ISSN 0956-9979 NOVEMBER 1998

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **In it to Win98 it:** Twenty anti-virus products for the new *Windows 98* platform line up in this month's comparative review, starting on p.12.

• **Does size matter?** What would you do if you received a brand new virus collection of over 14,000 samples? *NAI's* Peter Morley meets it head on in our feature on p.10.

• **Monster München:** VB'98 went with a swing in the Munich Park Hilton. If you missed it, or if you want to relive old memories, turn to p.6.

CONTENTS

EDITORIAL

Totally Bogus?

‘Detects 100% of viruses’, ‘Intercepts and disinfects all viruses, past, present and future’, ‘Complete virus protection’, ‘Total virus defense’ – we’ve all seen and heard such claims.

We also do not believe them… I hope!

So why do the marketeers insist on making these claims? Do they think that anti-virus software buyers less sophisticated than those who read *VB* are more likely to be enticed to purchase their products if they claim to do that which we know is impossible? Do they really have that little respect for their (potential) market?

“ we know what you meant, but that is not what you said ”

A product reviewed in this issue (*In-Defense*, p.20) claims to detect all viruses, including those not yet written, through a combination of new approaches (not scanning) and ‘artificial intelligence analyzers’ that detect virus behaviour. Such claims have been made for other ‘revolutionary’ products in the past. All those products have failed to live up to their claims. Cohen’s doctoral research shows that the task of programmatically determining whether an arbitrary program is viral is impossible to achieve reliably (at least, within a finite time frame). This fundamental finding suggests the designers of *In-Defense* and similar products are tilting at windmills.

But what about the scanner developers? Simple string scanning was, in most cases, replaced years ago by more sophisticated scanning, focusing on a program’s entry point (where code execution will begin), ‘top and tailing’ methods (most parasitic program viruses attach their code to either the beginning or end of the host), and so on. Code emulation helps with polymorphic viruses and many developers include heuristic methods in attempts to detect newer and more esoteric viruses.

With such an array of detection methods at hand, scanners often detect substantially more than just the viruses known to the developers at the time a particular revision was completed. They will also detect some newly written viruses – both minor variants of existing viruses and completely new families. Perhaps that is what prompted a *Symantec* marketeer to gush effusively over *NAV’s* ability to ‘hunt down and obliterate viruses before they’re released’. Another case of ‘we know what you meant, but that is not what you said’. In this case the wording carries a very unfortunate implication for the industry (or, at least, for *Symantec*) – that the industry produces viruses.

I have discussed the vagaries of marketing-speak with senior product and marketing managers at *Symantec*. They seem to have come to an understanding of how I would rather that they did not misrepresent what *VB* test results mean when writing advertising copy. Let’s hope they extend this enthusiasm to other anti-virus marketing efforts. I guess it is a small blessing that the claim was not that *NAV* will detect *all* viruses before they are released.

And what of *NAI’s* claim to provide Total Virus Defense? It is literally true: *NAI* has a product called that. However, the name is more than just a handle, an identifier for the product. In this case it strongly implies something about the product. The implication is that the product is, what is known inside the industry as, TOAST – The Only Anti-virus Software That you will ever need.

‘Total’ is a dangerous word – it is an absolute. Zero degrees of freedom…

With *NAI’s* increasing focus on providing tools for managing all aspects of the corporate network, the ‘total’ may be intended to refer to the ‘whole package’, the ‘complete kit and caboodle’. However, as a major anti-virus vendor, it suggests something more – the claim of complete virus detection. I do not know for sure what *Copithorne & Bellows – NAI’s* UK public relations firm – uses for virus protection, but whatever it is, it hasn’t prevented them sending *VB* a Laroux infected spreadsheet of pricing details (see *VB*, June 1998, p.5) and this month a press release infected with W97M/Class.B. Whatever they use, it does not provide a total virus defense.

Maybe the product they use is heading to toast…

NEWS

Intel-ligent Manoeuvre

Proving that there is still scope for surprises in this market sector, *Symantec* has purchased *Intel's* anti-virus business. This late-September move substantiated earlier, stifled rumours, starting with the *Computer World* (New Zealand) article 'Further Development of LANDesk May Go To Partners' of 20 July 1998.

