ISSN 0956-9979

FEBRUARY 1999

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Conference call:** The call for papers for VB'99 goes out in this issue. That and details of this year's *Virus Bulletin* conference are on p.3.

• **Creating a monster?** Eugene Kaspersky ponders the evolutionary potential of Internet viruses. His feature article starts on p.14.

• **This is your life?** Righard Zwienenberg, *Norman's* Senior Research and Development Engineer, is under the spotlight in our occasional Insight series on p.8.

CONTENTS

EDITORIAL

NEWS

FEATURE

INSIGHT

VIRUS ANALYSES

FEATURE

PRODUCT REVIEWS

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England.**
Tel +44 1235 555139. /99/$0.00+2.50

EDITORIAL

The Shape of Things to Come?

The last year has seen quite a shakedown of the anti-virus developers, and this theme seems set to continue, at least through the first half of 1999. In no particular order…

Symantec spun a deal with *IBM*. As a result, *Symantec* supports *IBMAV* customers, *IBM* encourages its customers to change to *NAV* and *IBM's* anti-virus researchers keep working on their projects but for *Symantec's* benefit. *Symantec* is hoping it can pick cool, new technology (the still 'imminent' Immune System); *IBM* can stop worrying about failing to market an anti-virus product, and the cynics are still puzzled.

“ the future anti-virus market will have room for two large players…”

I cannot believe *Symantec* saw any solution to *NAV's* weaknesses in the corporate network arena with that deal – after all, it struck a similar deal a few months later with *Intel*, for part of its *LANDesk* network management suite to resolve that one. And speaking of *Intel*, it pretty much silently switched detection engines in its *LANDesk Virus Protect* range earlier in the year.

NAI bought *Dr Solomon's*, lock, stock and help-desk software. *VirusScan* could thus be enhanced with an historically top performing engine from a team that understood the high negative value of false-positives. *NAI* said it wanted *Solly's* European market-share; the Solomonites said it proved they were hurting *NAI's* market-share in their most recent foray into the US market; the trenchant techie in me says *NAI* realized it needed to buy some reasonable virus detection technology, as even *NAI* would not be able to hold its position for much longer on brand name alone.

Data Fellows, *FRISK Software* and *KAMI* (later to become *Kaspersky Lab*) made some interesting moves early on in this game. *Data Fellows'* approach with *F-Secure Anti-Virus* tying two engines together has not been adopted by other developers but provides an interesting solution to the question of how to use two products simultaneously but without them conflicting with each other.

Carmel has dropped out of the game. See 'Acquisitions and Closures' on p.3 for the scant details. This is probably no great loss apart from to those who had pre-paid service or upgrade contracts.

The purchase of *EliaShim/eSafe* by *Aladdin* is interesting insofar as I can see no significant motive for it or likely benefit to existing *eSafe* customers. This one I'm watching with interest.

Most recently, the *Computer Associates/Vet* deal also has some interesting implications. Nothing definitive has been said yet about the intended relationship between the existing detection engine used in the *CA* product line (from *iRiS*) and the newly acquired *Vet* technology. It is said the *Vet* product will continue under that name (probably only in the Australian or Oceania market?), but matters are still very unclear to those of us on the outside, at least.

Two or three of the remaining 'smaller names' in the industry are likely to disappear in the first half of 1999. In general, it seems that things are shaking out to fit the scenario Jimmy Kuo of *NAI* described at VB'98, when he said the likely future anti-virus market will have room for two large players in each major market region – the 'local favourite' and a major multi-national.

… and Changes at VB

After twenty issues of VB and what has seemed at times like half a life and at other times only half a year, I am hanging up the 'Do not Disturb' cap and setting aside the Editor's pencil. Francesca is taking over the day-to-day tasks of Editor while a technical consultant will soon be appointed to run product tests and comparative reviews and to maintain the test-sets and the like. I'd like to thank our readers who have been supportive (with plaudits, criticisms and responses in between) and especially Jessica, without whose understanding and encouragement this sojourn from New Zealand would not have been possible.

Catch you on the Net!

NEWS

VB'99 in the Pacific North West

VB'99 is to take place on Thursday 30 September and Friday 1 October, 1999 in Vancouver, British Columbia. Considered one of the most scenic cities in the world, Vancouver is an exciting conference destination with spectacular backdrops of coastal mountains and the Pacific ocean. The Hotel Vancouver, venue for VB'99, is the city's landmark hotel and has won several industry awards for its excellent facilities and services. Located in the heart of the downtown area, the hotel is within easy reach of the city's shopping and business districts and many of Vancouver's major attractions.

The conference will include presentations in both technical and corporate streams and a full product exhibition is to run concurrently. The traditional welcome drinks reception will be held on the evening of Wednesday 29 September and the gala dinner takes place on Thursday 30 September. A full partners' programme is planned.

We are currently seeking submissions for inclusion in the conference programme. Abstracts of about 200 words and an executive summary of 50 words, for inclusion in the conference brochure, must reach *Virus Bulletin* by Friday 26 February 1999. Please send your submissions to the Editor (ceskie@virusbtn.com or fax +44 1235 531889).

For details regarding booths at the exhibition or sponsorship opportunities; email Joanne.Peck@virusbtn.com or tel +44 1235 555139 ▮

Acquisitions and Closures

Following the extensive 'reorganization' of the anti-virus vendorscape that gripped much of 1998, the very end of the year and early 1999 brought further news on this front.

First, the Israeli developer *Carmel* was noted to have 'disappeared'. Its Internet domain was still registered, but attempts to contact the company and various of its employees failed. A reliable source informed *Virus Bulletin* that some of the programming and technical staff had moved to *Security 7* – a firm specializing in general computer and network security.

Carmel was the original developer of the detection engine used in the once-popular *Central Point Anti-Virus* (*CPAV*) product (purchased by *Symantec* in 1994). *CPAV* was subsequently used as the basis of the anti-virus utilities bundled with MS DOS v6.*x*. The 'disappearance' of *Carmel* leaves open the question of what support and upgrade options (if any) remain for licensees of its products.

In mid-December 1998, *Aladdin Knowledge Systems* (*AKS*) and *EliaShim* announced an agreement for the former to acquire the latter. *EliaShim* and its *eSafe* subsidiaries were

Prevalence Table – December 1998

Virus	Type	Incidents	Reports
Cap	Macro	327	30.0%
Class	Macro	236	21.7%
ColdApe	Macro	152	14.0%
Laroux	Macro	55	5.1%
Win95/CIH	File	43	3.9%
Temple	Macro	28	2.6%
Form	Boot	22	2.0%
Npad	Macro	22	2.0%
Brenda	Macro	21	1.9%
Munch	Macro	20	1.8%
Parity_Boot	Boot	19	1.7%
Hark	Macro	14	1.3%
Concept	Macro	13	1.2%
AntiEXE	Boot	12	1.1%
Appder	Macro	11	1.0%
Chack	Macro	8	0.7%
CopyCap	Macro	8	0.7%
Groov	Macro	7	0.6%
NoNo	Macro	6	0.6%
DelCMOS	Boot	4	0.4%
Ripper	Boot	4	0.4%
Suck	Macro	4	0.4%
Eco	Boot	3	0.3%
Marburg	File	3	0.3%
MDMA	Macro	3	0.3%
ShowOff	Macro	3	0.3%
Moloch	Boot	2	0.2%
Nutcracker	Multi-partite	2	0.2%
Quaint	Boot	2	0.2%
Ravage	Boot	2	0.2%
Stoned.Angelina	Boot	2	0.2%
TPVO.3783	Multi-partite	2	0.2%
Wazzu	Macro	2	0.2%
Win32/Cheval	File	2	0.2%
Others [1]		26	2.7%
Total		**1089**	**100%**

[1] The Prevalence Table includes one report of each of the following viruses: ABCD, AntiCMOS, Burglar.1150, Compat, Delwin, DiskWasher, DZT, Empire.Monkey, HLLC.Dosinfo, Inexist, Jerusalem.1363, Kenya, Komcon, Mental, Neuroquila, NF-B, Nottice, NYB, Paix, Polyposter, Rapi, Redemption, Russian_Flag, TPE and Yankee_Doodle.

valued at 1,240,000 *Aladdin* shares and US $6.5 million in cash. The announcement also says an additional amount, not exceeding US $5 million, may be payable based on *eSafe's* revenue performance in 1999.

The *eSafe* product line and staff will become part of *Aladdin's* Internet Security Unit. *Aladdin* is probably best known for its software security and licensing products (loosely, 'dongles' and associated licence management software) and is developing hardware tokens for storing passwords, private keys and electronic certificates.

In the middle of last month, *Computer Associates* (*CA*) announced the purchase of the assets of the privately-held Australian company, *Cybec Pty Ltd*. *Cybec* is the developer of the *Vet Anti-Virus* product range. The financial details of the deal have not been disclosed. Roger and Sally Riordan are retaining the *Cybec* name.

All *Cybec's* employees have been invited to join *CA* and the deal is being heralded as a friendly one. *Cybec* was the target of a more hostile 'approach' by *NAI* during 1998, according to an Australian newspaper report.

CA has been widely rumoured to be looking to buy an established anti-virus technology for some time now. Its current anti-virus offerings are based on virus engine technology developed by Israeli *iRiS Software*, and this relationship is over seven years old. There is little concrete information available about *CA's* development plans and *iRiS's* likely involvement in future developments once CA has integrated the *Vet* product and team ∎

Not So Happy?

Mid-January saw reports of a new virus that reproduces in a similar manner to Win95/Parvo (see *VB*, January 1999). Known as Win32/Ska, it seems more likely to succeed than Parvo, however, as it targets the email recipients and posted newsgroups of the user(s) of host machines.

It works by modifying WSOCK32.DLL so that two critical functions are redirected to Ska's code. Indications of infection are files called WSOCK32.SKA, SKA.EXE and SKA.DLL in the *Windows* system directory. *VB* plans to carry a detailed analysis in the near future. Ska was distributed in Usenet posts of a program called HAPPY99.EXE. That program produces a firework display when run.

Fortunately, messages generated by the virus have a telltale mark. Sites with email filtering capabilities may wish to quarantine messages with the *X-Spanska: Yes* header until their anti-virus software is updated to detect Ska ∎

New Word 97 Macro Vulnerability

On 21 January *Microsoft* released a patch for *Word 97* that addresses a newly discovered vulnerability. This hole has subsequently been exploited in a new virus and used to distribute another recently released virus.

Central to this vulnerability is *Word's* failure to alert the user to possibly unauthorized macros. It transpires that a *Word* document that does not contain any macros can be made to load a macro-containing template without raising any warnings. This may not sound serious if you assume the template and document must be distributed together – that would raise suspicions.

The real 'problem', however, is that the template can be referenced by URL. That means documents with no macros can load viral templates from anywhere on the Net so long as the host machine is connected. This will happen without the user being alerted to the existence of the macros in the remotely-sourced template.

A supported patch for *Word 97* is available on the web at http://officeupdate.microsoft.com/downloaddetails/wd97sp.exe. The patch requires the SR-1 version of *Word* (either from the full SR-1 release or the SR-1 patch). If you are planning to implement SR-2, do so *first*. Applying the SR-2 update after this security patch is applied to SR-1 removes the security patch! ∎

Et tu, Caligula!

Imagine a macro virus that steals PGP secret keyring files. You do not have to – it has been written and released. The virus, named W97M/Caligula by its author, simply checks the registry for the handler for files of class PGP Encrypted File, strips the directory name from it and searches that and all subdirectories for files matching the default filename of the *PGP v5.x* secret keyring. If that file is found, a scripted FTP session transfers the file to a collection point.

What was thought to be the initial release of only the source code may have been a leak by an acquaintance of the author. Regardless, it appears someone has taken that source and created an infected document, or the author distributed his work. It is to be hoped that neither Caligula, nor the ideas it encapsulates, become widespread ∎

Point of Order

The first three-application macro virus, O97M/Triplicate, was recently released. Following news of both the first *PowerPoint* macro viruses last month, and the first VBA/VBS cross-infectors the month before, it should not be surprising that the author of those creations was behind Triplicate's production.

Apart from being the first virus known to work on three *Office* platforms, Triplicate shows some other advances. In particular, it disables *PowerPoint's* macro virus protection, which has been a surprising omission from the earlier *PowerPoint* viruses. Given both how easily it is achieved and that plenty of examples of several methods are available in existing *Word* and *Excel* viruses, it is, however, surprising that Triplicate implements this in a manner that only works from an infected *Word* file ∎

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 January 1999. 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 with a user-updatable pattern library.

Type Codes

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

Accom.1280

CN: A 1280-byte prepender containing the texts 'GETCOMMSTATE', 'BOARDSTATE', 'AACOM', '*.com', 'URSORICONHAN', 'SETWINDOWPOS' and 'SETSYSTEMTI'.

