruby.1055            00B9 FD01 8DB6 2500 8BFE E802 00EB 09AC 93AC 21D8 AAE2 F8C3
```

Sisoruen

CN: Two encrypted, appending, fast, direct infectors containing the texts 'autoexec.b*', '*.c*' 'prompt pf::Sisoruen::$g' and '????????C??'.

```
Sisoruen.453         ACF6 D0C0 C804 F6D8 3E32 863D 01F6 D8C0 C804 F6D0 AAE2 E9C3
Sisoruen.465    C8F6 D83E 3286 4901 F6D8 D0C8 D0C8 D0C8 D0C8 F6D0 AAE2 DFC3
```

Snark.819

CER: A prepending, 819-byte virus containing the text ' SNARK '.

```
Snark.819            BA11 01B9 2000 CD21 B440 BA00 01B9 1303 CD21 E8A0 FE1F 5AE8
```

Topol

DMER: A multi-partite virus infecting DOS boot sectors on floppies, MBRs on hard disks and EXE files. While infecting a file, the virus overwrites its EXE header. It contains the text: 'TOPOL'.

```
Topol                C706 4C00 A800 A34E 0050 B873 0050 CBC5 0600 012E A3D6 002E
```

Valhala.758

ER: An encrypted, appending, 758-byte virus containing the text '—va1ha1a r3mak3 v2.00, r3zid3nt, 3x3 inf3ct0r, crypting, n0 d3structi0n, gr33tz t0 gjh cr3w&suck!+-'. The virus recognizes infected files by comparing values of initial SP and CS (taken from the EXE header), for infected files SP = CS + 2.

```
Valhala.758          B9B4 02D1 E941 2E31 0383 C702 E2F8 2E8B 9EF4 0232 DFD1 E302
```

VCC.573

CEN: An appending, 573-byte, fast, direct infector containing the texts 'DEBUGGING IS VERY ILLEGAL (NOT!)', 'No, I think thats right. The idea is this will prick the boil. It may not. The history of this thing has to be though that you did not tuck this under the rug yesterday or today, and hope it would go away.', 'Ehrlichman Virus', 'eMpIrE-X', 'I-EAS Virus Creation Centre v0.19β', '[EV]', '[eX]', '[IE-VCC v0.19β]' and '*.*'. Infected files have the byte 43h ('C') at offset 0003h.

```
VCC.573              B802 422B C999 CD21 B440 B93D 028D 9606 00CD 21B4 3ECD 21C3
```

Woesti.200

CN: An encrypted, overwriting, 200-byte virus with the texts '-=[Fuckings go to WOESTI.]=-' and 'C*.*'. The first twelve bytes of infected files are 27h.

```
Woesti.200           C08E D8F7 1605 00EB 019A F716 0500 1FBE 3001 8BFE B198 AC34
```

Xchg.118

CN: An encrypted, 118-byte overwriter with the texts 'Sea4, CodeBreakers', '*.com' and '[Xchg Rate]'.

```
Xchg.118             B931 00BE 1401 89F7 E802 00EB 2CAD 86E0 ABE2 FAC3 2E2A 6F63
```

XM.828

CER: An encrypted, 828-byte appender containing the text '[XyeBo_MHe], (c)Midnigh┼Pr0wler'. Infected files have the word E940h at offset 0000h (COM) and the word FEFEh at offset 0010h (EXE).

```
XM.828               2EFF 348F 066C 0431 066C 04FF 366C 042E 8F04 4646 D1C8 FF0E
```

XM.2379

CER: A polymorphic, appending, 2379-byte virus containing the texts 'sf-mail.cfgt-mail.ctlSF-MAIL.CFGT-MAIL.CTL', 'COMMAND.COMDOS4GW.EXEIBMBIO.COMCOMEXEcomexe', '[Miscellaneous]', 'DoorWay_Password 'GLORY'', 'T-Password GLORY', 'prompt pg', 'path c:\bat;c:\bin;c:\dos', '[XyeBo_MHe], (c)Midnigh┼Pr0wler -=Version 2.1=- Bugs fixed! Almost harmless...', 'cls', 'k,,c:\dos\keyboard.sys' and 'c:\autoexec.bat'. All infected files have the word FAFAh at offset (EXE) and the byte Fah at offset 0000h (COM). The following template detects the virus in memory only.

```
XM.2379              B8F9 F9CD 213D 304E 740B B430 CD21 3C04 7203 E928 081F 0733
```

Zlodic.52

CN: A simple, overwriting, 52-byte direct infector containing the texts 'Zlodic' and '*.cOm'.

```
Zlodic.52            B802 3DBA 9E00 CD21 93B4 40BA 0001 B134 CD21 B43E CD21 B44F
```

TUTORIAL

Free Macro Anti-virus Techniques Part 4

Jimmy Kuo
Network Associates Inc

[*In the final instalment of his self-help series, Jimmy covers some options that were referred to in earlier sections and offers some* Excel*-specific suggestions. He concludes with a discussion of the techniques he employs on his own machine and some advice to people handling suspect files which must be used before expert anti-virus assistance can be obtained. Remember that unless otherwise stated, file locations are the* Office 95 *defaults and may vary for* Office 4.x *and/or* Office 97 *users. Ed.*]

ENDBAT.BAT

In previous sections there were references to ENDBAT.BAT. In the battle against macro viruses, it is important to know that many macro viruses have payloads which attach extra code to the AUTOEXEC.BAT file. Next time the machine is started, the code added by the macro virus will execute. Thus, it is important to come up with a method which prevents such payloads from taking effect.

With batch files, there are two different ways to transfer control to another batch file. One is to 'call' the second batch file, then control is returned to the 'caller' after completion of the file. The second transfers control directly, without returning control upon completion.

First, you create an empty batch file called ENDBAT.BAT. Instead of letting the AUTOEXEC.BAT end by executing the last instruction, transfer control to ENDBAT, which finishes the startup process. With this setup, no code that is added by a macro virus to the end of AUTOEXEC.BAT ever gains control. In effect, none of that code runs.

This same setup will cause software installations that add to the end of AUTOEXEC.BAT to fail in the same way. In such situations, simply move the ENDBAT transfer to the 'new' end of AUTOEXEC.BAT.

Pro: ENDBAT.BAT immunizes against the effect of viruses adding additional code to the end of AUTOEXEC.BAT.

Con: May interfere with software installations that write to AUTOEXEC.BAT. Prior to macro viruses, this was the main use for this setup.

Rename DEBUG.COM and DEBUG.EXE

Another method favoured by virus writers is a debug script. This is a readable text file of debug instructions, which is sent to the DOS utility to create a binary file. Usually, this is used to deliver virus programs or other binary data.

You can rename or remove debug from users with no need for that program (the majority do not ever need or know how to use debug). Verify that the program is no longer available by typing 'debug' on a command line. If the program still runs, the job is not yet complete.

Pro: You will not be affected by viruses that use debug to deposit payloads onto machines.

Con: Users do not have the program to use. Not a problem for most users.

Excel Macro Viruses

XLSTART is similar to *Word's* startup directory. Any template file found in XLSTART is automatically loaded into *Excel* on startup. This behaviour is exactly the same as *Word's*, and means that you can use similar code as discussed in the Check the Startup Directory section.

```
dir /b/o/a \msoffice\Excel\XLStart >
    %TEMP%\xls.lst
```

Add the following to AUTOEXEC.BAT:

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

The Pros and Cons are exactly the same as with the same function for *Word*.

Create a PERSONAL.XLS File

You should have learned not to do this by now. This technique is equivalent to creating a Payload macro to address *Word* macro viruses. Laroux.A checks for the existence of a file called PERSONAL.XLS in *Excel's* XLSTART directory. If one exists, the virus does not infect. Thus, if you put a file of that name in that directory, you will be immunized against Laroux.A, the most widespread of all *Excel* viruses.

However, as happened with *Word*, other viruses appeared and other variants of Laroux now exist, rendering this technique effectively useless. To create such a file, simply take an empty *Excel* file and place it in the XLSTART subdirectory under *Excel*.

Pro: Works against Laroux.A.

Con: Only works against Laroux.A.

An interesting consequence of cleaning a Laroux infection from the system is that it leaves a clean file by the name of PERSONAL.XLS (or BINV.XLS, etc.) in the XLSTART directory. Although I do not suggest creating one to thwart infection, I do recommend that the file should be left there. There are two reasons for this. Firstly, it has already been proven that the particular strain of the virus is spreading nearby and a defence against it is prudent. Secondly, removing the file is actually more work than leaving it.

Author's Recommendations

I have so far made hints as to the usefulness of each method, without suggesting which combination to use. I am not going to commit to that, but rather tell you which I use.

I have a read-only NORMAL.DOT. In addition, I use Prompt to Save Normal Template. Earlier, I made a quick comment that the two do not conflict and that it is possible to use both. Both are meant to warn you by the end of the day if your environment has been infected. However, why use both? Is one not enough?

The first answer is that it does not hurt, so why not? The second is that some viruses try to undo one or the other. Some even try both. So, using two techniques means a virus has to attack both simultaneously to circumvent the protection. If nothing is happening, both are quiet, so they will not disturb your everyday work.

I also use the DisableAutoMacros template as distributed in the separate file NOAUTO.DOT. Most viruses make use of some sort of auto macro in order to spread – all In the Wild viruses do. With this macro in place, viruses will not activate automatically and the chance of spreading something, even if you come in contact with it, is reduced. Furthermore, as described in its own section, an MIS director can create this file and send it to the whole company to be placed in the appropriate location. Thus, this can have a wide corporate impact with little effort.