As part of the agreement, *Symantec* will license *Intel* systems management technology and inherit *Intel's* 18,000 registered anti-virus customers. Like *IBM* before it, *Intel* will now no longer market anti-virus products, and will recommend that customers purchase *Symantec's Norton Anti-Virus* (*NAV*) product range.

Technology from the *NAV* engine will be integrated into *Intel's* existing *LANDesk Management Suite* and, more significantly, into a new line that *Intel* has had under development for most of this year. Attendees of the *Virus Bulletin* conference in Munich at the end of October may have heard Cindy Snow's paper on the product's architecture. [*VB'98 conference proceedings are available on CD; for details email Joanne.Peck@virusbtn.com. Ed.*]

Symantec aims to use *Intel's* systems management technology to help build the Digital Immune System that it is developing with *IBM*. Customers migrating from *Intel* to *Symantec* products will receive support directly from *Symantec* during the transition period ▮

Ad Nauseam?

The Ottawa-based *Corel Corporation* is the latest in a seemingly long line of manufacturers to issue virus-infected CDs. In this case, the second pressing of *CorelDRAW 8.0 for Mac OS* was the bearer of 'unintended gifts'. During the first week of October, *Corel* confidently maintained it had retrieved 95% of the CDs infected with the D strain of AutoStart 9805 (see *Virus Bulletin*, July 1998, p.6). Unlike some recent cases of infected CDs, *Corel* took the immediate action of recalling all outstanding CDs from the affected batch and issuing a public statement – for details, check http://www.corel.com/draw8mac/virusinfo.htm/.

Despite being 'extremely concerned', *Corel* nevertheless played down the effects of the worm, which removes earlier versions of itself and does not intentionally damage files. Just the right degree of 'concern' was perhaps provoked by claims that this variant is 'not detected by many popular virus-checking utilities'. *Virus Bulletin* considers this judgement to be unlikely, given that this virus was covered in the July issue of the magazine, which was prepared in mid-June. Worried users of the program, which, we are assured, 'continues to sell well', are asked to contact *Corel* customer services on +1 800 7726735 ▮

Prevalence Table – September 1998

Virus	Type	Incidents	Reports
Laroux	Macro	41	18.7%
Win95/CIH	File	35	16.0%
Concept	Macro	16	7.3%
CopyCap	Macro	10	4.6%
Class	Macro	9	4.1%
Paix	Macro	9	4.1%
AntiEXE	Boot	8	3.7%
Cap	Macro	8	3.7%
MDMA	Macro	8	3.7%
Form	Boot	7	3.2%
AntiCMOS	Boot	6	2.7%
Wazzu	Macro	6	2.7%
Monkey	Boot	3	1.4%
Ripper	Boot	3	1.4%
Stealth_Boot	Boot	3	1.4%
Temple	Macro	3	1.4%
WelcomB	Boot	3	1.4%
Angelina	Boot	2	0.9%
Appder	Macro	2	0.9%
Junkie	Multi-partite	2	0.9%
Marburg	File	2	0.9%
NightShade	Macro	2	0.9%
Others [1]		31	9.2%
Total		219	100%

[1] The Prevalence Table includes one report of each of the following viruses: ABCD, Baboon, Bleah, Bye, Cascade, Compat, CountTen, Edwin, Generic_Boot, Goodnight, Groov, Habir, HLLP.4080, Int_AA, Int40, Ivana, mIRC/Goaway, Neuroquila, NF, NightFall, Nop, Nottice, Npad, NYB, Pccpo, PingPong.B, ShowOff, Techno.1123, Teocatl, Tubo and Unashamed.

Data Diddlers

The worst nightmare of an accountant storing important data in spreadsheets is that a small number of cells may be unintentionally changed. A whole discipline of designing spreadsheets to cross-check for such things exists, but many business-critical sheets are not designed following these principles. A new *Excel* macro virus exploits this.