`Aacom.1280    89C3 B440 8A2E 2004 8A0E 2104 BA00 05CD 21E8 D500 BF50 04CD`

Atest.300

CR: An appending, 300-byte virus containing the text 'Anarchy virus v1.0(test)'. Infected files have the byte 2Bh ('+') at offset 0003h.

`Atest.300    B8E0 83CD 213D A18E 7444 B92C 0183 2E02 0013 8CDB 4B8E C326`

Austin.1353

EN: A 1353-byte, direct infecting appender with the texts 'Erin is a stuck up snot whos full of herself cuz stone cold said so', '*********AUSTIN 3:16******** *hey Jack ass youve been * *infected with Austin 3:16 * *virus, why? cuz Stone Cold* *said so! You thought your * *computer was secure well * *you were wrong Jack Ass * *Austin 3:16 lives and * *replicates!! Anti Virus * *programs Suck jack ass! * *you should never trust * *AVers you jackass * *AUSTIN 3:16 says i just * *infected your ass! *****************************', 'command.com' and '*.exe'. Infected files have 5343h ('CS') at offset 0012h. The payload triggers on 16 March, and prints the messages.

`Austin.1353    E87D 01B4 40B9 4905 8D96 0001 CD21 B800 4233 C999 CD21 B440`

Code.336

CN: A 336-byte, direct infecting prepender with the text '*.com'. Infected files start with EDE8h.

`Code.336    B440 8B1E 0A02 B950 01BA 50FD CD21 B43E CD21 B801 438A 0E15`

Dementia.4229

CER: An encrypted, appending, 4229-byte virus containing the texts '!#TEMP#!', 'REQUEST.IVA', 'RECEIPT.IVA ', 'CALLFAST.COM', '*.*' and 'Dementia] Copyright 1993 Necrosoft enterprises - All rights reserved I am the man that walks alone And when I'm walking a dark road At night or strolling through the park When the light begins to change I sometimes feel a little strange A little anxious when it's dark'. Infected files have their time-stamps set to two seconds.

`Dementia.4229    5E81 C680 108B FEFD B933 08BA ???? 0E0E 1F07 AD33 C2AB EB02`

Dg.386

CN: An appending, 386-byte, direct infector with the texts '*.com' and 'DG..'. Due to a bug, the virus reinfects already infected files.

`DG.386    B440 BA05 0103 D68B 9C5C 028B 8C85 02CD 21B4 3E8B 9C5C 02CD`

Die.488

CR: An appending, 488-byte virus containing the encrypted text 'VIRUS INFO. Name : Transformer. Vers : 1.01 Model : RC-487 Danger : 0 Stealth factor : 0 Creator : Light General (30.08.94)'. Infected files have the word 2424h ('$$') at offset 0003h.

`Die.488    B440 B9E8 0133 D2CD 2172 1126 8955 15B4 40B9 0500 BA5A 01CD`

Djin.167

CER: An overwriting, 167-byte virus with the text '[DjiN_DjiN] iS MaDe By THe GaBBeR aNd Is DaNGeRouS NoW'. Infected files have their date and time-stamps set to contain all zeros.

`Djin.167    B800 4233 C999 CD21 B9A7 00B4 40BA 0001 CD21 9933 C933 D2B8`

EU.353

EN: An encrypted, appending, 353-byte direct infector containing the text '*.EXE'. Infected files have the word 5545h ('EU') at offset 0012h.

`EU.353    CD16 C3E8 1400 EB24 E80F 00B4 40B9 6101 8BD5 CD21 E803 00C3`

Evasor.145

CN: An overwriting, 145-byte, direct infector containing the texts '*.c*', and 'Evasor v1.0 Pruslas [Los Sicarios de Midas]'.

`Evasor.145    B03E E815 00B9 9100 BA00 01CD 21B4 3ECD 21B0 4DE8 0400 EBC7`

Evasor.226

CN: An encrypted, overwriting, 226-byte, direct infector containing the texts '*.c?m' and 'Evasor v1.2 Pruslas [Los Sicarios de Midas]'.

`Evasor.226    E2F6 8DB6 3301 8BFE B9AF 00E8 0400 EB10 90BB ACFE C03E 3286`

Evasor.394

CN: An encrypted, appending, 394-byte, direct infector containing the texts '.c*' and 'Evasor v2.0 Pruslas [Los Sicarios de Midas]'.

`Evasor.394    ACF6 D0C0 C804 F6D8 3E32 863F 01F6 D8C0 C804 F6D0 AAE2 E9C3`

Evasor.426 — **CN:** An encrypted, appending, 426-byte, direct infector containing the texts ‘*.c*’ and ‘Evasor v2.1 Pruslas [Los Sicarios de Midas]’.

```
Evasor.426          B965 018D B648 018B FEE8 0300 EB22 90AC F6D0 FEC8 C0C8 04F6
```

Evasor.466 — **CN:** An encrypted, appending, 466-byte direct infector containing the texts ‘*.c*’ and ‘Evasor v2.2 Pruslas [Los Sicarios de Midas]’.

```
Evasor.466          B965 018D B670 018B FEE8 0300 EB4A 90AC F6D8 C0C8 04F6 D0C0
```

Gunia.836 — **EN:** An appending, 836-byte direct infector containing the encrypted texts ‘*.exe’, ‘c:\’, ‘RHIVAT’, ‘c:\gunia’ and ‘*.*’. Infected files have the word FFFEh at offset 000Ch.

```
Gunia.836           B440 B940 038B D583 EA05 CD21 E839 FFB8 0042 33C9 33D2 CD21
```

Heiko.2184 — **CER:** An appending, stealth, polymorphic, 2184-byte virus with the texts ‘“Lord of the Dance”;fuer Heiko, den ich fuer immer und immer liebe; (c) Wesley;rel.’, ‘at s7=87s10=255dt0190332332’, ‘chklist.ms’ and ‘at s0=1’. Infected files have their time-stamps set to two seconds. This template detects the virus in memory only.

```
Heiko.2184          AD01 D0AB E2FA 061F 5AB9 3408 B440 CC07 1F5E 5F73 03E9 81FD
```

Hypervisor.3120.C — **CER:** A stealth, appending, minor variant of the 3120-byte virus containing the encrypted texts ‘NET$BVAL.SYS’, ‘NET$OBJ.SYS’, ‘NET$PROP.SYS’, ‘NET$VAL.SYS’, ‘HYPERVISOR’, ‘SECURITY_EQUALS’, ‘SUPERVISOR’, ‘GROUPS_I’M_IN’, ‘PASSWORD’, ‘IDENTIFICATION’, ‘LOGIN_CONTROL’ and ‘HYPERVISORCOMEXE’.

```
Hypervisor.3120.C   3E8B A69A FEFB 061F 2EFF AE9C FE33 C08E D883 2E13 0404 9058
```

Lauren.615 — **CN:** An encrypted, 615-byte direct infecting appender containing the texts ‘[Lauren] 0.2beta Dedicated with love to Lauren.... Have fun in NYC sweetums!!! *snuggles* Love, Cody’ and ‘*.COM’. Infected files have the string ‘TBAV’ at offset 0003h.

```
Lauren.615          B921 028D BE22 01F6 159C 0EE8 1100 E2F7 C3E8 ECFF 5A59 58CD
```

Morgoth.223 — **CEN:** An appending, 223-byte direct infector containing the text ‘Morgoth’. All infected programs have the format of COM programs and the byte 2Ah (‘*’) at offset 0003h.

```
Morgoth.223         B409 80C4 37B9 DF00 8D96 0901 CD21 EBAE 33C0 9E9F 86C4 0505
```

Paraguay.2858 — **CER:** A polymorphic, stealth, 2858-byte appender with the texts ‘VIRUS’, ‘PARAGUAY’, ‘Ver. 3.0’, ‘Programmed by Int13h, in Paraguay, South America.ANTI-VIR.DAT’, ‘C:\COMMAND.COM’, ‘CHKLIST.CPS’, ‘AVP.CRC’ and ‘C:\WINDOWS’‘CHKLIST.MS’. Infected files have their time-stamps set to 60 seconds. This template detects the virus in memory only.

```
Paraguay.2858       B440 B92A 0BBA 350B CD21 26C7 4515 0000 26C7 4517 0000 B440
```

Sirius.614 — **CN:** An encrypted, appending, 614-byte, direct infector containing the texts ‘*.cOm’, ‘TV #5 by Sirius’ and ‘VFAC’. A minor variant of this virus contains a slightly different text ‘TV #6 by Sirius’. Infected files have their time-stamps set to six seconds. This template detects both variants.

```
Sirius.614          E802 00EB 13C7 4619 3114 8B96 5902 B91C 0190 9046 46E2 FAC3
```

Skank.602 — **CR:** An appending, 602-byte virus containing the texts ‘SKANK’ and ‘(C) Dark Chakal [SLAM]’.

```
Skank.602           B479 80F4 39BA 5102 B903 00CD 215A 81EA 3402 33C9 B800 42CD;
```

Tan.1174 — **CR:** An appending, 1174-byte virus containing the text ‘TANxxxxxxx’. Infected files have the word 2323h (‘##’) at offset 0003h and their time-stamps set to 42 seconds.

```
Tan.1174            3DFF EF77 5AB4 40B9 9604 BA00 01CD 2172 4EB8 0042 9933 C9CD
```

Trivial.39.I — **CN:** Yet another variant of this overwriting, 39-byte virus.

```
Trivial.39.I        CD21 B740 93BA 0001 B127 9090 CD21 B44F EBE1 2A2E 434F 4D00
```

Trivial.243 — **CN:** An overwriting, 243-byte, direct infector. In the virus code every meaningful instruction is separated by the jump instruction (E9h 01h) skipping over one byte of rubbish (E9h).

```
Trivial.243         E901 00E9 B200 E901 00E9 8AEA E901 00E9 B1F3 E901 00E9 CD21
```

Trivial.700 — **EN:** An overwriting, 700-byte, direct infector containing the texts ‘*.ExE’, ‘!VeRSiOn.!02’ and [InComE CopYRiGHt 1998 by PsYHarmeD, Minsk]’.

```
Trivial.700         BA9E 00CD 218B D8B4 40B9 BC02 BA00 01CD 21B4 3ECD 21B4 4FEB;
```

Trivial.1842 — **CEN:** An overwriting, 1842-byte, direct infector containing the texts ‘*.*’, ‘RENEGADE IS LAUGHING AT YOUR BACK... DA RENEGADE VIRUZ...PART I’ and ‘1998 by Renegade...’.

```
Trivial.1842        BA9E 00CD 21B4 40B9 3207 BA00 01CD 21B8 0157 5A59 CD21 B43E
```

WoodGoblin.4506 — **ER:** A polymorphic, 4506-byte appender with the texts ‘DPMI error #1380: data file is corrupted (bad CRC)’, ‘C:\CONFIG.SYS’, ‘DEVICE=DEVICEHIGH=AIADWEVDVSMSHIDRAV’, ‘Please wait...’, ‘CHKLIST.MS’, ‘Data recovery’, ‘All data have been succesfully DELETED!’, ‘Thanks for your assistance.’, ‘Press <reset> to continue.’ and ‘ WG04m Copyright (C) 1995-1997 by WoodGoblin’. The virus also infects SYS files.

```
WoodGoblin.4506     B99A 11F3 2EA4 C747 0E9A 118C 4710 061F 33C0 8EC0 6626 FF36
```

Xany.979 — **CN:** An appending, 979-byte, direct infector containing the texts ‘PATH=’ and ‘*.COM’.

```
Xany.979            8B96 0906 B000 E85C FF8B D5B9 D303 E864 FFC6 8602 0401 F8C3
```

FEATURE

Reflections on the Takeover

Peter Morley
Network Associates, UK

One morning a few months ago, a sudden staff meeting was called at *Dr Solomon's* in Aylesbury, UK. It was to be held that morning, in the warehouse, and 100% attendance was required. Geoff Leary told the stunned audience that the Board had accepted a takeover proposal from *Network Associates Inc* (*NAI*), and he introduced Bill Larson to the gathering. Bill would present the situation formally at a later meeting to be held at a local hotel. The Press had already been informed of the takeover.

I was mildly annoyed that I had not seen it coming. A fortnight earlier, an anti-virus industry executive pointed out that he felt we were very vulnerable. I had laughed it off with a comment that strong defences were in place, and that the price would be rather high. In retrospect, it is clear that a well-played bid can overcome almost any defences.

Several people have commented that the price was excessive. I do not agree. From *NAI's* viewpoint, a contested bid would have been a disaster, and the cost of the uncertainty would have outweighed the difference between their well-placed bid and a half-hearted, lower-priced attempt. *NAI* got it right from the start and *Dr Solomon's* Board could find no good reason to do anything other than accept.