In preparing this series, I actually tried most of these techniques. I plan to incorporate the checking of the XLSTART directory and *Word's* startup directory. The XLSTART directory technique is the only significant method against *Excel* viruses. Sadly, I was infected by Laroux.A from within my own company recently. (Luckily, I recognized it immediately within a minute.) So, it is starting to hit home, and more *Word* viruses seem to be taking advantage of the startup directory technique as well.

Lastly, all *Office 97* products are programmed to alert if any macros exist in an incoming document, be that *Word*, *Excel*, *Powerpoint*, *Access* or any other program. The products have the macro alert on in their default mode. Do not turn it off until you hit your first false alarm. Even then, judge how much trouble the false alarm has caused. If you feel that it is not a problem, leave the setting on. The alert is not perfect (see Vesselin Bontchev's paper for the 1996 Virus Bulletin Conference). Until you meet a false alarm, the macro alert does not cause you any headaches, and it takes effort to turn it off anyway. You might as well do that later rather than sooner.

Handling Suspect Documents

Here are some tips for MIS directors who must handle suspect documents. Use all the techniques above. If a file is suspect, create a clean environment by using the process outlined in the Disable AutoMacros section (see *Virus Bulletin*, April 1998, p.11). Examine the file using File, Templates, Organizer before opening it or any other files. If the suspect file does not have a ToolsMacro entry, use it to rename the macros with shortened names before examining them. If it does, create your own ListMacros menu option and use it instead of ToolsMacro.

Lastly (my only plug for my own product), if you scan the single file with the DOS version of *VirusScan* and it reports 'Analyzed: 1, Scanned: 0, Possibly Infected: 0' then the file has no macros and thus cannot have a virus. If it reports 'Analyzed: 1, Scanned: 1, Possibly Infected: 0', then the file *does* have macros. Send it to your favourite (or not) anti-virus researcher and they will tell you if it is infected.

Cleaning Infected Documents

There are also rare occasions when an MIS director must clean an infected document immediately, so the document can be used without delay. As anti-virus vendors, we recommend against this, but we also recognize that we cannot necessarily help you every minute of every day. Please use this technique with utmost care, and only if you cannot avoid it. This is best done on a standalone machine but if this is impossible, be extra careful!

After verifying that the virus does not have an EditCopy or EditCut macro, and that there are no templates in the startup directory or NORMAL.DOT, open the file while holding the shift key (or for the more adventurous, place NOAUTO.DOT into the startup directory). Select the entire document, and Edit, Copy to the clipboard, then File, Exit from *Word*. Next, delete NORMAL.DOT (or rename it) and remove all files from the startup directory. Restart *Word*, then select File, New to make a new empty document. Then Edit, Paste from the clipboard. Finally, File, SaveAs to a new file. In so doing, be sure that the file is not being saved as a template automatically. If so, the environment is infected. Assuming all the above is handled properly, pick up the phone and call your anti-virus vendor.

Acknowledgments:

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

VIRUS ANALYSIS 1

Time to Shiver

Dr Igor Muttik
Network Associates, UK

Viruses capable of infecting different kinds of objects have always attracted attention from virus writers. The reason seems fairly obvious – the more incarnations a virus has, the greater its chances of spreading. To a certain extent this is true – multi-partite DOS viruses (like Junkie and One_Half) are more common in the field than normal file infectors and they occupy higher positions in both the WildList and *VB's* Prevalence Table.

On the other hand, multi-partite viruses are more complex and it can be more difficult to get them to work properly. Thus, such viruses are also something of an intellectual challenge for virus writers. In any case, in the wild viruses which infect different kinds of objects are rather rare.

There have been attempts to create multi-application macro viruses, such as Cross (see *VB*, June 1998, p.11) and Teocatl (aka Strangedays). These are huge and very conspicuous viruses (about 300–500 KB in size) which contain bugs or design limitations that mean neither of them work very well. They are very much in the category of experimental viruses.

At the end of August, however, a new attempt was made and it was successful – the Shiver virus appeared and several variants of it were distributed in documents supposedly containing lists of cracked porno sites (URL/login/password records). The virus is much more compact than its predecessors – about 30 KB.

Shiver is the first virus capable of the successful infection of both documents (DOC) and spreadsheets (XLS) created in *Word 97* and *Excel 97*. It is also the first macro virus that uses *Windows'* DDE (Dynamic Data Exchange) mechanism, which is supported directly in VBA (Visual Basic for Applications). Using VBA's DDE functions, *Word* macros can send data and commands to *Excel* and *vice versa*. This is the mechanism Shiver uses to cross-infect between the two applications.

The author of this virus calls himself 'ALT-F11' and Shiver is not his first virus. He wrote the Groov family, one member of which is a polymorphic *Word 97* virus. Shiver is much more complex than Groov because of its ability to cross-infect *Word* and *Excel* files, but it is not polymorphic.

Infection via Documents

When an infected document is opened, the virus takes control via its AutoOpen macro. The first thing Shiver does is disable macro virus protection for both *Word 97* and *Excel 97*. Then it exports its whole VBA source code into a file called C:\SHIVER.SYS. This file is never deleted by the virus, and is thus a clear indication of infection.

The use of just one file for transferring the virus' source between applications could be problematic. Fortunately, (for the author of the virus), a feature of *Word* and *Excel* simplifies matters. The names of *Word* and *Excel's* auto-open macros are different for the two applications – they are AutoOpen and Auto_Open, respectively. Thus, the same VBA source can be used in infected documents and spreadsheets without having to deal with a name conflict.

Next, the virus infects the global template, NORMAL.DOT, by importing the source code of the virus into a VBA module called Module1. This infection mechanism is already known and it bypasses the anti-virus feature introduced in *Office 97 Service Release-1*. The virus disables the ToolsMacro and FileTemplates menu options, and access to the Visual Basic editor.

When the infected *Word* next starts up, Shiver creates WORD8.DOT (via the AutoExec macro), adding it to the list of default templates. The module name in this file is Sentry. If the user erases the infected NORMAL.DOT or otherwise removes the viral macros from it, this template restores the virus from C:\SHIVER.SYS. By building WORD8.DOT, the virus creates C:\SENTRY.SYS. This file is rather short – it only contains the code to import C:\SHIVER.SYS back into NORMAL.DOT.

When the infected *Word* is shut down, Shiver checks whether the system is already fully infected. It does this by examining a particular key in the Registry (see Fig 1). If *Excel* is not infected, the virus starts it (via DDEInitiate). This makes *Excel* run as a DDE server. Then it puts some formula macros into the cells of the new spreadsheet (using DDEPoke). Here the virus acts as a DDE client to *Excel*.

Shiver then launches the newly-constructed macro using DDEExecute. The purpose of this macro is to import the C:\SHIVER.SYS file into the PERSONAL.XLS file in the XLSTART folder (overwriting it if one already exists). As PERSONAL.XLS will be loaded automatically the next time *Excel* runs, Shiver has now effectively spread from *Word* to *Excel*. All *Word* documents and *Excel* spreadsheets opened thereafter will become infected.

Fig 1: Shiver adds this entry to the Registry once it has infected both *Word* and *Excel* environments.

After infecting *Excel*, Shiver sets a flag in the system Registry to indicate its presence. The value Shiver[DDE] in HKCU\Software\VB and VBA Program Settings\Office\8.0 is set to 'ALT-F11' (see Fig 1). If an error occurs, the virus may set this value to 'NoNoNo'. This key is used to check whether the system is already infected – if it is set to 'ALT-F11', the system is considered fully contaminated.

Infection via Spreadsheets

If the virus enters the system as a spreadsheet, it creates C:\SHIVER.SYS and then imports it into PERSONAL.XLS to infect it. The name of the module is again Module1.

Shiver then creates two files (C:\O6.REG and C:\O6.BAT) to modify the Registry key responsible for *Excel's* macro virus protection. O6.REG contains just:

```
[HKEY_CURRENT_USER\Software\Microsoft\Office\
    8.0\Excel\Microsoft Excel]
"Options6"=dword:00000000
```

The batch file runs REGEDIT.EXE, supplying 06.REG as a parameter. This modifies the Registry, disabling *Excel's* macro virus check on loading a spreadsheet. This method of modifying the Registry may seem unduly involved, but is used because, unlike *Word's*, *Excel's* VBA has no method for the direct manipulation of the virus-protection option.

Fig 2: Shiver breaks double-clicking on DOC files with this Registry addition. A similar change also afflicts XLS files.

When an infected spreadsheet is closed, the macro virus starts the sequence of actions resulting in the infection of the *Word* environment. The virus checks the Registry, and if the Shiver[DDE] value is not set to 'ALT-F11', it tries to infect *Word's* global template. *Word* is started as a DDE server and the viral macro in *Excel* is its DDE client.

Using the VBA SendKeys function, the virus launches the Visual Basic Editor (VBE) and feeds it the virus' full source (from C:\SHIVER.SYS). This process is clearly visible – the virus literally types 'c:\shiver.sys' into VBE's Import File dialog box. This is because the virus minimizes *Word* but cannot minimize the VBE. It then closes the VBE and *Word*, saving the infected NORMAL.DOT. All documents opened thereafter will be infected as *Word's* global template now contains the virus. Finally, as when infecting the system via a document, the virus sets the value of Shiver[DDE] to 'ALT-F11' to indicate complete infestation.

Shiver's DDE mechanism of cross-infecting other *Office* applications is much more elegant than the clumsy (and space-intensive) methods used by both Cross and Teocatl. Those viruses carry huge Debug dumps of binary droppers – for example, Teocatl contains dumps of pre-infected PERSONAL.XLS and NORMAL.DOT files, which occupy approximately 95% of the virus' code. That method is also more fragile, as it relies on the presence of Debug to write the droppers to disk. DDE is part of the operating systems on which Shiver's hosts run – in the Cross and Teocatl case, Debug may be missing. [*…or at least renamed! Ed.*]

Variants and Payloads

Three variants are known at the moment. These appeared within a week, so we anticipate more rewrites. The first two are quite similar and differ in that the .B variant checks whether 06.REG and O6.BAT already exist. All three differ in the document payload. Variants .A–.C have 314, 326 and 365 lines of VBA code, respectively – the third variant having a longer payload. In comparison, Teocatl has 3053 code lines, 2920 of which are taken by Debug dumps.