XM/Compat has a payload that randomly selects a small number of cells in a sheet and if the selected cells are numeric, it randomly alters their value by a small proportion (within ±5%). Users infected with this virus should be aware that this data-diddling payload can affect files that the virus has not infected ▮

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

Type Codes

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

Aardwolf.448

CN: An appending, 448-byte virus with the texts '[Aardwolf] Type.A' and '(c) Copyleft 1998 by Crom/CVC,Corea'. Infected files have the byte FFh at offset 0003h.

```
Aardwolf.448        3DF1 F175 05B8 F2F2 9DCF 601E 0680 FC4B 7403 E9AC 001E 33C0
```

Ale.1911

CN: An encrypted, appending, 1911-byte, fast, direct infector containing the texts 'Ligue Para esta Puta: Viviane', 'Alevirus 97 !!!!!!!!!! Call Now 743-4915 many files from virii service', 'Sao Caetano do Sul', 'Brasil!', and 'Aberto das 0:00 ate 6:00 A.M'. The payload displays the above texts with the message 'ALEVIRUS CORINGA' (in text-mode graphics) and corrupts the CMOS data.

```
Ale.1911            E817 005B B409 80C4 37B9 7707 8D96 0B01 CD21 E805 00E9 1EF9
```

AsciiV.613

CN: An overwriting, 613-byte, direct infector which infects one file at a time. The virus code can be typed in the text editor since all bytes representing virus instructions and data are in the standard ASCII range. Before running the infection routine the virus constructs the procedure in memory. This template can be found in all infected programs but using more sophisticated detection methods would be safer.

```
AsciiV.613          3244 3530 4435 3244 3558 5048 3535 3535 4A4A 5059 5850 3530
```

Cpp

CR: Two appenders which install in the Interrupt Vector Table. The 231-byte variant contains the text 'CPP'94', whilst the 243-byte one has '.COM'. Infected files have the byte 5Ah ('Z') at offset 0003h.

```
Cpp.231             B440 B1E7 CD21 E823 008B FAB0 E9AA 5840 ABB0 5AAA B440 CD21
Cpp.243             B1F3 E874 FFB0 E9AA 5840 ABB0 5AAA E868 FFB8 0157 595A CD21
```

Grade.956

CER: An appending, 956-byte (EXE) and 963-byte (COM) virus containing the texts 'Dedicated to Johny.', 'CAPCANA', 'COMMAND', 'CLEAN', 'GUARD', 'ANTI', 'FIND', 'HIEW', 'HOST', 'KILL', 'SAFE', 'SCAN', 'STOP', 'TRAP', '[School.1st_grade] by [LittleJ], (R) 1996', 'ANL', 'NOD', 'RAV', 'VIR', 'F-' and 'TB'. Infected files have the word 2E4Ch ('L.') at offset 0012h and the word 2E4Ah ('J.') at offset 001Ah.

```
Grade.956           3D34 6275 059D BBFE FFCF 80EC 4080 FC0B 7506 0AC0 7502 EB06
```

HLLC.Godsquad

EPN: A companion, fast, direct infector containing the texts '*.EXE', '.exe', '[Smuilt / DIFFUSION]', '[God Squad aka SPAWN.B]', 'You've got %s, by %s.', 'It's Smuilt's birthday today, but that only narrows me down to about 279452 people.', '.bin' and 'run.exe'. The virus copies original *.EXE files to *.BIN and writes its own code to *.EXE. The virus displays the messages on 22 June and 29 August. Reliable detection of this virus requires use of a longer than usual template or other means of detection.

```
HLLC.Godsquad       807E FE1D 7510 807E FF08 750A B808 0150
                    E8AC 0C59 EBF6 FF36 8206 E8E7 0759 8BE5
```

Hernani.321

CR: A prepending, 321-byte virus containing the text ' [HERNANI by Int13h] * Paraguay '97 * Victor Hugo rules '. Infected files have their time-stamps set to 60 seconds.

```
Hernani.321         4299 2BC9 CD21 B440 BA00 01B9 4101 030E 4501 CD21 B801 57BA
```