Bill Larson's presentation was friendly, forthright, and professional. He handled questions as if he had expected the difficult ones, and he did not flannel. There would be a period during which legal requirements were satisfied, and control would then pass to *NAI*. During this period, all staff would be interviewed, and decisions made about how integration would proceed thereafter.

The First Period

This first period of chaos is common to all takeovers, and I have several comments about it.

If an agreed takeover is cancelled before consummation, it is an utter disaster for both parties. Irrespective of the facts, the outside world will assume that something surprising and unpleasant has been detected. The share price will suffer. Big customers will be deterred by the uncertainty. It follows that facing such a situation, the staff should assume the takeover *will* happen.

In light of this, I think that there is nothing to be gained by not working normally during this period, irrespective of other activity. Certainly, participation in 'muttering groups' is unproductive to both companies, and to those doing the muttering. It pays for both parties to make a serious effort to understand the organization and culture of the other, particularly if you may want to change them later! Interviews should be taken seriously, no matter how informal and superficial they appear.

During this first period, I was particularly encouraged by the fact that every member of the *Network Associates* management team I spoke to was well aware that the world was watching intently, and many expected them to screw it up! They were unanimously determined not to. To my way of thinking, this attitude goes a long way towards compensating for technical shortcomings. I was also encouraged by the internal honesty, and the fact that there was no ducking of difficult issues.

I formed the view that *NAI* would get it together. Perhaps I was biased. I was confident the project marrying *NAI's* existing anti-virus range to *Dr Solomon's* detection and repair capability would be easier than they thought! The first period ended on time, and control passed to *NAI*.

The Second Period

The second period started with implementing the redundancies, and initial reorganizations, which had been planned during the first period. I must make the point that this is absolutely normal. When organizations are combined, it is vital to eliminate duplication of function, and get the new organization working productively. Unnecessary uncertainty is extremely costly. It is more important to take clear sensible action, than to delay and ensure perfect action. Better to do it, and correct minor errors subsequently.

NAI was obviously well aware of this principle – it should be, with experience of previous takeovers. The new organization was put in place faster than I thought it would be. It can now evolve normally.

Even before the consummation, I had known that a decision to use the detection and repair capability in *all* anti-virus products was inevitable. It came, soon after consummation, and much more quickly than I dreamed it would! This was implemented in *Scan v4.0*, and completed ahead of schedule. I think the world will love it. The other products will follow, now the techniques have been learned. The second period is now winding down, and we are approaching business as usual.

The Future

I think we are in for a good patch. *Network Associates'* anti-virus product quality will soon be ahead of the game. A well-balanced product range is always worth more than the sum of its parts, and is an ideal starting point for further developments.

INSIGHT

Norman Wisdom

Norman's Righard Zwienenberg is a straight-talking, no nonsense joker. His early days were spent frustrating his teachers at school – his education is still on-going today.

'I was born in The Hague, the Netherlands in 1967 which makes me... (Start->Program->Accessories->Calculator: 1998-1967=) 31 at the time of writing, most likely 32 at the time of publication. Most of my schooldays I spent out of the classroom, much to the disappointment of some of the teachers. I recall their famous saying 'You'll never accomplish anything in life!' – seems they were right! I was more interested in Sound and Light Engineering in the high quality plays and shows for which our school was famous in those days. During my last years there I even produced them. I also automated the school's administration (they only had XTs there at that time, real speed-monsters). After high school I went to classes full-time at the Technical University at Delft (what an experience!) but I changed to Technical Night College as I had started working and could not combine the work and study. I am still studying and hope to graduate in the second quarter of 1999.'

Starting Out

His first relevant experience with computers was when he was nine years old. He spent much of his time at the School Museum where there were several computers available to the public. 'I started out on a *Commodore* Pet-2001, with a whole 8 KB memory, a keypad as if it was a calculator and the speed of… well, nothing. Later they got some CBM 8032's with 16 KB and 32 KB. I started to work for them when I was in high school, teaching other school kids how to work on them or program.'

Later, he remembers, the Museum exchanged several computers for the C64, one of which he bought. He had his first assembler experience while programming the 6502 processor. He has fond memories of his early days with computers. 'After the C64, I got into Intel when I could afford my first PC-XT, made a side step to PDP-11 at university (because I had to) but went back to Intel again. Right now, you can buy a fairly good machine for less then my first PC-XT. Still, the old days had some magic. It was a great feeling when all the hardware worked, as there was nothing like Plug and Play in those days.'

His first encounter with a computer virus occurred while studying at the Technical University, where he came across a strain of the Jerusalem virus that became known as Jerusalem.1808.A-204. A-204 was the ID for the Software Engineering course at that time and he feels sure one of his fellow students modified the original Jerusalem virus from which this strain is derived. Following this episode and his publications on it, Righard got in contact with several other people in the Netherlands who shared the same interest (among them Frans Veldman from *ThunderBYTE*). He started to work on (at that time) Jan Terpstra's VirScan.Dat signaturefile, analysing viruses and adding signatures to VirScan.Dat which at that time was used by TBSCAN and HTSCAN. It snowballed from there.

'I got involved in the VSG (Virus Strategy Group) which was a combination of business, judicial and government bodies discussing the virus threat and trying to solve it. Then I got in touch with others like Alan Solomon and Vesselin Bontchev, which finally resulted in me becoming a member of the beer drinking club CARO early in 1992. My parents still can remember the days when Alan Solomon used to call me between 2 and 3am, but maybe the phone bills also help to recollect these events for them!'

It was at this time that he bought 20% of a company and started working as its R&D Manager. *Computer Security Engineers* (*CSE*) *Ltd* was a Jersey-based company ('naturally, the location was selected for its beautiful weather and not because of the tax climate!'). This was the start of his work in the anti-virus industry, helping with *CSE's* anti-virus product PCVP.

He soon became restless, and in September 1995, at the *Virus Bulletin* Conference in Boston, he decided to quit *CSE Ltd* and look for a new challenge. He was characteristically relaxed about his options. He selected three options out of the resulting propositions, talked to various people and went on holiday with his fiancee, Els (he married her on 2 May, 1997). As she is a very important part of his life, they made the decision together that he would discuss *ThunderBYTE's* offer.

Starting Over

'I started working there in November 1995. Together with Frans Veldman, I was responsible for the *ThunderBYTE* engine and the virus research. *Norman Data Defense Systems* acquired *ThunderBYTE* and I am now working for *Norman* in the team responsible for the engine used in all *Norman* anti-virus products. The transformation was not that hard as I was working closely with some of the *Norman* guys already due to the strategic alliance with *ThunderBYTE*. *Norman* is an innovative company in which I really get the chance and time to research new threats as the company genuinely values this type of work.'

It obviously hits a nerve to ask Righard what his current job title is. 'It would most likely be something like "Senior Research and Development Engineer", but what's in a title? I don't care much about any title if the work is not interesting. I know some people who are now vice-presidents in the companies they work for, but when I ask them vice-president in what, they can't tell me, but they feel overly important. If so, I usually introduce them then as vice-president wastepaper basket or vice-president ink-eraser. Even then I'm not far off the truth!'

His speciality, and the thing he loves most, is reverse engineering of formats and researching new threats, besides debugging. He also enjoys virus research. Recently he finished working on a rewrite of the ExcelFormula algorithms for *NVC* and will continue to work on its Access algorithms. After that, he is confident, there will be so many new security holes that he will find something to do.

He intends to stay with viruses for a while. 'There are so many new threats coming up and lots of opportunities for misuse, especially at a higher level. The macro virus problem is progressing to Visual Basic Script viruses, which again can be placed in HTML files, and *Office 2000* is coming up. With the open structures of Microsoft, I guess I'll be busy for a while. That is fine for me, but I do want to retire when I am 50.'

Settling Down

The Zwienenbergs have an 11-year old Golden Retriever that was raised from nine weeks by Els – 'We do not have any kids yet, but maybe by the time this is printed, we will be expecting!' The three of them live in a top floor two bedroom apartment in the southern region of The Hague. At the moment, their new house, with four bedrooms and separate office space, is under construction and due for completion in April 1999.

Righard is rarely away from his desk. 'I don't get much free time – I am the prototype of a workaholic. When I don't have anything to do, I jump back to work again. To be honest, my real passion is Els! Any free time I take is always spent with her and our dog. If we get tired of things, we pack our bags and move north to stay a few days with Els' family up there. Household jobs are done by both Els and me. I am a strong believer in equal rights and equal opportunities, so I am doing my share. Right now, all my hobby-time goes into preparing the new house, selecting furniture and so on. We both love playing video arcade and interactive multimedia games, especially the Star Trek ones. We are both really serious trekkies. I get plenty of exercise every day. For sports I walk the seven steps up and down a minimum of four times a day!'

Righard counts stamp-collecting as his official hobby, but he likes to cook too. He even treats this talent with typical comic modesty. 'I have worked in kitchens during summer holidays and often helped out the cook in the kitchen of my student fraternity. When I started to live with Els, it was somehow difficult to cook just for two people. One or two handfuls of salt too much in a meal for 400 people hardly gets noticed, but in dinner for two…'

Settling Scores

On the subject of viruses, and their creators, Righard has some serious opinions. 'Unfortunately, virus writers will always exist. They will use different techniques and platforms to write viruses on, so the emphasis will keep shifting. These new items generate a challenge for them, much like our challenge to protect our users. The virus writers are getting smarter by the day and all the security holes in operating systems and application software are, of course, to their benefit.

Virus writers tend to regard themselves as highly competent programmers who do not get a chance to work at big software development companies. Surprise: they are not competent. Writing viruses is easy compared to writing anti-virus products. The virus writers do not care about bugs, compatibility, crashes, etc. Our customers do, so a great deal of time is taken up by Quality Control nowadays. It is sad to see that people get into writing viruses when they could better use their time enhancing their skills and techniques to become real programmers.'

Personal experience has proved that his adversaries are not as formidable as they believe themselves to be. 'Some of them are so keen to boast about themselves that they make mistakes, revealing their true identity – among them Trident's Dark Helmet. I received threats from this guy by netmail. Later, in one of the virus-related echomail groups, a person R. De H. (initials only due to privacy) publicly admired Dark Helmet, and at the same time wrote offensive things about me. The latter was clearly a cut-and-paste action. The middle and last initial was D(ark) H(elmet). Great self-PR. He became really silent in the echomail when I pointed out the similarity.'

Righard is also quick to point out that the impression virus writers have of providing the anti-virus industry with jobs is a sorry misconception – 'sadly, if all virus writers stopped writing and at the same time all viruses on this planet were deleted, we would still have plenty of work.'

He is equally vociferous on the subject of ethics, and considers himself to be an active participator in the ongoing debate about the up- and down-conversion of macros. 'Ethics are very important, but should not be exaggerated.

Some of my respected colleagues are against up- and down-conversion on the grounds that it creates new viruses. Others, myself among them, have done it but only to make detection and repair capabilities in our product. Afterwards, I have always destroyed the samples.

'I do consider it unethical to keep or spread these samples. Since users can do it, for example by saving an Excel97 spreadsheet in Excel95 format or loading an Excel95 spreadsheet into Excel97, it is likely to happen anyway. Then we have to decide to do what our customers pay us for: protect them against new viruses.'

An article was introduced in the Dutch 'Criminal Code' in 1992 (article 350a and 350b) which makes it illegal to distribute viruses with the intention of causing damage. When the code became effective, the virus writing group known as Trident, responsible for several viruses and the TPE engine, became inactive. Righard is keen to see more countries introduce such legislation. 'A few countries do have them, but only a few. Virus writers often misuse phrases like freedom of speech or freedom of expression to defend their creations. Most of them do not care that other people's data is put in danger. How would you defend CIH? From the users' point of view, it can actually destroy their systems. I wonder how the author would react if the computer responsible for paying his/her salary were actually hit by CIH's payload…'

His preferred anti-virus methods differ and depend on the situation. He favours unique identification, but appreciates that, on occasion, the flood of new variants is so overwhelming that generic detection is necessary too. Righard knows that the future holds several new types of viruses on different platforms and he doubts that unique identification will continue to suffice. Moreover, he does believe in a form of heuristic detection, but is scornful of the extent to which some companies produce and promote it. He knows from experience that it is very difficult to set a threshold to prevent false positives.

The Final Countdown

His final word is on the future of his industry, a prediction which may seem familiar to readers of this column. 'The future of the AV industry will change completely. It will not disappear, but slowly it will transform into an anti-malicious code industry.

Furthermore, the trend will most likely be integrated security instead of separate products, certainly for the corporate market. Right now the number of technical people in this business is decreasing. It is already very difficult to cope with all new threats. The only way to get new skilled people is to buy existing companies and take over the people you would like to have in your team. But since the number of different companies is getting smaller, even this will become harder and harder. Maybe we have to make one big company where all developers and researchers will be working on the same (and only) product?'