Shiver has two payloads – one attached to the auto-open macro of each application. Thus, one of the payloads may trigger even if only *Word* or *Excel* is installed.

When a document is opened, the .A variant (with a probability of one in 800) modifies the Registry. The affected keys are HKCR\Word.Documet.8\shell\open\ddeexec and HKCR\Excel.Sheet.8\shell\open\ddeexec. Following this, nothing appears to happen on double-clicking a document or spreadsheet (in fact, the application opens briefly, then closes). To correct this, delete the key highlighted in Fig 2.

The .B variant has the document payload code commented out, whereas the .C variant has a quite different, larger, payload. With a probability of one in 75 it replaces several menu items with messages like 'Wanna do some MDMA?', 'Peace, Love and Drugs', 'I'll die happy, you'll just die' and many others. These menu customizations are stored in NORMAL.DOT. As *Word* provides no simple means of undoing these changes, the easiest fix is to delete that file but this may remove the user's own customizations. The payload, with a probability of one in 405, also creates C:\SISTER.DLL and displays it (see Fig 3).

```
Hey Man, I Kinda Like Your Sister
Hey Man, I Hope That's Cool
Hey Man, I Kinda Lose My Mind
Every Single Time I Find Your Sister
Suntanned By The Pool
Hey Man, I Wanna See Her Naked
Hey Man, I'm Always In Her Room
All Alone When No One's There
Going Through Her Underwear
Hey Man, I Gotta See Her Soon
Hey Man, I'll Never Get Her Pregnant
But Hey Man, How Can I Resist Her
The Day I Give Her A Wedding Band
Are You Going To Be My Best Man?
Hey Man, I Kinda Like Your Sister
I Kinda Like Your Sister
I Kinda Like Your Sister
I Kinda Like Her
```

Fig 3: The contents of SISTER.DLL. The *Word* payload of Shiver.C can write this file and display it with WordPad.

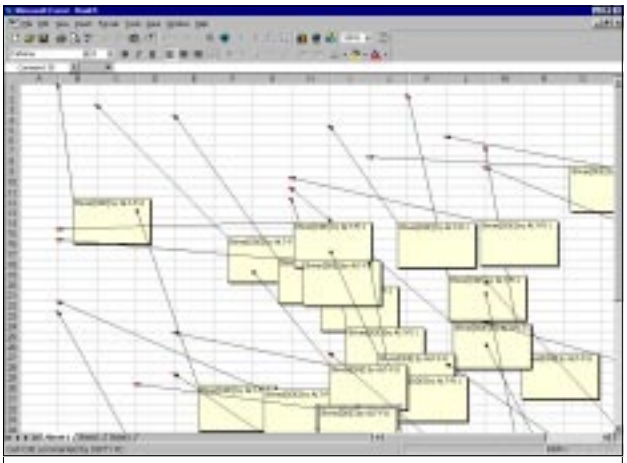

Fig 4: The visual mess produced by the *Excel* payload is easily corrected and, fortunately, no data is altered.

All three variants have the same *Excel* payload. On opening an infected spreadsheet, with one in 800 probability, Shiver attaches a comment to thirty randomly-selected cells in the top left 30 x 30 block of the active sheet (see Fig 4). No data is lost, but the sheet looks seriously messed up. This is rectified by applying Edit/Clear/Comments to the top left 30 x 30 corner of the spreadsheet.

Conclusion

It seems likely that Shiver is the precursor to a bunch of multi-application viruses using a similar approach. The use of DDE (and other VBA goodies, as yet undiscovered by the virus writers) is a much more elegant and portable cross-infection method than the machinations employed by both Cross and Teocatl. This method seems likely to be extended to more applications. It can now only be a small step to polymorphic cross-application macro viruses.

Shiver

Aliases:	None known.
Type:	*Word 97/Excel 97* cross-application macro infector.
Self-recognition:	Potential hosts that contain a VBA module named 'Module1' are assumed infected. If the value of Shiver[DDE] in the Registry key HKCU\Software\VB and VBA Program Settings\Office\8.0 is not 'ALT-F11', cross-application infection is initiated.
Payload:	Several, triggered randomly (see text).
Removal:	Delete NORMAL.DOT, WORD8.DOT and PERSONAL.XLS. In a clean *Word* environment, delete Module1 from infected documents via the Organizer. There is no simple, manual procedure for disinfecting spreadsheets.

VIRUS ANALYSIS 2

Breaking the Lorez

Péter Ször
Data Fellows

These days, writing *Windows 9x* viruses is not as difficult as it was a year ago. More and more *Windows 9x* viruses are documented, complete with source code, in virus writer magazines. Those sources can be used as 'study guides' in the creation of other viruses, which certainly simplifies the writing process. Nevertheless, it is still early days for *Windows 9x* and *NT* viruses, and new techniques are often to be found in the latest creations.

Once Upon a Time

Early *Windows 95* viruses were direct action infectors, hence not going resident. As they often caused a noticeable slowdown during infection, virus writers soon started looking for more effective ways of implementing fast infection. The most challenging problem of *Windows 95* viruses became the question of TSR mechanisms.

Creating an active virus is not an easy task and despite their complexity, early *Windows 95* viruses evolved swiftly. Punch (*VB*, April 1997, p.8) and Memorial (*VB*, September 1997, p.6) used specific VxD droppers, the result of which were full VxD-based virus bodies. This made the virus code complicated, because it had to be converted into different file formats. Just a couple of months later we saw viruses capable of direct VxD and *Word* document infection, such as Navrhar (*VB*, November 1997, p.15).

Soon after that, Anxiety (*VB*, January 1998, p.7) simplified things by patching into the VMM directly. Then Cabanas (*VB*, November 1997, p.10) introduced a per-process resident strategy by hooking the imports of host programs (so far this is the only method which also works under *NT*). Last but not least, HPS (*VB*, June 1998, p.13) demonstrated VxDCall hooking by patching KERNEL32.DLL's local heap, and installing itself in shared memory.

Most *Windows 95* viruses are Portable Executable (PE) infectors, although some infect DOS COM and EXE programs, VxDs, *Word* documents and 16-bit *Windows* New Executables (NE) as well. Others may accidentally infect dynamic-link libraries (DLLs) which are linked in PE (or NE) formats. In these cases, the infection is unable to spread further because the 'standard' entry point of a DLL, which such accidental infections usually intercept, is not called by the system loader. Normally, a DLL's execution starts at its specified DLLEntry-point.

Win95/Lorez is a PE infector that targets KERNEL32.DLL with a new attack. Instead of modifying the entry point, it patches the export RVA (Relative Virtual Address) of a

critical API to point to the virus code at the end of the DLL. But how is it able to replace the original KERNEL32.DLL with an infected one?

Executing Lorez

The virus code gains control when an infected program is executed. The entry point of the infected program points to the first byte of the virus body. At first Lorez tries to determine the OS version, checking the stack for the KERNEL address and calculating the base value of the return address. This new trick seems likely to be used in the future, as it simplifies the infection process.

Lorez tries to infect in cases where the base vlaue is BFF70000h (*Windows 9x*) and 77F00000h (*NT*), or it terminates by executing the host program. It tries to install itself under *NT* too, but perhaps the virus writer did not have the time or equipment to test the virus under any system other than *Windows 95*? Some of the code's assumptions are wrong and for this reason the virus is unable to work under *NT*. However, it works under *Windows 95* without any major side effects.

Next, the virus obtains the addresses of the KERNEL32 APIs it needs to call. It also acquires the address of the ExitProcess API, which is used not by the virus but by the original host program. These addresses are stored in a DWORD table at the end of Lorez' code for use by its own 'GetProcAddress' function. Its attention is now turned to infecting KERNEL32.DLL.

KERNEL32.DLL/EXE Infection

First, Lorez obtains the *Windows* and system directories using standard APIs. Normally, KERNEL32.DLL is in the system directory and cannot be written to when it is running. It can, however, be read so the virus copies it to the *Windows* directory. Then it sets a kernel infection flag for itself, before calling its standard PE infection routine.

The infection routine is passed the full path and name of the file as a parameter. It calls the GetFileAttributesA function, setting ECX to 12345678h before the call. This is a check for its presence in memory – Lorez' GetFileAttributesA routine simply terminates if ECX is 12345678h. If that happens, KERNEL32.DLL infection is not necessary. However, if the function returns the correct attributes and no error, Lorez assumes it has not already hooked the API, so it saves the original file attributes and changes them to archive, thus allowing the file to be written to. The copy of KERNEL32.DLL is opened for infection and the DLL's original time/date are saved (including the creation time/last access/last written to fields) for later use.

Lorez then checks if the file is an EXE by looking for the 'MZ' marker. After that, it moves to the PE header to check the PointerToSymbolTable field. If this is non-zero it aborts the infection. This field is zero in most PE files (including the original KERNEL32.DLL) and is usually only used in OBJ files and PE files with COFF debug information. At the completion of infection, the virus sets this field to a random value to avoid re-infection.

If the host seems clean, Lorez checks the kernel infection flag. Since this is set during initialization, the virus skips the Base Address check. The Base Address of the executable has to be 40000h in normal EXEs – Lorez does not infect the rest. This is a simple, effective way to avoid infecting nonstandard applications.

Next, Lorez gets a 'random' number from GetTickCount to use as an infection marker in the PointerToSymbolTable field of the PE header. The size and attributes fields of the final section are changed to reflect the increased size and executable status. If infecting KERNEL32.DLL it determines the location of GetFileAttributesA, otherwise it modifies the PE entry-point. To do the former it finds the image offset of GetFileAttributesA from the export table.