Knave.534

EN: A 534-byte direct infector which infects three files at a time. It contains the texts '*.c?m' and '[Knave] Type.A by Red_Devils/CVC,Corea 98/04'.

```
Knave.534           B440 B916 02BA 0000 CD21 B000 E826 00B4 40B9 0400 C606 2102
```

Levitate.800

CR: An encrypted, appending, 800-byte virus containing the text '!!! Attention !!! Welcom to the LEVI Virus I'm sorry for you —> =:('. Infected files have their time-stamps set to 62 seconds.

```
Levitate.800        E87E 005E B8FF BBCD 213D BBFF 7460 8CD8 488E C026 803E 0000
```

LittleDevil.1981

CER: An appending, 1981-byte virus with the texts 'COMSPEC=', '03/25/97', 'Your computer is infected by the LITTLE DEVIL (v1.0b) virus You are very lucky that your computer is infected by the B version. if it was the A version your harddisk was completely destroyed == Date: November Monday ==', '== Generaton: no.255 ==' and '[MADE FOR VIRUSSTUDY ONLY]'.

```
LittleDevil.1981    B90D 00FC F3A4 B440 B9BD 0733 D2E8 F4FD 7303 E9CE 00B8 0042
```

Llap.791

CN: An encrypted, prepending, 791-byte direct infector with the texts 'Death has entered the world, prepare to die !', 'Invented in 1996 by UFP Headquarters', 'Live long and prosper!!!' and '*.COM'.

```
Llap.791        0609 01B4 3033 0602 012E 8036 0D01 062E 8905 4B83 FB00 75DE
```

Next.1721

CER: A stealth, encrypted, appending, 1721-byte virus containing the texts '(Type_E/Last Ver.) VIRUS...', ' (c) KOV (Knight Of Virus)/ Corea 9192/04/02', 'HWF-TBCLCOWCTKDEIBT2 ' and 'NEXT'. The latter is stored in the text-mode graphics. Infected files have their time-stamps set to 58 seconds. The following pattern detects the virus in memory only.

```
Next.1721       E8D1 005B B440 B9B9 06BA 7407 CD21 C3B4 4233 C933 D2CD 21C3
```

Nomad.888

CN: An 888-byte virus which infects one file at a time and contains the texts '*.exe' and 'yO!!! I could have made some mischief to you but I * * lEfT it out. I'm the #Nomad Virus# - Mikee's World'. Infected files have their time-stamps set to 32 seconds.

```
Nomad.888       B440 8D96 0000 B978 03CD 218B 86F1 03B1 04D3 E08B C88B 86D8
```

Nostardamus

CER: Two stealth, polymorphic appenders with the texts 'XX:YY HOME RUN !', 'Formatting disk X:', 'The NOSTARDAMUS-Erase' and '40Mb'. The 2147-byte variant also contains the text '(CopyLeft) Version 2.7 beta by Populizer' whilst the 2222-byte one has '(c) v2.7 beta by Populizer'. Infected files have their time-stamps set to 20 seconds. The pattern is for memory detection only.

```
Nostardamus.2147    BAE0 07B9 0B35 81F1 683D B472 80F4 32CD DB3B C19C 33C9 33D2
Nostardamus.2222    BA0F 08B9 1800 B44C 80F4 0CCD DB5B 59B0 7A34 7080 E1E0 02C8
```

Pofu.5392

CEMR: A multi-partite, polymorphic, encrypted, 5392-byte virus which infects MBRs on hard drives and executable files (after booting from an infected disk). It contains the texts 'system', 'SYSCMNDR', 'OS=Windows_NT', 'AVP.EXE', 'NAVBOOT.EXE', 'AIDSTEST.EXE', 'COMMAND.COM', 'SCANREG.EXE S-ICE.EXE', 'CHKDSK.EXE', 'NTDETECT.COMCOMSYSEXE', 'WIN.COM' and '*******Bytes*Memory 655 360Dis is PowerFul v3.0 (c)'98 // DarkKiller'. The following template detects the virus in memory only.

```
Pofu.5392       832E 1304 06B1 06CD 12D3 E08E C0BA 8000 B903 00B4 02B0 0CCD
```