VIRUS ANALYSIS 1

The New Frontier?

Darren Chi
Symantec

On 17 December 1998, a large telecommunications company discovered that a virus was loose on its *Windows NT* network. The company's anti-virus vendor was notified and publicly announced the virus on 21 December. Named Remote Explorer, it was characterized as a 'smart network virus' – a form of 'cyber-terrorism'. This description was later deemed inaccurate after additional research revealed that it was not so smart after all. In fact, calling it a network virus is an exaggeration.

In short, Remote Explorer becomes resident on *NT* as a service, infects EXE files, and renders non-executable files unusable. It has no other destructive behaviour.

Residency

Remote Explorer can only become memory-resident on an *NT* system running on an *Intel*-compatible CPU. It installs as a service using standard *NT* API calls. This happens when a user with administrative privileges runs an executable file infected with the virus. Once installed, the virus remains active, even if the system is restarted. The viral service resides on the system as a file named IE403R.SYS in the %winroot%System32\drivers folder.

A user can easily determine whether or not the Remote Explorer service is installed. The Services applet in the Start menu under the Settings menu lists the services currently resident on the system. If the viral service has been installed, it will appear as a service with the name Remote Explorer. As Remote Explorer resides as a service, it cannot operate under *Windows 95* or *98*. With the former, running an infected program causes *Windows* to display an error message about a missing DLL. With the latter, the virus succeeds in extracting itself and running the host file but cannot become resident.

Despite claims that Remote Explorer can travel across networks, this is not the case, at least, not technically. The virus can only become resident on an *NT* system if an account with administrator privileges, or at least one that has the right to load a service, runs an infected program. The virus itself does not have the capability to 'crawl' across network cables and through hubs to install itself on another *NT* system.

Other rumours about Remote Explorer describe the virus stealing administrator privileges. This is not true, but it may appear that way because once Remote Explorer installs itself as a service, it appears to have no problem infecting files, whether on local or network drives.

Infection

Once installed as a service, the virus then goes about infecting files in the background in a way that is intended to be unnoticeable to any user who may be currently working on the system. Essentially, the virus has two modes of operation. In the first mode – on any weekday between 6am and 3pm – it sets its thread priority to the lowest setting. In the second – on Saturday or Sunday or any weekday between 3pm and 6am – the virus sets its thread priority one notch above the lowest setting. The thread priority of a program determines how often the program gets CPU time. Thus, the virus will only get a chance to infect additional files during those times when almost nothing else is happening on the system.

These times appear to have been selected within working hours for the first mode and outside working hours for the second. (In the United States the difference in time zones between the West coast and the East coast is three hours.)

When the virus does get CPU time, it looks for files to infect both on drives local to the system and on network drives. Network drives include both mapped network drives and those accessible using a UNC path. It is the infection of files on network drives that gives Remote Explorer the notoriety of being a ‘network virus’. The fact is that this is no different from the method other DOS and Windows direct action viruses use to infect files. They simply search for target files using standard operating system calls that work both for local drives and for network drives.

Unfortunately, the virus affects both executable files and documents and other data files. It considers files with an EXE extension to be executables. A file with any other extension, except DLL, OBJ, and TMP, is considered non-executable. When an infected EXE is run on a system on which Remote Explorer is already installed, the virus extracts the stored copy of the original host file and runs it. On a system where the virus is not yet installed, running the infected file using administrator privileges will install the viral service and then run the original host.

Infected EXE files change in size, often by around 100 KB. The original host is stored in the infected file in a compressed form, so the difference in size depends on how well the host compresses.

After Remote Explorer touches a non-EXE file, the file becomes unusable and appears corrupt, but it retains its original size. Luckily, these effects are reversible.

Executable File Host Storage

One might characterize Remote Explorer as a virus that masquerades as the host, while having the original neatly folded into a compact form and tucked away in one of its pockets. It is a *Windows* 32-bit executable file in standard Portable Executable (PE) format. The PE format divides an executable file’s contents into sections such as code, data, and resources.

When Remote Explorer infects an EXE, it replaces the host with a copy of the virus, adding the original (as compressed data) to the resource section of the copy. The compression algorithm is the deflate method described in RFC 1951 and the compressed data is stored in gzip format as described in RFC 1952. When an infected file is executed, the virus gets control, attempts to install itself as a service on the system, then decompresses the host to a temporary file to run it. Disinfection requires locating the resource containing the compressed host and restoring it to its original form.

Non-Executable File Corruption

Oddly enough, Remote Explorer ‘corrupts’ non-executable files in a way that allows them to be ‘uncorrupted’. Perhaps the virus author envisioned that the virus would be able to reverse these effects as corrupted files were accessed. On a system where the viral service is not installed, the file would then just appear corrupted. As it stands, however, Remote Explorer has no such ability.

The corruption consists of three steps. First, the original file is compressed into gzip format with the Deflate method. Following this, the virus encrypts the compressed result. The encryption algorithm encrypts each byte by adding a value to the byte and then permuting the bits in the result. The value added to each byte and the permutation depend on the history of all bytes encrypted so far. Lastly, the encrypted result is written to the start of the original file and all remaining data in the file following the encrypted result is overwritten with random data. Fortunately, Remote Explorer’s encryptor uses the same key. Thus, restoration involves decrypting the encrypted data, then decompressing the result to the original.

Conclusion

Remote Explorer is certainly not a marvel of ingenuity. The virus’ code was written in C rather than in assembler and depends upon API calls and functionality from external libraries such as one for compression. It contains no sophisticated routines or mechanisms. In the end, Remote Explorer is a very ordinary virus.

Remote Explorer

Aliases:	W32.RemoteExplorer,WinNT.RemEx, W32.RICH, Win32.RemExp.
Type:	*Windows NT* service-resident direct action EXE infector.
Residency:	*NT* service named Remote Explorer.
Infects:	Executable files with EXE extension.
Payload:	Encrypts non-executable files.
Removal:	Replace infected files from backup or originals. Some vendors provide decryption for affected data files.

VIRUS ANALYSIS 2

Curse of the Incas

Snorre Fagerland
Norman Development AS

In August 1998 I received a virus sample from the field. It had been distributed via Usenet, and had caused a few solid infections around the world – the US, the UK and Germany. This new Win95/98-specific virus was W95/Fono.17152, also called Inca. It turned out to be quite multi-faceted, with a few interesting features. It was the first *Windows 95* infector that also infects boot sectors; a slow polymorphic virus, it places virus droppers all over the place and tries to exploit the *mIRC* chat program to spread over the Internet.

Residency

The residency code is not very advanced compared to the fast and ruthless method used by Win95/CIH (see *VB*, August 1998, p.8). Fono drops a VxD into the *Windows* system directory, causing it to load from the [386Enh] section of SYSTEM.INI – a method that was used in the early days of *Windows 95* viruses (the first to use it was Win95/Punch).

When virus authors discovered how easily *Windows 95* memory could be manipulated, the VxD method was all but abandoned; VxDs are obvious and cumbersome. However, Fono's author obviously wanted it to be able to activate from DOS droppers and boot infections, i.e. 16-bit code. The simplest way to achieve this involves a VxD, enabling the virus to set up for infection even if *Windows* has not yet loaded. The VxD is always called FONO98.VXD.

When starting *Windows 95*, the VxD Device_Init procedure calls VMM with the Close_Boot_Log API requesting, predictably, the boot log to be closed. After that, it installs an IFS hook and enables Fono to infect different targets on File_Open and hooks into the V86 interrupt chain, filtering Int 13h requests in order to install the dropper on diskettes. Int 13h filtering is made possible by using a trick borrowed from Hare and Dodgy – deleting the 32-bit floppy device driver HSFLOP.PDR from the IOSUBSYS directory. This causes *Windows 95* to use standard Int 13h floppy disk access. At this time Fono initializes all polymorphic buffers and all the decryptors will be static until the next bootup.

Windows 95 File Infection

The IFS hook set up by the initialization procedure triggers if the function called is File_Open. If the file opened is a PE EXE or SCR file larger than 8 KB Fono will attempt infection, unless the file has a PE header offset equal to or more than 1024 bytes or the application imagebase is set to something other than the default 400000h. It will not infect DLLs either, even if the DLL has an EXE extension.

A potential host is read into memory and the infection takes place there, reducing disk access. Fono creates a new, randomly-named section in the file, placing its code there. Appropriate fields in the section table and PE header are updated to reflect this, and the entry point relative virtual address (RVA) is redirected to the Fono code. These are standard tactics for PE infectors.

When an infected *Windows 95* program is executed and the code has been decrypted, Fono scans the KERNEL32.DLL memory for the GetModuleHandleA and GetProcAddress APIs. From then on it uses GetProcAddress to get the APIs it needs for the next step – creating the VxD via a dropper (W95INCA.COM). This file is created and executed by Fono, which gives it three seconds to finish up before attempting to delete it again. It would have been just as easy to do this directly, but I assume the author wanted to reuse the dropper code.

Droppers

Fono knows four archive formats – ZIP, ARJ, LZH and RAR. LHA and PAK files use the LZH format, so Fono looks for those as well. When an archive of one of these formats is opened, Fono places an uncompressed dropper in the archive. The droppers are trivially encrypted COM files, with random four-letter names. No files inside the archive are infected, as Fono does not know about compression algorithms.

As mentioned earlier, Fono tries to exploit the *mIRC* chat program to spread over the Internet. If the 32-bit executable, MIRC32.EXE, is opened while Fono is resident, Fono will install several files in the *mIRC* home directory – more on this later. W95INCA.COM is dropped to disk and executed every time an infected file is run.

The droppers that Fono installs in different instances are all similar except for a rather trivial encryption scheme. They drop the VxD, which is compressed inside the dropper, in the system directory. After changing SYSTEM.INI to load the VxD they just terminate. Fono's compression is moderate – decompressing the VxD increases its size from about 11 KB to 15 KB, i.e. a compression factor of 26%. In comparison, ZIP has a compression factor of 60%.

Floppy Boot Infection

The second entry point in the VxD is the V86 mode Int 13h hook. On a read request for the boot sector of a 1.44 MB diskette, Fono will infect the floppy. The original boot sector is replaced by a polymorphic one, and more virus loader code is written to the last sector of the root directory. The compressed VxD is written to the last cylinder of the floppy, occupying all 36 sectors on that cylinder. The total number of sectors in the BPB is decremented accordingly.

When booting from an infected diskette, the boot sector, which consists of just a few instructions (apart from garbage) will load the boot loader code from the root directory (0,1,14) into 0:7C00h and jump there. This code hooks the timer tick (Int 1Ch), reduces available memory by 1 KB, and copies itself into this hole. Its job done for the time being, it assumes that sector 0,1,1 on the first hard drive is a valid system boot sector, and loads and executes it. In effect, it causes the computer to start from hard disk. There is little error checking and no check against the partition table to verify that the boot sector it loads from the hard drive actually belongs to a bootable partition.

Monitoring the timer, the virus checks for changes to the Int 21h vector. When this vector has changed three times, Fono assumes DOS is up and running, and that it is safe to use and monitor DOS calls; it hooks Int 21h. The next execute request (Int 21h, AX=4B00h) triggers installation of the VxD in the *Windows* system directory. After this, the system hangs, forcing the user to reboot and load the VxD.

Polymorphism

Fono uses encryption and/or polymorphism in all its forms except the VxD. The boot infection is polymorphic, but not encrypted. This polymorphism consists of a multitude of MOVs, followed by many arithmetic statements (XOR, ADD, ADC, etc) designed to arrive at an Int 13h statement with the registers set to read the rest of the Fono loader code. Such a boot sector contains little static code; but the composition of statements is very unusual and a dead give-away that something odd is going on. The DOS droppers are polymorphically encrypted, but the algorithm is not very advanced and a good emulator chews right through it.

The infection of Win32 executables is something else entirely. Fono uses a true 32-bit polymorphic entry routine that consists of a multitude of CALLs, RETs and JMPs along with standard trash code. This decryption loop represents something new – standard linear decryption with a combination of functions and keys is replaced with a table translation scheme. Inside Fono there is a 256-byte translation table for opcodes. This table is, in the beginning, just a line-up of all opcodes from 0h to 0FFh, but a random number of bytes are switched around so that some opcodes or bytes have their values changed, for example:

```
00 01 02 03 04 CC 06 07 08 09 0A 0B… FF
```

Every byte is looked up at translation-table + byte-value and replaced with the value there. In the example, most bytes are not affected except for 05h and CCh, which will be replaced with CCh and 05h, respectively.