At this point the code does some nonessential checks for the name of the export section in order to eliminate different versions of KERNEL32.DLL for *Windows 95, 98* and *NT*. Despite this, Lorez only works on *Windows 95*. This function eventually patches the GetFileAttributesA export RVA to point to the hook function in the virus code, saving the original so it can jump back to it.

Finally, it writes the new PE header to the host and appends its body to the end of the victim. The size of the virus is 1766 bytes, but the effective length varies because of the section alignment. It resets the attributes and date/time stamp and executes the original host.

Going Resident

There is now an infected KERNEL32.DLL in the *Windows* directory but the original is untouched in the system directory. When the PC is next booted, *Windows 95* will load the infected version. It seems that KRNL386.EXE uses the 16-bit LoadLibrary search logic and thus will load KERNEL32.DLL from the current directory first.It appears that the current directory here is the *Windows* directory while the directory in which the application is executed is the system directory. (The Win32 version of LoadLibrary uses the directory from which the application loaded then the current directory and so on.) For this reason the infected KERNEL32.DLL will be loaded at boot time.

When a DLL that a newly-executed application requires exports from is already loaded, *Windows 95* does not reload the DLL. Instead it attaches the application to the DLL by mapping. When any PE program is executed (for instance, a Win32 anti-virus program) in a Lorez-infected environment, it will be attached to a non-reliable, infected representation of the KERNEL32.DLL.

The new hook function will redirect the call to the infection routine when the application calls the GetFileAttributesA API. The general infection routine is the same during

KERNEL32.DLL infection, but the entry points of the victims are modified to point to the virus code instead of looking for GetFileAttributesA exports there.

Conclusion

As we know, *Windows 9x* is far from secure. *NT* is able to stop Lorez' KERNEL32.DLL manipulation, because the system loader makes several additional checks before it loads the image. Most importantly, *NT* will only load KERNEL32.DLL from the SYSTEM32 directory.

Further, system DLLs contain a checksum in their PE header. Unlike that of *Windows 9x*, *NT*'s loader calculates the file's checksum before loading a DLL. If this does not match the recorded checksum, the loader halts during the blue screen boot-up with an error message.

These additional checks do not mean that such a virus cannot be implemented for *NT*, but they do make it more complicated. While the checksum algorithm is not documented by *Microsoft*, there are APIs available for these purposes. Even this is not enough for the *NT* loader – there are several other checks to pass, but it seems prudent to assume that virus writers will solve these problems.

Lorez is based on the Yurn virus and it only works under *Windows 95*. Win32 viruses are already on the road to polymorphism – Lorez's infection technique may be combined with a polymorphic engine in the future. This would lead to scanning problems similar to those caused by polymorphic inserting DOS viruses like Zhengxi and Nexiv_Der (see *VB*, April 1996, p8 and p11 respectively).

Lorez

Aliases: None known.

Type: Windows 95 PE infector, attacks KERNEL32.DLL.

Self-recognition in Files:
If the PointerToSymbolTable field in PE header is non-zero, the virus does not infect. This field is set to a random value during infection.

Self-recognition in Memory:
With ECX=12345678h call function GetFileAttributesA. The virus' handler simply terminates without returning an attribute and no indication of error.

Hex Pattern in PE files:

```
58FF E08B 8557 1740 0050 B978
5634 12FF 95E6 1640 0089 8553
1740 0083 F8FF 7501 C36A 208B
```

Payload: None.

Removal: Replace infected files from backup or from clean originals.

FEATURE

1998 – The Year of the Net?

The last twelve months or so have seen many interesting new directions followed by the virus writers. Perhaps most notable among these, and an area in which we fully expect we will see a lot more development, is the addition of network-awareness to viruses.

In the Beginning…

PCs started out as the small-fry of the business computing world. Much as their growth and success has seen them become the home computer of choice, this largely follows their unprecedented success in the corporate realm. Today we have near-ubiquitous networking. However, in the heady days of the first ten years of the PC's development, networking was not only rare but the most impenetrable of the black arts of computer configuration and support. Those times seem much more distant than the seven or so years that actually separate that era from this.

Prior to this current age of ubiquitous networking – in fact, until quite recently – virus authors have largely depended on fate to distribute the seeds of their labours. Reports from the past show that some boot infectors, and the very occasional file infector, had a 'lucky break', usually being widely distributed on magazine cover diskettes or driver diskettes. Very rarely, infected application software has also been distributed.

In the good old days, 'sneakernet' was the commonest form of PC-to-PC software transmission. Thus, it was quite understandable that boot viruses and multi-partites accounted for most virus infection incidents. Despite some initial trepidation, the initial blossoming of Internet usage saw little or no change to this. Although there were occasional instances of viruses and Trojan Horses posted widely on Usenet, for example, the balance of field infection reports was little changed.

Net Results

Of course, the cost of networking has fallen dramatically, along with the cost of all other PC components and software. Further, the difficulty of setting up and configuring networks has fallen, though not as quickly as prices. One advantage of the PC's popularity has been increased production and lowered cost. Another benefit has been that as network software has become more standardized, it has become easier for software developers to write more universally useful code.

With increasing interest in 'the Net' amongst both end-users and service and product suppliers, the last few years has seen a huge rush of interest in establishing and partici-

pating in Internet activities, especially the World Wide Web. In fact, the ease of configuring Internet connectivity is attributed by many as a major reason for the tremendous success of *Windows 95*. This was not a fortuitous event. Many networking interfaces were standardized by *Microsoft* and others to ease and encourage development of network-enabled (and enabling) applications, both for *Windows 95* and its ultimate successor, *NT*.

Standardization has its costs though. As others have already observed (e.g. see *VB*, August 1998, p.10), greater homogeneity in terms of operating system and network functionality, produces an Internet more vulnerable to systematic attack. The very homogeneity of systems that has contributed so greatly to the success of the Internet can also be seen as something of an Achilles heel.

Lessons from larger systems often apply to the humbler, and so it is here with desktop PCs. It should not be surprising that increasing homogeneity (and universality) of PC networking has similar pros and cons to those seen amongst the Internet's server population. Thus, just as it seems hackers and system crackers are taking advantage of the increasing similarity of the systems that comprise the Internet's backbone, virus writers are showing signs of network-enabling their creations.

The Dynamics of Spread

Many people say that the nature of viruses is that they spread. This is not necessarily so. Viruses *replicate*. Replication can be across a network link to media on another machine, but should that happen, it is usually accidental, or at least incidental to some particular infected machines. Viruses do spread, but that is not a fundamental requirement. We shall ignore for now a question researchers have not really settled – are worms a subset of viruses or are the two disjoint? [*I believe they should be defined as disjoint sets, but I accept that if worms are a subset of viruses we would have to add a requirement of some form of spread to separate 'those viruses that are worms' from 'ordinary viruses'. Ed.*]

Viruses have mainly infected a single class of host, with DOS executables and PC boot sectors by far the commonest classes until recently. However, some more adventurous virus authors have had their creations infect two (and occasionally more) types of host. Some of the early multi-partites, as such viruses are known, were quite 'successful' in spreading widely, but in these cases, multi-partism *per se* was not the most important factor.

The relative success of boot infectors over file infectors transfers to multi-partites that include boot infection in their repertoire. This was largely responsible for the early multi-partites' spreading, as most infected boot sectors and executables. Some 'odd' multi-partites (COM and BAT, say) never had a chance of becoming widespread. People probably exchange BAT files much less than they do COMs, so such a union was unlikely to go far.

The lesson to take from this is that mobility is important for a virus to become widespread. Sneakernet distributed virus-infected diskettes between machines. Disproportionately, diskettes were carriers of boot viruses. This supported, or enhanced, the position of boot viruses, and some multi-partites, amongst infection reports.

The significance of mobility became highly obvious (if it was not already) with the emergence and spread of macro viruses. Many *Word* users edited their email (received and sent via *Microsoft Mail* or the *Exchange* client shipped with *Windows 95*) in *Word*. Many of them were unaware that they were often sending *Word* document files under this arrangement and upon receiving such a message from another *Word Mail* user, opening the message to read it resulted in the document being opened in *Word*.

Independent of *Word Mail* issues, documents are exchanged more often and more readily than programs or diskettes. Early macro viruses ran riot in such environments.

Following its release, Concept rapidly became the most widespread and most commonly reported virus. It had the double-whammy advantages of both being the first *Word* macro virus released into the wild and of being launched into an increasingly 'email-centric' world. The first meant it had a major head start on the anti-virus developers, while the second gave it its wings.

Hitchin' to Go...

Keeping this potted history in mind, we shall look at some more recent developments on the front lines.

Along with increased interest in PE infection – the gatekeeper of *Windows 9x* and/or *NT* infection – the last year or so of virus developments may have been most notable for the interest some virus authors have shown in improving the 'spreadability' of their handiwork, not just its infectibility. In the days of sneakernet, there was little the virus authors could do in this regard, other than deciding to write a boot virus or a boot/something multi-partite. However, ubiquitous networking has opened up many opportunities for the enterprising virus writer…

Perhaps the first of the recent crop of viruses displaying concern for their ability to spread was Anarchy.6093 (see *VB*, October 1997, p.6). It implemented a hitherto unseen, and seemingly odd multi-partism, infecting executables and *Word* document files. Anxiety can be thought of as a proof of concept that DOC files can be 'externally' infected (i.e. other than in the *Word* environment), and was in fact technically interesting for this.

A broader view, however, suggests something deeper. The author may well have realized that new macro viruses could spread far and wide much more readily than new executable file infectors. Technically, Anarchy.6093 does not infect DOC files – it turns them into Trojan Horses, which in turn drop and run an infected program. Even so, hitching a ride

with a file type that is more likely to be disseminated than your primary host is a smart move if the virus author is interested in spreading his creation around.