Preacher

CR: Two prepending viruses with the texts 'PREACHER\p~~~tmp', '*.com' and '!Jesus Reigns!'. The 549-byte variant also contains the text 'COMMAND.COM'.

```
Preacher.524    BA00 01B9 0C02 902E 030E 9A00 B440 CD21 B801 572E 8B0E 9600
Preacher.549    BA00 01B9 2502 902E 030E 9A00 B440 CD21 B801 572E 8B0E 9600
```

Riverco.2959

ER: A stealth, polymorphic, 2959-byte appender with the texts 'NOTBMEMXXX', 'TBDRIVER CO', '.EXE', 'your PC is now controler for michelangelo' and 'Packed file is corrupt'. Infected files have the word 4C44h ('DL') at offset 0012h. This pattern detects the virus in memory only.

```
Riverco.2959    BE27 00BF E90B B91B 0BAC 2632 06CF 0BF6 D0AA E2F5 6107 1FC3
```

Roman.1995

CR: An encrypted, 1995-byte appender with a long message in Russian containing a few English phrases 'Roman (1) Virus!', '.COM', '(C) -= Pilz =-, 16 years old.' and '.909-436X' and the text 'comCOMcommandCOMMANDdrwebDRWEBroman!.113ROMAN!.113'. Infected files have the word 4252h ('RB') at offset 0003h.

```
Roman.1995      2E8A 04E8 0F00 2E88 0449 4683 F900 75F0 61FB EB1A 9052 5350
```

TinyD.273

CER: An appending, 273-byte virus containing the text 'C_TinyD'. Infected files have the word E94Dh at offset 0000h (COM) and the word 4354h ('TC') at offset 0012h (EXE).

```
TinyD.273       B440 B911 0133 D2CD 2132 C0E8 63FF B440 B918 00BA 1501 CD21
```

Trivial.Dest.49

CN: An overwriting, 49-byte, direct infector with the texts '*.COM' and 'DESTROER' (see Wit.2663).

```
Trivial.Dest.49 BA9E 00B8 023D CD21 93B4 40BA 0001 B131 CD21 B44F EBE6 B44C
```

VCL.470

CEN: An encrypted, overwriting, 470-byte, fast, direct infector containing the texts '*.COM', '*.EXE', '[VCL]' and 'Fuck you Danzig. You are a piece a shit...We know about CUNTS..Love FLT'. The virus uses two different encryption schemes matching the following templates.

```
VCL.470         ??BF 0701 B9E0 0081 35?? ??47 47E2 F8C3
VCL.470         ??BE 0701 B9E0 0081 34?? ??46 46E2 F8C3
```

Wit.2663

CN: An encrypted, 2663-byte direct infector containing the texts 'LOVE', 'chklist.*', 'anti-vir.dat', 'msav.chk', '*.avb', '*.log', 'tbscan.sig', 'smartchk.cps', '*.ms', '[INQUISITOR II] Copyright (c) by Wit 1997.' and a poem in Russian. This virus drops Trivial.Dest.49. It is doubly encrypted and mildly polymorphic, so no simple detection pattern is possible.

Yelet.2105

CER: An encrypted, appending, 2105-byte virus, containing the texts 'YeLeT 0.9, just another bug in your Micro$oft System...', 'c:\windows\win.com' and 'scavtbf-fi'.

```
Yelet.2105      8DB6 0E01 2E8A 042E 3286 0301 3C90 740A 9090 902E FE86 0301
```

Zlodic

CN: Two, simple, overwriting, direct infectors containing the texts 'Zlodic' and '*.cOm'.

```
Zlodic.52       B802 3DBA 9E00 CD21 93B4 40BA 0001 B134 CD21 B43E CD21 B44F
Zlodic.60       B802 3DBA 9E00 CD21 93B4 40BA 0001 B13C CD21 B43E CD21 B44F
```

Zlodic.666B

CE: An encrypted, appending, 666-byte virus containing the texts '*.COM' , '*.EXE', '-*Zlodic.666*-' and 'MIEM=RULEZ'. All infected files have the byte 90h (NOP) at offset 0003h.

```
Zlodic.666B     33F6 B96E 023E 8AA2 3301 32E0 3E88 A233 0146 3BF1 7702 EBED
```

CONFERENCE REPORT

Tram, Bam, Danke Schön

What better place to ponder the developments of the anti-virus industry, not to mention the German obsession with radishes and gravy, than the sumptuous surroundings of the Munich Park Hilton, venue for this year's *Virus Bulletin* conference? Over two hundred and fifty delegates, speakers and exhibitors from all over the world made VB'98 the biggest and, judging by the feedback, the most successful of our conferences to date.