Exploiting *mIRC*

The *mIRC* chat program has been shown to be a viable platform for malware distribution. First came SCRIPT.INI viruses (see *VB*, April 1998, p.7), then the DMSetup series, more successful than the scripts as they did not need to be downloaded to the *mIRC* home directory to spread. Fono's author has recognized the spread potential in IRC, and installs several items in the *mIRC* directory if MIRC32.EXE is detected on the machine.

It creates another COM dropper (INCA.EXE) and overwrites SCRIPT.INI with one instructing *mIRC* to send this dropper and the SCRIPT.INI to others at a JOIN or PART event. These occur on IRC at the moment someone enters or leaves the user's current channel or chat-room. In addition, the SCRIPT.INI installs several backdoors enabling others to cause damage.

Fono also drops REVENGE.COM which writes a random password to your CMOS then reboots. This is executed by *mIRC* when someone types 'el_inca'. Another backdoor is initiated when someone types 'ancev'. This causes *mIRC* to put the host into fileserver mode, with unlimited access for the person who typed the word. To hide this, MIRC.INI is overwritten with instructions to turn the fileserver warning off. Luckily, Fono was a little late in supporting *mIRC*. The SCRIPT.INI and DMSetup incidents meant most of the obvious security holes in *mIRC* were patched. Further, IRC users have increased awareness of the potential threats.

Summary

Fono is a simple yet complex virus. Some of its methods have been lifted from other viruses (Hare, Zhengxhi) – other methods are new, but rather crude.

The evolution of *Windows* viruses follows that of their DOS forebears – from direct action to resident; from unencrypted to encrypted to polymorphic; from single target to multiple targets. *Windows* is a much more complex operating system than DOS; once the virus authors overcame this initial complexity, they started using it to their 'advantage'. Fono is, I fear, a taste of what may be to come.

Win95/Fono.17152

Alias : W95/Inca, W95/El_Inca.17152.

Type: Resident Win95 multi-partite PE infector.

Self-recognition in Files:
CRC field in PE header 12345678h.

Hex Pattern in PE Files, Boot Sector and Droppers:
Not possible, as it is polymorphic.

Hex Pattern in VxD Control Procedure:

```
CD20 8A01 0100 32C0 A2A1 0000
00A2 A300 0000 A2A4 0000 00A2
A200 0000 66E5 40C1 C810
```

Payload: Installs *mIRC* backdoors for fserve and execution of a CMOS password Trojan.

Removal: Replace infected files from backup or originals. Format infected diskettes.

FEATURE

Viruses and the Internet – Whatever Next?

Eugene Kaspersky
Kaspersky Lab

Are real, complex network viruses possible in the modern global network? 1998 could be named the year of the hacker's attack on the Internet. The main modern Internet applications – Internet browsers and email clients – were hit. We have been waiting for it because the global network is the cherry on the virus writers' metaphorical cake. Practically all known underground hacker teams are investigating the virus capabilities of Internet applications and trying to attack them. Good old DOS viruses have been forgotten, the aim of modern hackers is The Virus, spreading freely via the Internet, infecting local networks, remote workstations and home PCs.

Unfortunately, they have had some success in this. During 1997 and 1998 several new viruses appeared which use the Internet to spread themselves by email. The majority of them are macro viruses which use standard *Windows* functions to access the installed email reader, create an attachment with a virus copy and send the infected messages to randomly selected addresses. A careless user receiving this message and opening the attachment in *Word* or *Excel* (depending on the virus type) gets infected, and the virus continues spreading in infected messages – but this time from a new address.

WM/ShareFun sends infected attachments via *MS-Mail* (see *VB*, April 1997, p.10), the *Word 97* macro virus Antimarc uses *Outlook Express* to spread itself; Win/RedTeam parses the *Eudora* Outbox inserting a message containing an infected EXE (May 1998, p.6); the *Word* macro virus Innuendo uses the universal method that allows the sending of infected messages by any type of installed email software; the *Windows* virus Parvo (January 1999, p.7) uses *Windows* API functions to access Internet resources.

The second half of 1998 also saw attacks on the second major 'Internet application' – the web browsers. The first target was the well promoted Java language, widely used in web page development. In August 1998, an unknown virus writer released the first virus to infect applications written in Java. The second application to succumb was Visual Basic Script (VBS), which is also widely used in web pages. Based on *Windows* script infectors, the first viruses to spread via HTML files appeared in November 1998, opening the vista of an infected web.

Despite this, not all known viruses which attempt to use Internet resources to spread are as dangerous as may be imagined. Often, their harmful potential is purely theoretical: the viruses are either visual (if they are sent as attachments in email), or non-functioning – popular Internet browsers generally have security features that either cancel any attempt at virus-like behaviour, or warn the user.

Is there a real possibility that today's theoretical dangers will develop into tomorrow's 'Internet creatures'? Is computer terrorism possible in modern networks? Is it safe to read incoming email from unknown addresses and visit unknown Web sites? What about known addresses and Web servers? Will it ever happen that we partly or absolutely stop using the Internet?

These are quite complex questions, and it is not easy to answer them. In an attempt to do so, let us investigate from the hacker's point of view, swapping our security expert hats and examining the Internet through the eyes of the experienced hacker. The best way to test a security system is to try to break it, right? Welcome to the opposite side of the field. We should 'invent' the scenario of a modern network virus' life cycle, then go back to security and ask the obvious question: what do we do for protection?

Scenario

So, we have decided to create a super-virus which lives in Internet networks, copying itself from server to server, and infecting everything in its way. First of all, we have to define the target of our investigation, then the tools and means to reach it.

What is the target? A network virus – penetrating imperceptibly from the Internet into a local network, infecting it and then migrating back to the Internet to spread to and infect other remote networks. This means that the virus has to be executed automatically on a PC producing neither error nor warning messages while infecting objects on the local network. It then needs to propagate copies of itself back to the Internet, i.e. the virus must have a complete 'network life cycle'.

At the same time, the virus has to be absolutely compatible with most popular kinds of Internet application, and with all modern versions of it. There are no tricks with beta versions of unknown software allowed – our aim is a virus that is 100% compatible with modern networks; a virus that spreads everywhere, catching the network resources it needs and infecting everything.

The target is now clear. What tools can we use? Email and browsers – these are the most important kinds of application to pay attention to. Other 'whimsies' like chat channels should be left out because our targets are corporate networks and Web servers, not teenagers' home PCs. Data transfer protocols like FTP do not get a mention either, because they only allow data to pass to a remote server, but not to execute there. Anyway, FTP may be used as an additional tool, but not here. Then we should divide our main task into several steps. Step 1 – penetrate a computer from the Internet. Step 2 – infect the local network. Step 3 – look for Internet connections and expansion to the Internet from the infected local network.

As has been mentioned, the virus only has two ways to infect the local machine from the Internet – being opened or executed as an attachment from the affected email message, or while accessing an infected Web page via an Internet browser. In the first case, the simplest way to go to the local computer is via a specially prepared *MS Office* component: a *Word* document or an *Excel* spreadsheet (an *Access* database and *PowerPoint* can also be used, but they are not as widely installed).

All of them allow macro programs to embed into a file and in all of them there are colonies of macro viruses in residence! Unfortunately, (remember – we are hackers), macros written in Visual Basic for Applications are not good enough to spread our virus over a local network, executable files (DOS or *Windows*) are more suitable. No problem – VBA is powerful enough, and it can carry, drop and run any kind of executable file.

In the case of an Internet browser, it is sufficient to include in an affected HTML file a script-program, written in VBS, Java Script or some other language supported by modern browsers. The virus script, when activated, can drop and run the embedded executable file as well as the VBA macro (see above), or download and execute it with standard browser functions.

All of the above also applies to email systems because many popular mailers (*Outlook*, *Netscape*) support the HTML format, and script programs are accessed and executed in the same way as by a browser. The virus need only send its message in HTML format and add a necessary set of instructions.

The first step is complete. The virus (or its 'head') is active in the local computer and has completed itself with the main EXE virus component. Now is the time for Step 2, which is not new. There are thousands of different types of computer virus, from primitive COM infectors to extremely complex *Windows* ones, that infect the local network, and will do so in the future. So, this step is easy.

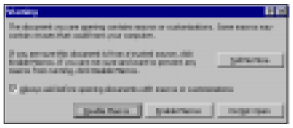
For Step 3 there are two main varieties of Internet infection: emailers and browsers. With email, the virus just needs to get some real email addresses, create a new message with an infected HTML-based attachment or add its HTML dropper to an existing message. With browsers, the virus can use several methods to spread, from silly searching and infecting a Web server in the local network (if it exists), to a very complex method which involves, for instance, installing itself as an IIS filter (Internet Information Server) and, without modifying the original contents of the Web server, sending out infected HTML pages.

Of course, to get access to a local Web server and moreover to an IIS service the virus has to have the correct (Administrator) privileges. Most viruses manage this: in practice, if a parasitic EXE virus infects any workstation, it will eventually occupy all the PCs on the local network, including the Administrator's workstation. In order to access the Web server or IIS it just needs to check the privileges of the current user. If they are too low – hang on, that is not the time to spread. If they are the Administrator's – onward!

The scenario is complete. All the necessary steps have been described, and a terrible virus can be written by following them. Is this scenario good enough? Are there some mistakes and inaccuracies? The second and third steps seem to be absolutely correct – such viruses already exist (except those affecting IIS server).

What about the first step – infecting a local machine from the Internet source? Why is it that virus writers cannot find easy ways to spread their parasitic creatures through the Internet? Unfortunately, they wonder about that too, and some have even tried it – with no success up to now. [*This article was written at the end of November 1998. The HTML/*Excel *breach had already been found by Eugene's team, but it had not been announced by* Microsoft *and* Finjan, *and it was not known to virus writers. Ed.*]

On the road that our fantastic Internet worm has to take, there is a closed door: to penetrate a local computer from the Internet it has to load its code into system memory, execute it and get access to local files and resources, i.e. bypass the protection mechanisms embedded in Internet browsers and *Microsoft Office* applications. Due to this barrier, global network viruses will (fortunately) remain fantastic creatures, never to occur on our lovely networks – emailers and browsers will just kick them out of protected computers and not allow them to spread.



Standard Protections

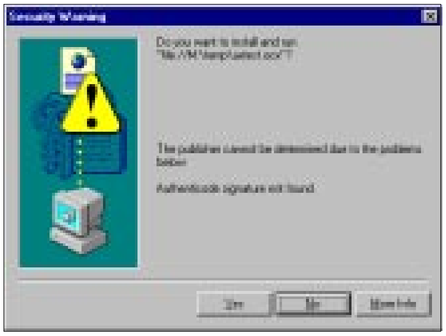
So the spread of Internet worms to local computers is cancelled by a set of quite powerful protections. To learn more about the potential methods of Internet intervention it is necessary to analyse the most important of them. We should start with *Microsoft Office* and the macro viruses which have found a comfortable 'ecological niche' there. There are several versions of *Office*. The version released in 1995 already had features conducive to macro viruses and their free life there – support for Basic macro programs and access to a set of *Windows* functions, are both built into *Word*, *Excel*, *Access* and *PowerPoint*.

It is also necessary to note that macro viruses are the most widely spread viruses and (pay attention!) do not have any network abilities. Even though they are replicated millions of times and sent via the Internet thousands of times a day in infected documents and spreadsheets, most of them do not use any Internet functions. They only use the global network in a passive manner – infected files are attached manually here and opened manually there. Moreover, there are blocks to their spread – the built-in *Office 97* mechanism warns the user about the potential danger of a macro program that appears in a file. Anyway, macro viruses are worth a mention because they can be used as one of the parts of a multi-component network worm.

The second route an incoming Internet virus can take is via programs downloaded and executed on a local computer during Web site visits. There are several types: Java applets, scripts in HTML pages, active components, additional tools in I-Frame, dynamic HTML (DHTML). These tools are used in the development of complex Web pages, where, in addition to static information, there is dynamically changing data.

There are known viruses which infect Java applications and HTML scripts, but they fail to spread. To infect any object on a remote workstation or server the virus has to open it, i.e. to get access to remote files. Typical web browser protection settings either disable any access to data and resources on the computer, or display a clear ActiveX warning message about possible danger.

The same goes for active controls: the browser's protection warns a user about downloading unsigned (uncertified) controls. You would think that with all this information, you could relax and use the Internet without thinking about potential trouble, but… News services often bring us stories of recently discovered bugs or exploits in computer security. Of course, such bugs are already fixed, and the patch is available here, there and everywhere. So the question has to be asked – do these protections really work against all the above listed methods of virus spread?


Security Warning

Do you want to install and run "file://M:\temp\select.ocx"?

The publisher cannot be determined due to the problems below:

Authenticode signature not found.

Holes and Breaches

The possibility of modern Internet viruses depends on the quality of security protections guarding the computer from malicious programs coming from the Internet. Let us find out: is the protection built into *Office* and web browsers really safe, do they protect computers adequately?