It's Getting Dark in Here!

Further awakening of network awareness amongst virus writers occurred with the appearance of the *mIRC* script viruses (see *VB*, April 1998, p.7). For many researchers, the fundamentally viral nature of these scripts was overshadowed as they focused on the fact that the scripts copied themselves from machine to machine and thus were, superficially, worm-like. Although by no means settled amongst virus and security researchers, worms have traditionally been described in terms that do not require (and perhaps even prohibit) the need for a file-based copy (other than in the original 'launcher').

The simplicity and power of the *mIRC* scripting language exposed by the SCRIPT.INI viruses has stimulated those wishing to spread various kinds of malware. *mIRC* is very popular, thus, it should not be surprising that it is now used as a distribution mechanism for worms and viruses, despite its author fixing the security design flaw that made straightforward SCRIPT.INI viruses possible.

To use *mIRC* as a distribution mechanism, all a virus or a Trojan has to do is determine where *mIRC* is installed and drop a simple text script in the correct folder. The script itself only needs to contain a single line setting up a condition to send the Trojan or an infected file (or a dropper) to other IRC users. Despite incessant warnings against doing so, a (significant) proportion of the recipients of such 'anonymous' file transfers will accept them and then run the received program to 'see what it does'...

For example, while writing this article, *VB* received an email message from someone whose PC was infested with copies of DMSetup, NetBus and Back Orifice. It transpired that his children had recently received all three while on *IRC* and had run them. With moves in many countries to increase Internet access in schools (as if access to so much unstructured information is an educational panacea!), we can only imagine matters getting worse!

There is an important lesson here – apart from not running any 'strange' software (which you all knew anyway). Once a technique with greater applicability than just allowing a viral replication mechanism comes to attention, we will likely see it used again and again. As the default *mIRC* script folder is no longer the default download folder, the openings for *mIRC* script viruses have reduced significantly. However, the power of the scripting language is now being used in other malware.

The DMSetup Trojans 'replicate' between *mIRC* users through this mechanism. Two recent viruses also use the technique to send copies of themselves to other, probably uninfected, computers in the hope that the recipients will run these files.

One is the multifaceted Win95/Inca (a polymorphic PE infector with a lot of other tricks) and the other is W97M/AntiMarc. Yes – a *Word 97* macro virus cashing in on ubiquitous networking! It drops copies of infected documents randomly and arranges to distribute them, and not only via *mIRC*, as just described. It also extends the ideas in the next section to *Outlook Express*…

Red Alert

The RedTeam 'email virus' (see *VB*, May 1998, p.6) generated more heat than light for a while earlier this year. RedTeam is a *Windows* executable virus that will send copies of itself to people in your *Eudora Mail* address book, should you have that mail program installed. Some may argue that its reverse-psychology approach for trapping the unwary into running the email attachment dropper is its most interesting feature, and that may well be so.

Fortunately, because the virus was not released in what would seem the most 'natural' way for it, we do not know how effective that ruse would have been. As its cover is now blown, the success (or otherwise) of any future release of RedTeam into the *Eudora Mail* community should be quite limited. However, independent of that issue is the point that it was one of the first PC viruses that tried to distribute itself via email.

Approximately a year before RedTeam, the *Word* macro virus WM/ShareFun had shown a similar interest in email. As with RedTeam, it was tied to a particular email application – in its case, *Microsoft Mail*. RedTeam placed messages with self-infected attachments directly in *Eudora's* outgoing email queue. ShareFun, however, took a very different approach. It used the *Word* macro language's ability to 'remote control' other DDE-enabled applications, sending keystrokes to *MS Mail* much as a user completing the same task would. [*This technique is used to assist Shiver's cross-infection methods – see p.9, this issue. Ed.*]

We Don't Need No SMTP

Late in May 1998 anti-virus researchers started receiving samples of a new *Word 97* macro virus – W97M/Groov (also known as Groovie and Ipattack). Some of these came from the wild – for example, *VB* received a sample from a small, local company that has Internet email but otherwise has not seen much use for its network connection.

In fact, the people at this company seemed to know nothing of the Internet except email. They became aware of the virus' presence because of a bug in its code (no news there!). Analysis showed that part of Groov's payload tries to save some network configuration information into a file. The method used should only work properly under *Windows 98*, and maybe *NT 5.0*, when it is released.

The mechanism used fails under *Windows 95*, but that does not prevent the next step of the payload running. That step is an attempt to upload the file that should have been

created in the first step to the FTP site of *FRISK Software* (makers of *F-PROT*). This was implemented through the simple means of creating script and batch files to run the standard *Windows 9x/NT* FTP program.

What was the virus writer's point? Electronic bragging? Network tagging? Does it really matter? This virus shows that it is easy to transport a file from an Internet-connected host machine (the 'target') to any other. Groov is not elegant, but proofs of concept are often not – a few lines of simple error-checking code would have made Groov much less prone to giving itself away.

Ten days after receiving Groov, *VB* received news of a new macro virus, W97M/PolyPoster (also know as Agent). Apart from being a mildly (and slow) polymorphic *Word 97* virus, PolyPoster also had a network-related payload. *Agent* and its freeware counterpart *Free Agent* are very popular *Windows* Usenet News clients. W97M/PolyPoster looks for an *Agent* installation on the host machine. If found, the virus drops copies of infected documents into Agent's outgoing message queue, and next time Agent syncs up with a News server, those documents will be posted for the world at large to see.

… Nor Third-party Programs!

All of the 'network exploits' discussed so far depend on the host computers having copies of additional network applications installed – *mIRC*, *MS Mail*, *Eudora*, *Agent*, etc. Although having another application doing their network bidding eases the virus writers' development efforts, it also limits them to functionality that is already implemented. If widespread distribution is their ultimate aim, they are limited even further to those very commonly used applications and protocols.

Chronologically, the network-aware viruses that followed the *mIRC* SCRIPT.INI business were the variants of the Semisoft family. The author of these large viruses was clearly not content to be limited to existing protocols. The Semisoft variants set up listening processes on their host machines and attempt to connect to several other machines. Not all the details of the networking code are understood, but it appears that Semisoft listens for instructions from those machines and may send files and other information from the host to them.

There is much more to the Semisoft family, but this aspect of remote control may be its most interesting feature. Although some Semisoft variants have been seen in the wild, the virus' size probably works against it having much chance of spreading successfully.

Of course, mention of remote control via the network raises the spectre of Back Orifice and NetBus. Although not viruses, these network backdoors are widely agreed as worthy of detecting as Trojan Horses. This is because of the surreptitious manner in which they install themselves and hide their presence from the users of their host computers.

Another piece of non-viral malware with its own network code is the IE090898.EXE Trojan (mentioned in September's Editorial). It avoids depending on any particular email program by implementing its own SMTP client code. This allows it to deliver its email 'payload' directly, so long as the host has an active Internet connection.

So Where Now?

Imagine a program (you would call it a Trojan Horse) that searches through the files on your computer, finds all those containing the phrase 'company confidential' and then sends them off to some anonymous re-mailer address. On reflection, you may feel you are safe – such a program would never get past your stringent security policy of 'only IT vetted and installed programs', would it?

The bad news is that this can all be done from a macro in one of the *Office* products. It may or may not be a viral macro – an unannounced or Trojan Horse macro would do just as well. If you have an email gateway scanner, it would not catch the rogue program if it were new. Furthermore, if it did not use 'typically viral' functions (which it would not have to) it would also be unlikely to trigger any scanners that include heuristic macro analysers.

Please do not worry too much – it has not happened… yet!

However, as shown above, the components of something like this have been developed. There are also, many other easily-spoofed email and News clients, not to mention the other network protocols that could be utilized.

Imagine one of your corporate memos or reports being 'leaked' to the whole world of Usenet or some wide-distribution mailing list. [*Or to your competitors… Ed.*]

Apart from the business implications of anything sensitive in that document, what does the fact that your company was hit by a simple virus or Trojan (and these things usually are simple!) say to your corporate partners? As you would have prior to this incident, they will judge you in light of their belief that this kind of thing only happens to others – the less reliable, less trustworthy…

Is there a solution? Removing all forms of external access to electronic data is not a viable option, but executable content may be able to be blocked at the network perimeter. If the rogue program is missed there, it seems reasonable to expect a *secure* mechanism for *preventing* document-borne macros from running. The anti-virus industry has, thus far, failed to convince *Microsoft* that providing such an option is is a good thing. Maybe those who fund development of *Microsoft* products should now make a stand?

The time to influence the next generation is running out. *Office 2000* approaches completion. Release of the second wide-area beta is imminent (if not overdue). If you are concerned about the ease with which *Word*, *Excel* and friends allow system integrity to be breached, make your feelings felt to *Microsoft* now.

PRODUCT REVIEW 1

Sophos Anti-Virus v 3.13 for Windows 98

As the foam dissipates in the wake of *NAI's* assimilation of *Dr Solomon's*, the companies seeking to take advantage of this murky chaos include *Sophos*. A strong contender in the UK, this long-time competitor of *Dr Solomon's* will also be looking to expand, claiming its independence from the whims of shareholders as a selling point. *Virus Bulletin* leaves these matters for the reader to judge – the mechanics and implementation of *Sophos*' anti-virus package are much more the chosen field of scrutiny.

Sophos Anti-Virus (SAV) is available for a wider collection of machines than many other products on offer. DOS, *NetWare*, *NT* on *Alpha* and *Intel*, *OpenVMS*, *OS/2*, *Banyan Vines* and *Windows 9x* are all currently supported, with a host of Unix variants in preparation. *VB* last tested *SWEEP* on *NT Server*, so the *Windows 95/98* workstation version is evaluated here. One of *Sophos*' claims is that all its products achieve identical detection results – not a claim to be tested here. Similarly, results of on-access and on-demand scanning should be identical, and this was more easily subjected to examination.