Data Fellows (Finland) sponsored the delegates' badges, *NAI* (GmbH) the conference bags containing full proceedings on paper and CD, and a *Norman* (Norway) the VB'98 tee-shirts. A big thank you to all the sponsors who were instrumental in making VB'98 so conspicuously chic.

Weary German commuters waiting for the seven o'clock tram home on Wednesday evening may have been puzzled to see the cream of the world's anti-virus industry straggling along the twilight Munich streets in a disorganized crocodile. To add insult to injury, VB'98 appropriated the next three trams and set off sightseeing to the sounds of lederhosened 'oom-pah' and clinking glasses. The first was rather eclipsed by the last. Cheers to *Sophos*, sponsor of this welcome drinks reception with a difference.

On Thursday morning we were treated to the full glory of the sun on the Alps at breakfast on the fifteenth floor. The location of the VB'98 dining rooms presented something of a challenge due to the smooth velocity of the elevator which left some of us fighting an odd combination of travel sickness and vertigo, not helped by the previous evening's surfeit of schnapps. Nonetheless, there was a full and enthusiastic turnout for *VB* editor Nick FitzGerald's opening address – 'What a difference a year makes' – a talk which set the tone for the entire conference.

Carey 'Muffin' Nachenberg preparing his paper on heuristic virus detection.

Nick's presentation encompassed a range of issues and perspectives, considering the major events of the last year along with predictions for the future of both development and developers. He highlighted the importance of the industry's awareness that anti-virus is a service not a product, and introduced the idea of a united front of developers and vendors – to be debated up to the last minutes of the speakers' panel on Friday evening.

He went on to cover the mutating corporate face of the industry and new, serious developments in virus creation like CIH. Further topics included the increase in Win32 expertise, 'improved' macro techniques, the net-ification of viruses, 'monster' viruses, and viral 'nastygrams'. He remarked on the emergence of new classes of host – *mIRC* SCRIPT.INI viruses, Access macro viruses, cross platform and Java viruses. Lastly, he touched on non-virus developments – the apparent 'disappearance' of Trojans, new hoaxes like Bloat and network backdoors.

While representatives from all the major international anti-virus companies exhibited, they formed a noticeably smaller crowd than last year, reflecting the effect of recent mergers and acquisitions on vendor numbers. There was unprecedented press presence this year, from publications including *Chip*, *Focus*, and *Secure Computing*. Independent journalists from all over Europe also attended.

From the off, *NAI's* Jimmy Kuo's keynote address 'Add Common Sense, Stir' promised to be challenging and controversial. He discussed several issues that were the focus of later sessions in arguing that an overly-rigorous, 'scientific' approach to product design decisions could be detrimental to the industry's clients.

The papers were presented in two streams – corporate and technical. Previous *VB* editors Ian Whalley and Richard Ford looked at the problems and pitfalls of certification schemes and general anti-virus software reviews. Meanwhile, Péter Ször presented a detailed and well-received paper on Win32-specific virus threats. Rounding off Thursday's morning session, Robert Stroud tracked evolutions in the virus scene and the implications of the changing threat for anti-virus and security policies while Carey Nachenberg discussed heuristic virus detection.

Randy Abrams then described the lengths to which his team go to ensure *Microsoft* does not ship known viruses in retail and digitally-signed software. Unable to influence the design and coding teams to use 'clean room' techniques leaves his Product Release Services group in a similar position to that which the magazine cover CD people face.