The easiest way to get into the computer is to fool the user and force them to answer 'Yes' to a corresponding warning message. That trick is used by the 'NoWarn' HTML virus – it displays its own message, hiding the standard ActiveX warning. Only the warning's Yes/No buttons are visible.

Of course, such a tricky technique cannot be considered a 'hole' in the protection. Any sensible person will not be fooled. However, are you sure that all employees in your company are that sensible? Are you sure all of them will press the correct buttons after the message 'For best viewing of this site it is necessary to install our video accelerator, please answer Yes to the following questions'? Remember that to infect the local network the virus has to infect just one workstation.

Now, let us turn our attention to macro viruses. The mechanism of macro virus prevention built into *Office* has two negative points. The first is that it only appeared in the *Office 95a* release and not in earlier builds. Secondly, it cannot separate trusted macros from those received from unknown sources (let us hope we will see this in *Office 2000*). As a result, it is often switched off in the case of companies that use macros to build documents, or when other special macros are in use (for instance, text auto-correction or value auto-recalculation).

Thus, while the macro virus protection in *Office* does its job, it is not switched on and active all the time. This creates the hole for our Internet worm: Internet browsers are able to load an infected *Word* document or *Excel* spreadsheet with the necessary set of instructions, open it on a local machine and activate the virus macro code.

The *Internet Explorer* ('medium security') protection, installed by default, allows this, referring the security checking to *Word*/*Excel* – but the virus warning there may be disabled! The virus can propagate into the local computer, disable other security settings, download and run its EXE component etc, without any warning messages from *Office*, email or browser protections.

Re-read the last sentence. Do not trust your eyes and read it again. If the *Office* macro warning is disabled, and *IE* security settings are the defaults, it is possible to get the Internet global network virus. Under these conditions it is possible to get a virus with an HTML component placed on a remote Web page, which, being opened, downloads and opens an infected *Word* document. Then, using the macro program, it spreads into the computer.

So, a two-component HTML/*Word* virus is able to spread to a local network from the Internet if the *Word* macro warning is switched off. Is it possible to infect the local network if the *Word* macro warning is on, and the warning for HTML scripts is disabled? Yes, unfortunately. When the ActiveX protection is disabled and scripts are able to access local resources, the virus script can disable any *Office* protection with a few instructions.

So, the global network virus is possible under limited conditions – when one of ActiveX's or *Office's* protections is disabled. In addition to scripts there are other program types which can be used in HTML pages instead of the HTML/*Word* combination, because Internet browsers are complex and universal enough applications to be integrated with operating systems and lots of add-ons. While working with OS resources and applications, depending on the situation, different security scenarios are in use to allow the safe viewing of web pages.

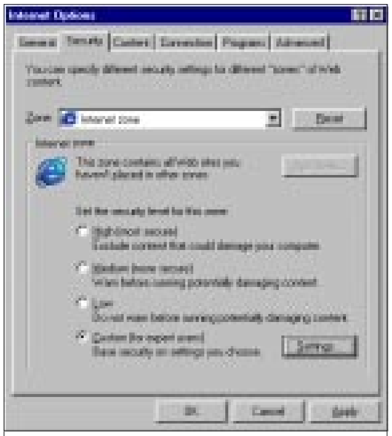
Do these security scenarios cover each other, or are there holes? Is it possible to fool security protections with a complex, multi-component HTML/Java/*Word* virus, which disables them one by one and penetrates the local computer and thus the whole local network? Unfortunately, we must concede there is no guarantee against a virus breaking these protections.

What Should We Do Then?

Despite this black conclusion we should not panic (after all, this is the first rule if your computer gets a virus, right?). The probability of the emergence of the complex Internet virus-monster we have 'built' above is very low. To develop such a program requires a great deal of programming experience. Virus writers are generally at the other end of the spectrum, being too young and inexperienced to create such a virus. By the time they gain the necessary knowledge, they have usually stopped writing viruses and put their minds to more useful projects.

There have been exceptions – the Morris (or Internet) Worm that ten years ago (November 1988) saw a network virus paralyze many local networks in the USA. Since then no such incidents have been registered…

So, how can we protect ourselves? Use the standard features of browsers and emailers, configuring them in such a way that the risk of down-loading and executing the affected objects is minimized. Disable all browser and email features you do not really need – the unnecessary downloading of *Word* and *Excel* files is a bad idea.



You can specify different security settings for different "zones" of Web content.

Zone: Internet zone

Internet zone

This zone contains all Web sites you haven't placed in other zones

Set the security level for this zone:

○ High (most secure)
Exclude content that could damage your computer

○ Medium (more secure)
Warn before running potentially damaging content

○ Low
Do not warn before running potentially damaging content

● Custom (for expert users)
Base security on settings you choose

Internet Explorer users should check the Custom security settings, making them as conservative as possible.

All security configuration menu items should be set to 'High', except the ones you *really* need lower. That will protect your computer… for a while. In the near future anti-virus software will likely check the security level of installed Internet applications, and report any weaknesses.

[With special thanks to the Moscow Web development company *Actis* (http://www.actis.ru) as well as to Mr. Grigori Nikonov (gregoryn@actis.ru), for their help in studying the 'weak points' of modern Internet browsers.]

PRODUCT REVIEW 1

Calluna Hardwall

Another, less traditional product was selected for review this month. *Hardwall*, from Scottish *Calluna Technology*, was the centre of some attention at the Fall Comdex in 1998, when *Calluna* held a 'hacker challenge' at its booth. The product stood up to four days' worth of abuse from all-comers. In North America the same product is sold under the name *PC Bodyguard*.

Calluna (perhaps better known for its Type-III PCMCIA hard drives) claims that *Hardwall* 'provides a complete and risk free hardware solution to the growing problem of external attacks from viruses and hackers against desktop PCs'. *Virus Bulletin* felt that this rather bold claim should be put to the test.

Some History

Commercial anti-virus efforts have been dominated by software products. In fact, they have been dominated by software products dedicated to virus *detection* with the traditional software approach of scanning for known viruses. Forms of heuristic detection, behaviour blocking and the like have also enjoyed varying degrees of attention, but little market success on their own.

Scanning has several benefits over the more generic methods, prime among them being accurate identification of what did the 'attacking'. Coupled with the research that generates the detection information used in a scanner, the product developer often gleans much useful information that is valuable in subsequent clean-up efforts (should that be necessary).

For example, apart from adding their macros to documents, many macro viruses change registry settings, re-label the hard drive of the machine they are on, delete files or folders and so on. These changes can have quite widespread effects, far beyond the application hosting the virus.

Despite the domination of the anti-virus market by software, hardware-based products are not new to the field. However, previous entries in this market have met with little, if any, success. This is generally ascribed to the decreased convenience such products can place on those using (or, at least, maintaining) the systems so protected.

These limitations often arise from the hardware taking a 'dumb' approach, with it being expected to do all the protection. Such solutions tended to prevent some 'uncommon but normal' actions. Further, the arcane, roundabout processes often needed to bypass the ‘protection’, as required by normal system maintenance, have often mitigated against their widespread acceptability.

So, is *Hardwall* any different? Its approach ignores changes to most critical system areas during normal operation. That probably does not sound like a good security design, but the system (boot) partition can be completely rebuilt between machine restarts. This flexibility is coupled with a further, critical, design consideration: *Hardwall* strives to reduce transitive access between ‘normal work’ and possibly infective or damaging (‘hacking’) agents.

How it Works

Hardwall separates ‘work’ from ‘play’ (or ‘desk work’ from ‘web work’) by allowing the separation of data on different ‘user partitions’ and then enforcing access to those partitions one at a time (between restarts). Thus, the system can be configured with, say, an ‘Internet partition’ and a ‘work partition’. Once set up, material on one partition will not be able to be accessed (in fact, cannot even be seen) while another is the active user partition.

Note the ‘between restarts’ condition in the two foregoing paragraphs. *Hardwall* users who have been trawling the Net (say) and wish to get back to the accounting package have to restart the machine (in fact, a power-cycle is required) and select the user with access to the appropriate software and/or data file partition. Once done, the user partition containing the Internet downloads from the previous session is invisible and the accounts partition that was invisible when surfing is now available.

There may seem to be an obvious ‘hole’ in this scheme. While browsing the web, something could be saved to the boot partition, which *Hardwall* leaves fully writable at all times, then accessed from there and spread to another partition after a subsequent restart. The solution to this is, perhaps, the cleverest part of the product. It reserves a portion of the disk to record writes to the boot partition. These redirections are used for all subsequent disk actions during that session and any following a soft reset (one without a power cycle).

A power-cycled restart makes *Hardwall* clear its redirection of the modified parts of the boot partition and forces the selection of a user profile for the new session. Once a user has been selected from the menu following such a startup – even if it is the same user as in the previous session – the boot partition will have ‘reverted’ to its standard state.

The Package

Hardwall arrived for review in packaging similar to its software counterparts. The box was illustrated with a photograph of the card, artwork reminiscent of a recent television sci-fi series and some impressive-sounding claims you would probably dismiss if they had been made by Californian anti-virus companies. One side panel of the box clearly states the requirements for installation, while the opposite one lists the contents therein.

Opening the box divulged the contents claimed by the packing list – specifically, one *Hardwall* card, a 12" IDE drive cable and a CD containing the software and on-line manual. The desire to install the card into a test machine and start tinkering was resisted, and the User Guide (in *Adobe Acrobat* PDF format) was studied carefully first. It transpired that this was a good move, as installing the card into the host PC is virtually the final installation step!

The manual would be 45 pages if printed. Although it contains no Index, this is not a major hurdle to its successful use as an installation guide. As such it takes the usual approach of tracing the steps of installing and configuring the *Hardwall* card. The Table of Contents should probably be enough to jog one’s memory should subsequent reference to an important detail be necessary.

Most of the manual is given over to installation and configuration issues, save a two page glossary and a half-dozen pages introducing the product, its operational fundamentals and some expected uses for it. From experience, the warning in the second paragraph of the Before You Begin section cannot be emphasized enough – despite taking ‘every step to ensure that the installation process is as simple as possible [it] is quite complicated’.

Any process likely to involve on-the-fly partition resizing, the movement of files between existing and new partitions, and the like, has the potential for all manner of complex problems should anything untoward happen. This is a process that can cause serious trouble at many points. It is not to be undertaken lightly – as the manual says ‘any important information… already stored on the hard disk… should be backed up before beginning installation’. Having the luxury of a test lab and machines that can be restored to their standard configurations quickly from image backups, this advice was ignored and installation begun.

Installation

The installation and configuration software must be run before installing the *Hardwall* card into the host PC. The setup program’s first dialog is a stern warning to close all other programs before continuing. Given the nature of the procedures that are likely to follow, it is good advice.

Accepting that advice and continuing with the installation, the splash screen of the setup program was then displayed. This offered four actions – beginning or quitting the installation, and viewing frequently asked questions or product information. The last two are not at all detailed, being more ‘marketing’ than technically oriented.

The manual warns against installing the *HardWall* card before installing the software. The reviewer wondered whether it might not, in fact, be prudent for the developers to include a large warning at this point that the *HardWall* card should not yet be installed in the computer. Subsequent testing showed that the machine will not boot properly, with the card's BIOS extension complaining that the software is not installed properly.

Making the most of *Hardwall* needs a good understanding of drive partitioning, what *Hardwall* does with partitions and the uses of the host machine. The installation software requires a minimum of five partitions on the boot drive. These are the boot partition (which becomes Write-Many Recoverable or WMR in *Hardwall* terms), a partition that is read-only while operating in 'protected mode', two user partitions and the 'recovery' partition. Up to six further user partitions can be supported.

This requirement will seldom be met as most machines will have one partition filling the boot drive. Allowing for this, *Calluna* supplies a copy of *PartitionMagic Special Edition* (*PMSE*) from *PowerQuest* to assist with any repartitioning that may be required. This licensed copy of *PMSE* may be upgraded to the full version of *PartitionMagic*.

The *Hardwall* setup program responds differently, depending on the partitioning scheme it discovers on the host drive. Drives with a single existing partition (whether it fills the drive or not) have an alternative partitioning scheme suggested, based on splitting the existing partition into the minimum required number of partitions. If the suggested scheme suits, the installer can accept it and have *PMSE* launched in automatic mode, reconfiguring the partitions to the suggested scheme. If the suggested scheme does not suit some changes can be defined and automatic mode started, or the installer can opt to do this all manually in *PMSE*. The latter is the only choice for users with more than one but fewer than five existing partitions. Installation on drives that meet *Hardwall's* partitioning requirement offers the continuation of the installation or the option of manually reconfiguring with *PMSE*.