This is also the first evaluation on the *Windows 98* platform which might be expected to bring with it new problems in the production of anti-virus software. This fact was to some degree responsible for a *Windows 98* review prior to the comparative review next month, where twenty-two products will stand the trials allotted them. *Sophos*' products have always been stable to the point of indestructibility – were this to change under *Windows 98* it would be a bad sign for reviewer sanity.

The Package

Hitherto known as *SWEEP*, *SAV* has undergone both a name change and a major presentation overhaul during the last six months or so. *SWEEP* lives on as the name for the on-demand component of the package, with *InterCheck* the on-access one. With the product hand delivered and the test of the *British Post Office* thus denied, sturdiness was tested by throwing the box at hard surfaces for a while – the packaging thus gaining first honours in the resilience category. The all-new artwork is devoid of any one overwhelming colour, or portraits of the *Sophos* proprietors in appealing guise, thus avoiding amusing remarks on that front.

Inside this admirable protection is the usual collection of paper proffered by anti-virus companies, with a couple of novelties unique to *SAV*. A 16-page Quick Start Guide is provided for the main product, backed up by a 130-page *Windows 95* User Guide, and a 190-page *DOS/Windows 3.x* guide. This raises two points. Firstly, the package was submitted as fully compatible with *Windows 98*, though not yet labelled as such – *Sophos* will not be relishing a revamp to change mere words when the product is identical for the *Windows 9x* family. Secondly, the package includes in its licence the whole of the DOS version of *SWEEP*, so that scans may be made after a clean boot into DOS. To this end, two media are included in the package – a CD and a permanently write-protected *SWEEP* floppy disk. The contents of these are discussed later.

Stickers are provided for use as warnings for known infected disks, and a further collection for disks scanned and found clean. To this is added a mouse mat of interesting design, having the property that it looks far more impressive from a distance of several yards than at arm's length. Finally, the *Sophos* Reference Guide 1998/1999 takes up a good-sized chunk of box space. This last is more tome-like than the usual offerings – 370 pages of assorted information covering various aspects of data security, viruses and the complete *Sophos* product range.

Surprisingly, the products are described in terms of technical function rather than the familiar, self-congratulatory voice of the marketeers. *Sophos* has an odd attitude to losses incurred by its software. The licence includes the usual actions for which *Sophos* will accept no liability, yet specifically states that in the event of death or serious injury due to the use of *SAV*, it will accept responsibility.

The Disks

The Emergency *SWEEP* disk contains but two objects, *SWEEP* itself and an instructions file. The Software Master Copy CD is, by comparison, possessed of a vast array of files, most divided amongst the various products represented upon the CD. *Sophos* has opted for the trusting approach that although licensed for perhaps only one

platform a user should have all versions of *SAV* available for evaluation. The CD thus contains the full *Sophos Anti-Virus* range, plus various associated add-ons. Such a volume of material could be a little daunting, but an autorun menu makes an effort to alleviate things.

The menu is divided into Quick Install, Advanced Options, What's New, Documentation, Explore CD and Exit. The first two cover the installation procedure, discussed later. Explore CD and Exit are self-explanatory, and resulted in no surprises when used.

Installation

Installation can follow two paths – Quick Install or the Advanced option. The former was investigated first. The setup program detected the correct platform, and gave a radio-button choice between a local or central install or upgrade. Local was chosen for this test, and the check box for installation of *InterCheck* was also selected as the default. Source and destination directories were offered and accepted. Final approval of the selected settings started the installation and took less than a minute.

With *InterCheck* selected there followed a pre-scan for that program. *InterCheck* uses a method of on-access scanning *VB* expects will become more common. The theory is that once a file has been scanned and declared clean, it will be clean unless it changes. Change is detected by check-summing, which is faster than *SWEEP's* virus scanning. Thus, *InterCheck* stores a checksum file, obviating the need for repeated scanning of files. Clearly, this list of authorized programs cannot stand forever and is spawned anew each time a fresh installation or *SAV* update is completed.

InterCheck thus starts slowly upon its first use, with the current version launching a DOS box for creation of the initial checksum (an alternative version on the CD does much the same but in a less antiquated-looking fashion). This process took between thirty seconds and five minutes on machines ranging from a bare installation of *Windows 98* to those having a gigabyte of typical desktop applications, but is much slower if large volumes of OLE files (*Word*, *Excel*, *PowerPoint*, *PageMaker*, etc) are present.

The advanced options section of the CD menu includes the installation of *Adobe Acrobat* and a facility for the production of disk sets for the installation of *SAV* on machines lacking a CD-ROM drive. The standard medium used for the distribution of *SAV* is the CD-ROM, though floppies are available upon request. The third advanced option is the installation expert system. This proposes a series of questions concerning network availability, operating system and desired client-server relationships leading up to a decision as to which files must be installed.

At the end of this interrogation a single mouse click launches the appropriate installation. The only problem encountered here was in the naming of the operating systems – *Windows 98* was not included. This niggle continued into the installation however it was performed, the program being consistently labelled as 'SWEEP for Windows 95'. While not a major problem, it is an easily remedied cosmetic issue.

Documentation

With two manuals and a Quick Start Guide, there is certainly no lack of documentation. The Quick Start Guide, making use of autorun, does indeed allow a single user set-up in just half a dozen mouse clicks. A short overview of *SAV's* operation is also included, sufficient for the use of the programs without an understanding of their full set of features. The two main manuals cover operation of both the DOS scanner and *Windows 9x* implementation respectively, though the latter refers throughout to *Windows 95*.

SAV's InterCheck on-access scanner can be configured as a local or central installation, local being chosen here, as befits a standalone review. For this reason large parts of the manual are not applicable, which is more confusing than if the manuals had been split in two. This is particularly relevant to the options available within the *InterCheck* configuration files, where many do not apply to the stand-alone setting. That said, the degree of detail is certainly up to the expected standard, and a network administrator should have no problems implementing any number of options which do not appear in the standard interface.

Manuals are also provided on the CD in PDF format – *Adobe Acrobat Reader* is also supplied. The option to install *Reader* is given upon selecting the Documentation part of the CD menu, needed since the index to the documents is also in PDF format. The documentation includes the Reference Guide, Quick Start and Standard manuals for all currently supported platforms of *SAV*, and a smattering of *Sophos*-related technical papers.

Web Presence and Support

The *Sophos* Web site has also undergone a revamp, in line with that of the packaging, and features the animals from recent marketing campaigns. The standard features of an anti-virus Web site are all present, company profile, virus descriptions and evaluation downloads to name but a few. The collection of FAQs is particularly extensive, and covers the entire range of *Sophos* products. There is also the option to contact the technical support department of *Sophos* directly, so that questions not covered elsewhere may be answered. This is in addition to the standard *Sophos* telephone support lines which operate continuously and are included in the licence cost.

Updates for one year are included in the initial licence. These updates are produced on a strict monthly rota, with smaller updates for specific, newly discovered viruses available from the Web site. *Sophos* maintains a mailing list for notification of urgent virus issues, and in the event of an 'emergency' – CIH being counted in this category – floppy disks with updates are sent to all licence holders.

Fixtures and Fittings

Configuration of *InterCheck* is performed not from the user interface but from the *InterCheck* Configuration File. This sets a large number of parameters within *InterCheck*, though many, such as polling for software updates at startup, are of more concern in a network situation – the scenario *Sophos* targets in their design philosophy. There are no contents to this file in a standard installation, only nonstandard options require an entry.

The new GUI-based version of *InterCheck*, due for release in the next month but included on the CD as an optional installation, has an extended list of relevant options. This includes the on-the-fly disinfection of OLE files (previously a task reserved for *SWEEP*), which is activated by a single line within the configuration file. The downside of this is that there is no built-in editor or menu-driven method for producing these files – manual editing of the file being the only method supported. *Sophos* support staff used many expletives when asked their opinion of this state of affairs. *SAV* also comes with a selection of distribution tools which are less well-documented than the rest of the package. These are of more relevance in a corporate environment, and thus not covered in this review.

The configuration of *SWEEP* consists of toolbars on the main *SWEEP* GUI. From here scheduled scans may be prepared, as may spur-of-the-moment scanning. Four areas for configuration are available from these menus; File, Options, View and Help.

The File menu allows the instigation or cancellation of a *SWEEP* of files or memory, the setting of a log folder and exiting the program. The start and halt scanning operations are also available through large push buttons in the main GUI, though, irritatingly, these are not available through simple keyboard shortcuts – of particular frustration during bulk scanning of the diskettes. The Options menu is the main area where major changes are effected. From here, and further sub-menus, you can alter the level of scanning, and select which files are to be scanned by default. Scanned extensions may be changed globally, though the choice to scan all files is made for each scan. It is possible to exclude certain files from the scan, but not particular folders.

The main *SWEEP* dialog allows scans to be started immediately and access to the main configuration options.

Automatic disinfection may be selected for OLE files or boot viruses, though *Sophos* chooses not to disinfect COM/EXE file viruses through *SWEEP*, suggesting that restoring from backups is a better option. Notification of infection may be emailed to selected personnel, with log files also an option. Book-keeping is aided by options to restore configuration to defaults, and clear log files.

Individual scans are produced simply, and may consist of drives, directories or individual files. By combining them into groups, a wide variety of scans may be produced. The View menu options include display of a progress bar during the scanning process, and it is also possible to include or exclude all file names scanned, from both the report file and the on-screen report. For the more inquisitive soul, the virus library details the viruses known to the *Sophos* engine.

Scheduled scans may be prepared for a very wide selection of times, days being selectable independently, and with the configuration options noted for *SWEEP* above also available for the scheduled scans. Each scan may be configured to produce a separate report file in a nominated location. By default, a scheduled scan is set up every day to be run at one o'clock – presumably the *Sophos* assumption of universal lunchtime. The last of the menus, Help, is self-explanatory and due to the self-evident nature of most functions, largely redundant.