A hitherto untroublesome peculiarity in the partition table of *VB's* standard test-lab PCs caused *PMSE* to abort its operations on the test machines. It rather unhelpfully reported 'Partition table error #110 found'. The lack of numerical indexing of its error messages is a point of some annoyance, and something that *PowerQuest* should fix. *PMSE's* explanation of this error was eventually uncovered and the judicious modification of the partition table with a disk editor fixed the problem. This fix would probably be well beyond a 'typical user' yet seems to be caused by a bug in some versions of *FDISK*, so may not be uncommon.

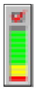
Once the drive was suitably partitioned, *PMSE* restarted the PC and the *Hardwall* setup program resumed. It offered options to install either or both of *PMSE* and the *Hardwall* software, followed by typical *InstallShield* dialogs seeking direction on which folders to install to and your agreement to various licence conditions. The obligatory bar graphs indicated progress and then the *Hardwall Configuration Manager* was displayed. This utility allows the different users to have various personal settings retained between sessions. Thus, items such as MRU lists for popular applications, Web histories and so on can be recorded to the current user's partition at shut-down and restored from there to the system partition or Registry when the user starts a new session.

Next, the system is shut-down and the *Hardwall* card installed. This requires a free 8- or 16-bit ISA slot to provide it with power. Once the card is installed, the primary IDE hard drive is connected to the card and the supplied IDE cable connected from the other socket on the *Hardwall* to the IDE adapter or motherboard.

Finally, the machine is restarted. The first sign of the card's presence is that after the usual BIOS tests are complete, a menu is displayed offering a choice of user (based on the names of the user partitions). At this point, unprotected mode *must* be selected so the software installation can complete without being removed at the next power-cycle.

Using Hardwall

Hardwall's user selection menu is displayed at startup, prior to the operating system loading. Apart from selecting one of the users in the list, pressing the F1 key and entering the password selects unprotected mode, which allows full access to the whole disk. The main indicators of *Hardwall's* presence on *Windows 9x* machines are an icon in the system tray and the WMR meter, which shows how much of the recovery partition is still available.



Further, warning dialogs are presented when the contents of the boot partition are affected by standard file system APIs (file create, write, delete or rename; change attributes or time-stamp; directory create or remove). Similar warning dialogs are displayed under *NT*, but neither the meter nor system tray indicator are present. The warnings can be disabled and re-enabled easily under *Windows 9x* but there was no obvious method of such control under *NT*.

Testing

Hardwall is not an anti-virus product. It prevents anything from viruses and other forms of malicious 'assault' to installation programs from accessing certain parts of your disk. It also provides a measure of intransitivity between partitions. It will not prevent viruses (or other attacks) from affecting the current user or system partitions. In the case of the system partition, this is fully reversible by simply power-cycling the machine. Doing so resets all changes to the boot partition, but leaves the user partition as was.

All manner of disk trashers (FORMAT, DELTREE and various Trojan Horse programs), viruses and system reconfigurations were thrown at a *Hardwall*-protected system. It performed as claimed – preventing access to currently locked-out partitions and the boot track of the disk, but otherwise allowing 'normal' things to happen (even though they may be undesirable).

The monitoring software warns of changes to the system partition. This has multiple uses, monitoring what is placed where during software installation and highlighting ‘odd’ disk accesses (why do the *Microsoft Office* products write temporary files in so many other directories when a perfectly adequate, system-wide TEMP directory exists?) and alerting to potentially harmful actions.

As an example of this last use, DMSetup was monitored. Watching the flood of warnings for all the directories being created (with such unusual names), it was difficult to imagine anyone leaving it running until the disk was filled. The warnings quickly became onerous, so they were disabled for the remainder of that test. The usefulness of this feature would be greatly improved with two additions. One is an option to filter events by type (and possibly location) and display only those events. Further, being able to log (filtered) events to a file rather than a dialog box would improve this feature out of sight. A scrolling event list rather than the current sequence of discrete warning dialogs would also be an improvement. [*The developers plan some or all of these for the next release. Ed.*]

Several other ‘abuses’ were rendered unto the C: drive – it was converted from FAT16 to FAT32, DELTREE /Y C:\*.* was followed by a utility that overwrites all free space on the drive with nulls, and following a diskette boot the drive was unconditionally formatted. In all these, and the earlier cases, a power-cycled restart saw the return of the original drive, unmodified apart from minor changes in the registry due to things that are altered at start-up and vary with how long the machine has been running.

Monitoring requests between the controller and the drive, questions of overhead have to be considered. *WinTune 98* from *Windows Magazine* has two throughput tests – cached and uncached. The former mainly reflects disk cache performance while the second bypasses the cache entirely. Both were run on the test PC prior to installing *Hardwall*.

The tests were repeated with *Hardwall* installed and running in unprotected and protected mode. This was all retested with a second drive. In unprotected mode, throughput on the cached test was immeasurably lower than the baseline. In protected mode, cached performance for both drives was approximately 26 MB/s – down 25%. Uncached performance was less consistent across the drives. Following a fresh start with a new user, protected mode showed near-identical performance despite one of the drives being 25% faster without *Hardwall* installed at all. This suggests a processing bottleneck at the *Hardwall* card.

Is Hardwall for You?

Effective use of *Hardwall* comes at some cost. The device itself is not cheap. However, it is cheaper than obtaining a second machine and can be configured to provide much the same level of separation between ‘work’ and ‘play’, ‘safe’ and ‘dubious’, ‘secure’ and ‘less so’ as a two-machine setup provides.

The main differences between a *Hardwall* and a two machine setup are that the latter allows easy and guaranteed separation of network interfaces and physical isolation of the two environments, plus the ability to use both configurations concurrently. Those with the discipline to use a two-machine setup should not find a *Hardwall* machine unduly troublesome unless in the habit of running large downloads from the Net on one machine while working on their ‘secure’ machine. Others may find *Hardwall* invasive.

Therein, the rub. Ultimately, good PC security involves a degree of discipline from the users of the system. As PCs are seldom a part of highly-sensitive, centrally-controlled systems, that discipline cannot be completely enforced on their users. *Hardwall* offers a partial solution and in many cases may be more than enough, yet the lack of password protection on the ‘users’ of a *Hardwall* system seems to be a major shortcoming. [*Again, promised for the next version. Ed.*] Little Johnny can elect to be the Accounts ‘user’ just as easily as he can select his own user partition.

Closing Comments

In many ways, if you want to be virus-free, *Hardwall* is not for you. However, if you wish to reduce the risk from less security-conscious users and can stand the overhead, it is worth considering. Do not be fooled – for actively used and updated PCs this is not a ‘set and forget’ product, and it may incur additional maintenance overheads, depending on the complexity of the policies you wish to enforce with it.

Home users with children may be well-served by *Hardwall*, although may be most poorly situated to deal with its complexities. System administrators testing ‘dubious’ software could make good use of a *Hardwall* machine or two, as it allows for much faster restoration to a base configuration than image backups etc. Although possibly appealing to school and university lab administrators, they would be better served by systems with built-in security that can be administered remotely. All this said, *Hardwall* works as claimed and cannot be faulted for that.

Technical Details

Product: *Calluna Hardwall*.

Vendor: *Calluna Technologies Ltd*, 1 BlackWood Road, Eastfield, Glenroathes, Fife KY7 4NP, Scotland. Tel; +44 1592 630810, fax; +44 1592 630920, email; sales@calluna.co.uk, WWW; http://www.hardwall.com/.

Version: 7.0.

Availability: One free ISA bus slot, 4 MB disk space plus partitioning requirements and *Windows 95/98* or *NT v4.0*.

Price: SRP £169 + VAT (for introductory pricing, contact dsmith@calluna.co.uk).

Test Environment: 166 MHz Pentium-MMX with 64 MB of RAM, 4 GB hard disk, CD-ROM drive and a 3.5-inch floppy. The machine can be configured to run *Windows 95, 98* or *NT*.

PRODUCT REVIEW 2

Reflex Macro Interceptor v3.04

We continue our recent review interest in macro virus-only products by examining *Reflex Macro Interceptor* (*RMI*). For those who wish to build their own layered virus defence or see macro viruses as their only virus concern, *RMI* is designed for use on workstations, but has centralized installation and macro authorization features. It is a natural ally to *Reflex's* diskette authorization product, *Disknet*.

RMI is actually *Leprechaun Software's Macro VirusBuster*, repackaged and mostly re-labelled. Some of the registry entries created during installation make this clear (if you look for such things), as does an error message (see below) and the About dialog accessed from the System Tray icon.

So, What do You Get?

The review copy arrived on a single 1.44 MB write-enabled diskette. It was accompanied by a slim, 64-page, soft-covered, lay-flat booklet. As manuals go, this takes a straightforward, no-fuss approach to its task. Following a brief introduction and a description of some complementary *Reflex* products, the first chapter describes installation of the program while the second covers configuration and use.

The first, and substantially shorter, of the appendices outlines the configuration of *RMI* for installation from a server and explains how this can be completely automated so as not to require any user input. While this functionality is provided by *InstallShield*, it is an elegant procedure compared to some and it worked well in a simple trial.

Appendix B describes the slightly grander task of distributing *RMI* from an *NT* server using *Seagate's WinINSTALL*. This is an interesting-looking exercise, but was beyond the scope of testing for this review. A four page index rounds off the manual.

The layout of the manual was logical and easily followed. The screen shots were a little grainy, but good enough given that they are mainly guides for those following the same process on-screen during installation. Some of the content suggested that perhaps the *Leprechaun* manual had been the base work – a reference to a new command-line switch in the DOS program seemed out of place in this strictly *Windows* GUI application. Also, some index entries referred to other, non-existent ones.

Installation

The manual claimed *RMI* ran on *Windows 95* and *NT v3.51* and later. Being a trusting soul, your reviewer decided to try them both and – with an eye to adventure – *Windows 98* and *NT v4.0* as well.

Under *Windows 95*, after reaching the registration confirmation dialog, the setup program complained of being 'unable to load a required file' and suggested the installer 'contact Leprechaun Software Technical Support'. Dismissing this error dialog dropped back two steps in the installation process. One could seemingly cycle through this as often as desired, but it rapidly became uninteresting…

Reflex technical support expressed surprise at this, but investigated. The suggested workaround failed. It would have been problematic for some anyway, as it depended on copying one or two files from a successful installation, giving rise to a possible 'chicken and egg' situation.

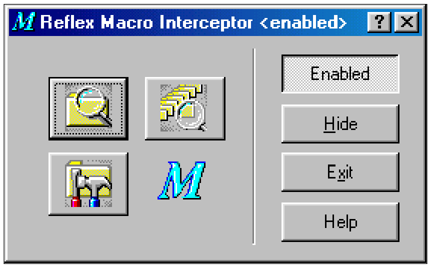
The neighbouring *Windows 98* test machine followed a route typical of *InstallShield*-driven installations. After a restart, the *RMI* icon appeared in the System Tray – double-clicking it produced the main *RMI* dialog window.

Installations to both *Windows NT v3.51 (SP4)* and *NT v4.0 (SP3)* were tried, and a series of problems encountered. Despite *NT v3.51* being mentioned as a minimum OS requirement, the setup program warned that *Windows 95* or *NT v4.0* or higher was required. On *NT v4.0*, Setup finished without complaining it was incomplete – surprising given that three 'severe' errors appeared during installation! After a system restart however, the service at the heart of the on-access detection component would not start properly, thus *RMI* was not running. Calls to *Reflex* elicited the information that the manual was wrong to claim any *NT* support. A new version, 'in testing' was expected to rectify this.

Configuration Options

Once installed and active, *RMI* normally presents itself as an icon in the System Tray. Right-clicking this allows access to the main *RMI* screen. This context menu also provides options to deactivate the scanner, start a manual scan and view product 'about' information.

RMI's main dialog presents a minimalist interface, with seven buttons. Help and Exit should be self-explanatory. Hide closes the dialog box and the Enable/Enabled button toggles active scanning on and off, changing its text label to indicate its current status. Of the larger buttons, the upper two provide different mechanisms for selecting drives, folders or files to subject to a manual scan. One opens a typical browse dialog while the other starts a new instance of Explorer.



The final button on the *RMI* dialog opens the configuration screen. Access to this can be password protected (but easily disabled via registry editing). Of note here, and possibly annoying, is that the only settings for which files to scan are entirely defined by the developers. This is also where automatic disinfection is enabled.

The Explorer context menu extension did not work in the review copy. An interesting alternative for mouse jockeys is that the main *RMI* dialog is a drag-and-drop client. Thus, an object can be dragged from Explorer and dropped there to invoke a scan of it.

The Advanced Options dialog has controls allowing *RMI* to be disabled and unloaded, enabling display of an icon in the System Tray, prohibiting the launching of Explorer and so on. There are also options affecting the 'severity' of *RMI's* detection and cleaning such as enabling heuristics, alerting on any macro and enabling 'minimal cure' where just viral macros (rather than all macros) are deleted.

Logging options are typical. Name and maximum log file size can be set, and there is a choice of overwriting or appending an existing log. By default all the loggable options –user actions, statistics, detections and cures – are enabled, but any combination that suits can be selected.

Virtually all configuration options can be preset so workstation installations rolled-out from a server have the desired settings. A detail here is that the key name for the 'overwrite log file' option is not the documented LogOverwrite , but LogOverwite. The author of the installation process however, implemented support for the documented name, so this option cannot be customized with a scripted setup.

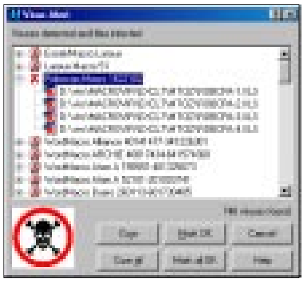
A *Reflex* representative suggested that 'detect all macros', although not the 'out of the box' default, should be enabled during testing. He claimed that most *RMI* customers use it that way. Given that *RMI* is mainly used as an adjunct to *Disknet*, it is probably used in somewhat 'more sensitive' environments than other scanners. Thus, a preference to detect any code entering an organization, rather than just what is known to be viral, would be seen as desirable, even at the expense of increased false alarm rates.

So, Does it Detect Viruses?

The manual and on-line help only mention *Word* and *Excel* macro viruses, and a quick test revealed that *RMI* did not seem at all interested in Access files. It missed all samples of the AccessiV variants in the Macro test-set.

Tested against only the macro viruses in the *VB* ItW test-set, *RMI* detected 96.5% of them in its default mode. The heuristic analyser is obviously important, even to this score, as disabling 'detect unknown viruses' lowered this rate to 43.0%. Under the 'all macros' option detection improved to 97.6%. Testing against the complete Macro test-set, 98.3%, 35.0% and 94.8% were achieved using the same test conditions respectively. The most 'troublesome' virus was XF/Paix, which is quite common in parts of France – all samples were missed with all options.

This is a respectable performance, but suggests that the 'consider any macro a virus' option is necessary if high detection rates are required. Use of that option could be problematic in environments where macros are regularly used, although this may be ameliorated by *RMI's* macro authorization function. Its name is also potentially misleading, as running under it, *RMI* ignores 'templates' (DOT and XLT files) that contain macros if they do not trigger the known virus detector or the heuristic analyser. This is documented, but the implications not clearly explained.



When a virus is detected, several treatment options are available. Files may be cured (all macros deleted by default) or marked 'OK'. In the latter case, the macros are 'remembered' and not alerted in future unless they turn up in other documents.The data file used for this need not be stored locally, although this option has to be divined by registry trawling or from technical support, as it is not mentioned in the manual nor is there an interface to this setting in the program.

In Summary

Assuming the installation problems are resolved, this could be a solid, if unspectacular, macro virus-only product. *RMI* does not provide the wealth of analysis of the *Portcullis* product (see *VB*, July 1998, p.18), nor the management convenience of that, or the *Data Fellows*, product (see *VB*, October 1998, p.21). In high-paranoia environments where products like *Disknet* are popular, an option to 'detect all macros' would probably be expected to detect *all* macros... If interested in this class of product, watch for future revisions, but right now *Reflex Macro Interceptor* feels and acts like a product a little short of maturity.

Technical Details

Product: *Reflex Macro Interceptor*.

Vendor: *Reflex Magnetics Ltd*, 31–33 Priory Park Road, London, NW6 7UP, UK. Tel +44 171 3726666, fax +44 171 3722507, email sales@reflex-magnetics.co.uk, WWW http://www.reflex-magnetics.co.uk/.

Version: 3.04.

Availability: 1.2 MB free disk space and *Windows 95/98*.

Price: 5–99 PCs £29 each; 100–250 PCs £17 each.

Test Environment: 166 MHz Pentium-MMX with 64 MB of RAM, 4 GB hard disk, CD-ROM drive and a 3.5-inch floppy running *Windows 98*.

PRODUCT REVIEW 3

In-Defense v2.10 – Part 2

[*This forms the concluding part of the review of* In-Defense, *started in the November 1998 issue. Ed.*]

Following the first part of this review, *Tegam* contacted *VB*, concerned that the wrong product version had been supplied for testing. *VB* is very cautious of accepting non-shipping versions for its standalone reviews, so the correct version was dispatched. It was unusual that this happened. Further, the original product was a manufactured (aluminium and printed) CD in a proper box. Its replacement was a CD-R disk with a stick-on printed label.

Detection Tests

Despite having some scanner-like properties, *In-Defense* cannot be fully tested against a disk full of viruses then examining the results. It will only detect some viruses through their actions or from integrity checking techniques.

Approximately one-third of the ItW Boot infectors were missed by *In-Defense's* on-access analyser. However, as with other aspects of the product mentioned in part one of the review, performance here was unstable. For example, a Diablo-infected diskette would be consistently alerted unless it happened to be presented for checking after an Eco.B sample, when it would always be ignored. After many such results, the reviewer was left wondering if some of the samples recorded as *detected* may have been dependent on which sample was presented immediately before!

A test PC was infected with Baboon, which was missed in 'scanner mode'. Restarting from the hard drive saw its presence missed by the start-up checks, however, manually running the same checks inside *Windows* triggered the expected alerts. That is very poor for an ItW boot infector with a damaging payload (see *VB*, November 1997, p.13).

A clean image was infected from a Hare.7786 boot. On restart various changes were noted and remedies rendered. Sadly, some infected EXEs were simply noted as having changed, but these were not deemed viral changes, so it automatically 'revaccinated' them (i.e. updated its database to reflect their infected state). A Bap.1536 infection was noted at boot and all files it had infected were flagged as 'increased by 1536 bytes and cleaned'. In neither case was it indicated that HSFLOP.PDR, the 32-bit floppy driver, had been deleted. *In-Defense* does not consider these drivers 'executable' so ignores their existence and integrity.

In-Defense does not detect macro viruses, but does detect macros – not all macros (although that option exists), but more than enough to be a nuisance. Initial testing suggested the user would be alerted to documents with auto- or system macros and given one of two warnings – 'New document with automatic macros detected' or 'Viral macros detected'. In the first case, options of eliminating or accepting the macros, or cancelling the warning, were given. In the second, accepting the macros was not an option.

This was unfortunate, as the 'powerful artificial intelligence analyzers' boasted of in the manual often let the product down. For example, a *Word* file containing a macro named Payload and any auto-macro would be claimed viral, even if each macro contained only a MsgBox statement. That may seem an unlikely scenario, but many people implemented something similar as an early defence against Concept. Virus detection based on macro names cannot honestly be labelled 'intelligent'. Further, some forms of auto-macros, as used in some of the newer class-infectors, were never flagged as being of concern.

In Conclusion

Some industry insiders believe a better approach to virus detection than scanning should, in practice, be achievable. Many customers are increasingly dissatisfied with the cost of continual updates. Unfortunately, *Tegam* has failed as convincingly as others before it to provide a solution. The claim of 'total protection' is a non-starter – it always has been and always will be. It should be hoped that by the late 1990s this would be understood by any company wishing to be taken seriously in the anti-virus marketplace.

In-Defense's approach is 'holey' and too often leaves it to the user to decide the appropriate action. This means it might be a handy tool for an experienced technician faced with a new virus the current scanner does not detect, but is likely to be self-defeating with 'ordinary users'. Instead of holding out the hope of universal detection, *Tegam* should sort out the worst shortcomings of the product, then market it more realistically. There *are* users willing to forego unending updates, so long as they receive good protection – for now, *In-Defense* cannot be recommended to them.

Technical Details

Product: *In-Defense for DOS/Win 3.x/95/98*.

Developer: *Tegam International*, 303 Potrero Street # 42-204 Santa Cruz, California 95060-2780, USA; Tel +1 831 4711413, fax +1 831 4201313, email sales@indefense.com, WWW http://www.indefense.com/.

Availability: 386 with 8 MB of RAM, 6 MB disk space.

Version Evaluated: 2.10.

Price: Single user $79. Multiple and site licences are available.

Hardware Used: Three 166MHz Pentium-MMX PCs with 64 MB RAM, 4 GB disk; one *Compaq* DeskPro 575 with 80 MB RAM, 2 GB disk; one *Compaq* DeskPro XE 466. All have 3.5-inch floppy and CD-ROM drives and connect to a UTP hub. They can be variously configured to run DOS, *Windows 95*, *98* and *NT v4.0*, *NetWare v3.12* and *v4.10*, and *NT Server v4.0*.

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

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 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

USENIX has issued a call for papers to those working in any practical aspects of security or applications of cryptography. **The 8th *USENIX* Security Symposium is to take place from 23–26 August1999 in Washington DC, USA** and the submission date is 9 March 1999. The event is planned around two days of tutorials followed by two days of technical sessions, papers, talks, works-in-progress, panel discussions and a product exhibition. Find more details about the conference and the call for papers at http://www.usenix.org/events/sec99/cfp/.

***Sophos* will be hosting an introductory computer virus workshop on 17 March 1999 to be followed on 18 March by an advanced session.** The two-day course will be held at the organization's training suite in Abingdon, UK. To register for a place, contact Karen Richardson; Tel +44 1235 544015, fax +44 1235 559935, or find more information at http://www.sophos.com/.

***Network Associates Inc* is to host a two-day live virus workshop from 23–24 February 1999.** The sessions are to take place at the *NAI* Training Centre in Aylesbury, UK from 9.30am to 4.30pm. For more information contact Caroline Jordan; Tel +44 1296 318881 or email caroline_jordan@nai.com.

***eicar's* 1999 conference 'E-Commerce and New Media: Managing Safety, Security and Malware Challenges Effectively' is to be held in Aalborg, Denmark from 28 February–2 March.** Two workshops take place on Sunday 28 February – 'Encryption and Privacy: The Global Policy Disorder' in the morning, followed by 'Managing Privacy and Security Software, Systems Management and Policy Issues' in the afternoon. *eicar* working groups are to meet from 17.00–18.30 that day. Delegates are reminded that they must pre-register with *eicar* for all the meetings and workshops. The conference itself will be opened by Rainer Fahs, chair of *eicar*, on Monday 1 March. A three-day exhibition starts on Sunday 28 February at 10.30am. For further details, contact Professor Urs E Gattiker of Aalborg University; Tel +45 96358962, fax +45 98153030, email Urs_the_Bear@bigfoot.com, or visit http://www.eicar.dk/.

Ed Wilding is hosting a one-day seminar, **'Investigating Computer Crime and Misuse', at the Mayfair Conference Centre, London, UK on Wednesday 24 March, 1999.** Contact; Tel +44 1572 757751, fax +44 1572 757752 or email ComSem@compuserve.com.

Command Software Systems* announced the release of *Command AntiVirus for Lotus Notes in mid-January 1999. The company claims that its HoloCheck scanning technology secures all virus entry points, including email, database and replications activities. For more information contact Esther Swann at *Command*; Tel +1 561 5753200, fax +1 561 5753026 or email eswann@commandcom.com.

At the end of January 1999 ***Data Fellows* and *Computer Associates International Inc* (*CA*) announced a technology development partnership.** *Data Fellows' F-Secure Workstation Suite* is to be integrated with *CA's Unicenter TNG*. For details contact Petri Talala, the firm's Development Manager in Finland; tel +358 985990501, fax +358 985990701 or email Petri.Talala@DataFellows.com.

WebSec'99 is to be held at the Mount Royal Hotel in London, UK from 23–25 March 1999. Optional pre- and post-conference workshops will run on 22 and 26 March. For further information on either the conference or the concurrent exhibition contact the organizers; Tel +44 171 7798944, fax +44 171 7798293, email misuk@misti.com or visit http://www.misti.com/.

The 9th *Computer Security Institute* (*CSI*) Annual Network Security Conference, *NetSec'99*, is to be held from 14–16 June, 1999, in St Louis, Missouri at the Hyatt Regency Hotel. Over 1500 computer and information security professionals are expected to attend the conference and its concurrent exhibition. For a new calendar of events or more details on the conference, contact *CSI*; Tel +1 415 9052626, fax +1 415 9052218, email csi@mfi.com or visit the *CSI* web site at http://www.gocsi.com/.

***VB* is currently seeking a technical consultant** for an immediate start at its Abingdon office. The ideal candidate must possess a good knowledge of computer viruses, web design (HTML), and popular operating systems and networks. A working knowledge of *Adobe PageMaker* and the *Microsoft Office* application suite would be an advantage. Duties include all the in-house product testing and comparative review procedures, maintenance of the *VB* web site, and compilation of the monthly prevalence table. This position also supplies technical support for *Virus Bulletin* subscribers. Contact Francesca at *VB*; tel +44 1235 555139, fax +44 1235 531889, or email editorial@virusbtn.com.