Detection

The *VB* test-set was examined on-demand using *SWEEP* in three different configurations, namely Quick scan, Full scan and Full scan all files. The first of these is the default setting. *InterCheck* was tested in its 'on' position – the only option available, results for which should, if *Sophos*' claims are to be believed, be the same as the default settings for *SWEEP*. Boot sector viruses were scanned only using the default settings for *InterCheck* and *SWEEP*.

As has become *VB* tradition for *SAV*, detection was at one hundred percent for the ItW boot sector viruses both on-access and on-demand. The same proved true against the ItW File and Polymorphic test-sets, where both *InterCheck* and *SWEEP* were able to detect all samples, regardless of settings used. The Polymorphic set includes a large number of Marburg samples the detection of which, in the light of recent wide distribution of this virus, is no doubt a pleasing result for the *Sophos* virus research team.

The missed samples were from the Standard and Macro test-sets, where results show a variation between scanning modes. With the most rigorous scan, on-demand with all files scanned and a Full scan of each, *SWEEP* missed samples of only three viruses, Class.A, Class.C and Pwd.A.

All the samples of Class were missed, a polymorphic *Word 97* macro virus. This novel method clearly caused some problems for the *SAV* engine, though *Sophos* claims that its most recent engine is able to detect this virus family. Pwd.A was detected in its template form, but not in the encrypted documents which it can produce. It appears that *Sophos* has decided, for whatever reason, not to crack this decryption but to leave such files effectively unscanned, though there will be some warnings and protection given as the template triggers *InterCheck*.

The next less rigorous scan was the Full mode where *SWEEP* chooses the files to scan. This resulted in the same misses as before, with the addition of AccessiV.A and AccessiV.B. These were missed due to the lack of the MDB extension in the default list of executable files scanned by *SWEEP*. This lack might be understandable on speed grounds, but is an odd omission when the ability to scan for the virus has been provided.

The least secure scan mode, Standard, was tested on both *SWEEP* and *InterCheck*. In the past *Sophos* has managed to keep a record of total agreement between their *SWEEP* and *InterCheck* scans, easily done since the scanning is done by the same engine for both, differing only in the source of requests to scan a document. In these tests, however, differences were noted.

In addition to those already mentioned, *SWEEP* missed the VxD samples of Navrhar and, as ever, the samples of Positron. The latter are mid-infectors, and without specific hacks are not detectable with the entry-point tracing used by a standard *SAV* scan. Surprisingly, in addition to these the *InterCheck* scan missed the DOC versions of Navrhar. These samples should be detectable by the same method used by *SWEEP*, and it can only be supposed that some internal glitch is causing this strange phenomenon.

Time Trials

It was a noticeable feature of scanning under *Windows 98* that the process was slower than on the same machine under *Windows 95*, albeit with an earlier version of *SAV*. Polymorphic viruses appeared to be unaffected but macro viruses slowed considerably. Such subjective observations, however, were possibly of more relevance to the operating system's behaviour than that of *SAV*.

It was thus decided to compare speeds of the *SAV* version tested under *Windows 95* and *98* in a more objective manner. Sources in the *Sophos* team acknowledged problems were known with the *Windows 98* smooth scroll, which is not multi-threaded, thus causing *SAV* to wait until scrolling has completed before scanning is resumed. As *SWEEP* uses a scrolling list for its on-screen report, this could account for the slow-down. Timed scans of the Clean test-set should obviate the scrolling problem, as no viruses should be reported. However, a test where scrolling was required was also run. Since the progress bar is optional, speed tests were also performed with and without this.

The results of these time tests showed that first impressions can be erroneous. The standard default scan of the Clean test-set took 192 seconds, compared to 329 for the same scan under *Windows 95*. Using a Full rather than a Quick scan increased this timing to 576 seconds for the 5500 files in the *VB* Clean test-set. In neither of these scans were any false positives produced.

It was the addition of file name listing as scanning progressed, however, that produced the largest change in timings. This change necessitated the introduction of scrolling to the status window, and the time taken increased to 1540 seconds. This is not so much a fault in *SWEEP*, as a glaring problem for *Windows 98*. The smooth scroll option is global, and cannot be disabled for one window. This means all applications are stuck with whichever setting the user has selected. Perhaps unforunately, the default setting is having smooth scroll on, so all products using scrolling will be adversely affected.

Diskette scanning speed was checked with a clean and dirty disk, identical other than an infection of Natas.4744 in every file on the latter. The dirty disk took 42 seconds to scan, the clean one just less at 40 seconds. Since the alerts in the dirty disk were triggering the smooth scroll in the results window, this is perhaps a result where the future will see small speed increases. *InterCheck's* overhead was measured using the standard *VB* XCOPY test. An overhead of some ten percent was observed.

Conclusion

Not unexpectedly, with a good record over the past few comparative reviews, *SAV* manages a good detection rate, missing a few exotic viruses whose detection is possible at the cost of significantly slowing performance, and more importantly missing the polymorphic Class variants. *Sophos* claims that the Class issue has been addressed in the latest version of its engine. It is also notable in this review not so much what *SAV* can do, but what it intends to do and was not tested. *SAV* is primarily designed as a network solution with many tools and options for its deployment in complex networks. Despite this, *SAV* acquits itself admirably in a standalone setting.

Technical Details

Product: *Sophos Anti-Virus for Windows 95/98*.

Developer: *Sophos Plc*, The Pentagon, Abingdon Science Park, Abingdon, Oxford, OX14 3YP, UK; Tel +44 1235 559933, fax +44 1235 559935, WWW http://www.sophos.com/.

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

Version Evaluated: 3.13 for *Windows 98*.

Price: Single licence £99. For multiple and site licence prices, please contact the developers.

Hardware Used: 166MHz Pentium-MMX workstation with 64 RAM, 4 GB hard disk, CD-ROM drive and 3.5-inch floppy drive running *Windows 98*.

Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/199809.html.

PRODUCT REVIEW 2

F-Secure Anti-Virus Macro Control

Data Fellows F-Secure Anti-Virus (*FSAV*), formerly *F-PROT*, has been a well-respected anti-virus solution for some time, its macro detection particularly well considered. It may seem a little perplexing then for *Data Fellows* to release another product aimed directly at the macro virus problem and a little self-defeating that there is currently no charge for *F-Secure Anti-Virus Macro Control* (*FSAV MC*).

However, *FSAV* is a scanner, while *FSAV MC* is more concerned with the authentication of existing macros, with the capacity to detect any macro viruses as a part of the procedure. A similar approach, utilizing checksumming of individual, approved macros is included in the recently reviewed *Defuse Enterprise* (see *VB*, July 1998, p.18).

FSAV MC has a three-pronged approach to preventing the use of viral macros. Since legitimate macros are used in common packages (*Word*, for example), there is a list of known, safe 'certified' macros which are authorized for use by default. There is also a list of known viral macros, which are banned at all times. All other macros are assumed to be dangerous and not to be accessed, unless the administrator specifically flags them as approved. Upon the discovery of an unauthorized or known viral macro, there are various options available at the administrator's discretion.

As mentioned relatively late in the manual, *FSAV MC* is a *Microsoft Word*-specific program, and as such will not provide protection against macro viruses in other *Office* applications. *Word 6* and *Office 97* are mentioned as being *Word* file formats, yet nowhere is there a statement as to which file formats are actually supported. Similarly, despite no evidence elsewhere on the matter, the manual states that *FSAV MC* requires *Windows NT 4.0*, though no service pack requirements were noted. More precision would be appreciated here, but it also provided an opportunity to test compatibility with other versions of *Word*, while *Office 97* was the default used.

Packaging and Documentation

FSAV MC was received as two files, a PDF of 178 KB providing documentation, and an executable of 2 MB. These are freely available from the *Data Fellows* Web site, together with the two varieties of update required for optimum performance of the program.

The manual claims *FSAV MC* is an 'Industry-Strength Macro Certification' system. This inspired apprehension that the language within might be coloured with quirky translational oddities, a fear which was, thankfully, unfounded. It has already been noted that the manual is vague on the subject of system requirements for the application, though it is more useful as a reference guide.

Fifty pages in length, it begins with a general overview of the program's function, and a Quick Start Guide. Further chapters discuss the use of the program, its configuration and upkeep using the administration tools provided. Distribution is also discussed and again there are tools provided for this. A glossary, index and brief overview of macro viruses complete the offering. The descriptions, examples and instructions are all of a clear and well-phrased nature, leaving no great unanswered questions.

The manual is aimed at the administrator. Since the administrator has an active role in the implementation of the system, and differences between his and the users' experiences are likely to be major, a separate user guide might have been a handy addition. It is also notable that there are rather more typos, mainly added spaces, in the manual than is common in *Data Fellows*' usual offerings. The new and gratis nature of the product might account for this.

Installation

The program was run on *NT 4.0 (SP3)*. *InstallShield* launched with a description of what would be installed. If this was approved the utility then installed the *F-Secure Manager*. For this application, required by *FSAV MC*, a shared administration directory must be created. Unusually, no default was suggested. An administrator password must also be chosen now, which in the current version could not be changed at any time subsequent to installation. Following installation, viewing the readme file or launching the manager were offered as options.

The readme file turned out to be edifying on a number of accounts. The need for *NT 4* was reaffirmed, and further defined as requiring *SP1* or *SP3*. A list of known problems was included – the inability to scan embedded documents, the unalterable password, the lack of description tagging for macros and the possible need to provide a domain when creating the shared administration directory. These will probably be addressed in future versions of the program.

As trichoschismatics of long standing *VB* installed the program without having a copy of *Word* already installed on the test machine, hoping to cause some grievous problems. Surprisingly, the entire package seemed to install perfectly, despite the lack of this seemingly vital component. What is more, the prevention of access for unauthorized macros was in place. This was the case during the subsequent installation of *Word 7*, which, as might be expected, installs a large number of macros and wizards. *FSAV MC* declared a number of these to be unauthorized, and blocked their installation on the machine.

The evidence suggests that interception of macros is performed as soon as the document is recognized as a *Word* one, detection occurring during file operations. This will be an advantage in cases where more than one version or copy of *Word* is in use on the same machine, since only one installation of *FSAV MC* is required.

The Interface

FSAV MC shares the same management program as other *Data Fellows* programs – the *F-Secure Manager*. In the configuration as tested, with only one *Data Fellows* product installed, the requirement to pass through the manager was relatively irksome as just another part of the program to step through before configurations could be changed. In the event that another *Data Fellows* product were installed, it prevented a cluttering of the system tray with a whole host of *Data Fellows*' purple triangles. This is one of the points where administration rights are checked in a multi-user environment. There is no real interface for general users, just messages stating that events have occurred.

Administrative Tools

Administration may be performed from one of two places, the individual workstation or the central installation point, with the latter having a wider range of control. The locations are, by default, linked by the workstation installations polling the server at intervals. Quite what these intervals are was not apparent, though they are certainly not frequent enough to cause any significant network load. As stated, this is a default setting, and workstations may be configured so as never to poll for settings, effectively locking them into a set configuration, regardless of administrator status. The administrator may also be set to poll for settings, useful when at least one workstation is set without the poll option, and allowing what amounts to remote administration.

Workstation settings may be reviewed using the toolbar manager icon, though alteration requires the use of the administrator password. Selected here are the action options on detection of an unauthorized macro. Any or none of the following may be chosen – send to the administrator, remove all macros from the document and create a backup of the document.

Removing all macros is the only method allowing the user to edit a suspect document, the unauthorized macro otherwise barring access. Also selected here are the files to check – all files or a list of likely document extensions. Included in the default list are the generic Wizard extension WIZ and the document extensions RTF and TXT, produced by some viruses in OLE documents to attempt circumvention of checking by standard default extension lists. A list of statistics for known macros is the remaining content of this window, with the poll-for settings box and administrator entry point completing the line-up.

The central administration program includes the window described above, tabbed for access as an alternative to the main macro administration area. Here macros may be certified for use in a number of ways, either in bulk from documents or disk areas containing documents, or individually. The Distribute option triggers the Autoinst Wizard, used to install the required parts of the application across a network with minimum administrator stress. Use of this Wizard allows installation to be triggered by a one line addition to network login scripts, or a one-time direct invocation of the installation routine at each machine, after which all is automatic and unattended.

Unusually, and admirably, all the functions available in this set of controls have keyboard shortcuts and even better, these are actually listed in the manual. This is certainly an area where other companies should take note.

Upgrades and Web Presence

Upgrades to the pre-authorized 'certified' macro list and the viral macro list are available from the *Data Fellows* Web site. The latest of these was downloaded and proved to be sizeable at 125 KB, incremental updating not being the order of the day. These updates must be installed manually into the Administrator installation, but from there are distributed using the Autoinst Wizard, depending on the poll setting of workstations. The provision of other means of support on the Web site is as yet minimal – perhaps due to lack of demand and the product's gratis nature.

Operation

Since documents must be opened individually for absolute checking by this kind of product, the standard *VB* testing methods become unmanageable as far as time taken to scan the *VB* macro virus collection is concerned. The operation of the program is also more dependent upon the macro authorization code than the virus detection, thus the tests of virus detection ability were more empirical than usual.

The actions for testing documents were checked by attempting various file operations on a document containing unauthorized macro code. The documentation did state access to be the trigger for testing, which turned out not to be the only activity where detection could take place – copying files also gave a trigger.

Overheads were tested by opening a number of authorized macro-bearing documents, with and without *FSAV MC* having been installed. These documents were standard *Microsoft*-supplied templates, from the *Office 97* templates directories – 25 WIZ DOT and DOC files totalling 3 MB.

They were all opened in seven seconds without *FSAV MC* loaded, rising to 74 seconds with checking enabled, definitely not a speed which will thrill.

The macro viruses from the July WildList were subjected to attempted approval with *FSAV MC*, as a check of virus recognition. The results were very disappointing. Most of the ItW macro viruses were accepted for authorization by *FSAV MC* which did not trigger any alarm, even on such antique viruses as WM/Concept.A. It seems therefore that the virus detection ability of *FSAV MC* is very unreliable.

Class.A and variants were particularly studied since they use class objects rather than standard macros, it being possible that this might hinder detection. These are also more problematical for pattern scanners by dint of being polymorphic. Despite being able to be approved, the presence of potential hazards was noted and thus Class infected documents could not be opened blindly.

Options for the treatment of unapproved macros were tested. Sending the document to the administrator worked as expected, but the removal of macros option was less effective. This caused an automatic exception error, terminating the manager's presence on the start bar, but thankfully still denying access to the suspicious file.

Conclusion

Generic virus detection seems more likely to succeed with macro viruses due to their less varied infection methods. The operation of *FSAV MC* is exemplary as a detector of macros, and can handle the blocking of unapproved macros with ease. However, the virus detection capability of the product as it stands is woeful, and may lull the less wary into a false sense of security rather than aid them.

As yet, a standalone virus scanner such as that produced by *Data Fellows* might be considered a vital adjunct to *FSAV MC*. Likewise, the actions on detection work well enough when denial and redirection are selected, but removal of macros is fatally unstable, there being no real solution but to avoid the flawed operation. That said, the program is free of charge, and clearly at an early stage of development. It will be interesting to revisit, and even as it stands would be useful for those with specific needs where *FSAV MC* can already deliver.

Technical Details

Product: *F-Secure Anti-Virus Macro Control*

Developer: *Data Fellows Ltd*, Pl 24, FIN 02231, Espoo, Finland; Tel +358 9859900, fax +358 985990599, WWW http://www.datafellows.com/.

Availability: .

Version Evaluated: 1.0.

Price: Non-commercial use, free. Commercial users: up to 50 licences, free; more than 50 licences, prices on application.

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

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

There is still time to register for the eighth annual *Virus Bulletin* conference and exhibition. **VB'98 takes place from 22–23 October at the Munich Park Hilton, Munich, Germany.** For more information, or to reserve your place, contact Jo Peck; Tel +44 1235 555139, or email Joanne.Peck@virusbtn.com.

Infosecurity Scotland '98 encompasses every aspect of IT security from hacker-proof Internet and Intranet systems to business continuity solutions and anti-virus implementation. The exhibition hosts two free keynote speeches and a series of 13 seminars. Over fifty international IT security companies will be exhibiting at Edinburgh's Royal Highland Centre from 28–29 October 1998; Tel +44 181 910 7790.

***Sophos* is hosting a practical *NetWare* security course** at its training suite in Abingdon, UK on 5 November 1998. The one-day, intensive course costs £325 +VAT. From 11–12 November, *Sophos* will also run a **live virus workshop**, which costs £595 +VAT. For details, contact Karen Richardson; Tel +44 1235 544015, fax +44 1235 559935 or visit the company's web site http://www.sophos.com/.

Compsec '98 takes 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. 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/.

***Network Associates* (formerly *Dr Solomon's*) is running a live virus workshop from 17–18 November 1998**, priced £695+VAT, at the Barns Hotel, Bedford, UK. For more details, contact Caroline Jordan; Tel +44 1296 318881 or email Caroline.Jordan@nai.com.

***Central Command*, the US and Canada distributors of *AVP*, announces bimonthly advanced computer virus workshops**, starting in November, aimed at System Administrators. The classes will be held at the company's corporate headquarters in Brunswick, Ohio. Class size is limited to 25, and the cost for the three-day workshop is $1695. For more information, contact Renée Barnhardt; Tel +1 330 273 2820 or email renee@avp.com.

The 25th Annual Computer Security Conference takes place at the Chicago Hilton & Towers, Chicago, USA, from 2–4 November 1998. The twelve track conference is preceded and followed by two-day seminars and *Computer Security Institute* members are eligible for a $100 saving on the conference fee. The affiliated exhibition runs from 1–3 November. For more details contact *CSI*; Tel +1 415 356 3371, fax +1 415 905 2218, or visit http://www.gocsi.com/.

Following the launch of the **British security standard BS7799**, as featured in this column in the August issue of *VB*, a new accredited certification scheme has been devised to evaluate products wishing to carry the mark. Independent auditors for the **c:cure Scheme** investigate ten key controls in their evaluation, from reports of security incidents through virus controls to data protection. For more details, contact the c:cure Scheme Manager in London; Tel +44 181 995 7799, fax +44 181 996 6411, email c_cure@bsi.org.uk or access the Scheme's Web site http://www.c-cure.org/.

***MIS* is to host two security seminars at the Regency Hotel in London.** From 7–9 December, the *Web and Intranet Security and Audit* seminar covers all aspects of planning, installing and maintaining a secure Web presence, including the control of viruses and the security challenges of active content. *Building Firewalls to Protect Your Internet Connection* is from 10–11 December. To register for either seminar, contact Debbie Rosen; Tel +44 171 779 8944, fax +44 171 779 8293, or email misuk@misti.com.

The International Information Systems Security Certification Consortium (ISC)² is an independent, non-profit organization formed in mid-1989 in North America. Its sole charter is to develop and administer a certification program for information security professionals. Applicants are now being invited to sit this year's **Certified Information Systems Security Professional (CISSP) Certification Examination**. In order to apply candidates must subscribe to the *(ISC)²* Code of Ethics and have at least three years work experience in one or more of the ten test domains of the information systems security *Common Body of Knowledge (CBK)*, which include Computer Operations Security, Cryptography and Security Management Practices. An Examination Study Guide and preparatory seminar have been devised to assist CISSP candidates. For more information about the test, fax +1 508 845 2420.