Christine Orshesky's paper followed, with a detailed description of defining, testing and choosing anti-virus software, comparing the procedure to that of purchasing a new car. In the technical stream, securing your Web browsers through proper configuration once you understand the threats, and by using 'anti-vandal' software were the themes discussed by John Morar and Dave Chess, and Shimon Gruper, respectively.

After tea, issues of dealing with anti-virus software in large and diverse corporate and university environments were tackled in the corporate stream by Ian Clark and Shawn Campbell, and Dave Phillips, while the technical stream grappled with macro virus issues. Jakub Kaminski investigated 'disappearing macros' due to shortcomings in the compatibility of WordBasic and VBA5 and Vesselin Bontchev warned of the 'evils' of macro virus upconversion if practised by anti-virus researchers.

The gala dinner, generously sponsored by *Network Associates* (USA) was a huge success, with interactive entertainment and a superlative pianist. Highlights included the mime clown's unrehearsed double act with one of the *Sophos* directors and four seemingly consecutive renditions of Barry Manilow's 'Mandy' requested by the less than voguish Assistant Editor [*I'm pleased* she *wrote that. Ed.*]. In a departure from the norm, an extempore dance session saw the hard core element boogie the night away.

NAI's Dmitry Gryaznov takes centre stage at the gala dinner.

A later start with only one session before coffee saw a quite respectable turnout to the first session on Friday morning. Daniel Diefenderfer pointed out some of the 'institutional' barriers to effective anti-virus software maintenance thrown up *within* a company and how to work around some of them. The technical stream heard Marko Helenius present the details of some of the automated virus replication and software testing systems he has developed for use in the University of Tampere anti-virus software evaluations.

After the break Paul Ducklin discussed strategies for catching viruses at their entry point to organizations, and Sarah Gordon and Dave Chess presented their research on 'the truth about Trojans on the Internet'. Meantime, attendees of the corporate stream were treated to Cindy Snow's description of the development processes involved in the evolution of *Intel's LANDesk Virus Protect*. This was followed by Bruce Burrell's exhaustive [*and exhausting, for him! Ed.*] study of the appearance of viruses in the WildList relative to their initial detection.

Postprandial proceedings kicked off with David Aubrey-Jones' consideration of the issues surrounding email encryption and the technical problems this can pose for email virus scanners. He was succeeded by Emily Hawthorn's presentation on the significance of mail gateway scanners. *VB* conference veteran Steve White reflected, for the technical stream, on problems that are still unresolved despite more than a decade of anti-virus research. Then Mikko Hyppönen, newly coiffured, described an approach to macro virus detection that alerts the presence of non-certified macros rather than that of known viral ones.

The final technical session saw Stephen Trilling contemplating the pros and cons of various approaches to incremental product updates. Next door in the corporate stream, Shane Coursen set the stage for a lively speakers' panel session during his paper on WildList developments with the suggestion that the *WildList Organization* (*WLO*) should provide reviewers with verified virus samples.

The speakers' session at the close of the conference proved popular and controversial – Nick represented *VB*, Carey Nachenberg *Symantec*, Steve White *IBM*, Dmitry Gryaznov *NAI*, and Paul Ducklin *Sophos*. Jan Hruska presided over a very enthusiastic full house. Shane Coursen, Ian Whalley and Richard Ford met Vesselin Bontchev head on in the debate over the *WLO* providing samples to reviewers.

Paul Ducklin reiterated the value of presenting a solid team of developers and vendors, despite differences in marketing and development techniques and priorities. The audience appeared to be surprised at the level of technical cooperation between anti-virus companies, given the extent of inter-marketroid bickering. It seemed to be a staple theme of this year's conference that the 'you' and 'us' developer/vendor dichotomy be screened by a manifestation of unity from the 'same side'.

On behalf of the whole team, big thank-yous go to several people. Petra Duffield manages to outdo herself every year, and VB'98 went especially smoothly thanks to her organizational expertise. A big thank you Pet from everyone in your corner. Thanks are also due to Dan 'Roger Irrelevant' Trotman, making his memorable début as a conference helper along with the new conference coordinator and subscriptions manager Jo Peck.

Kim and Müsli, so cheerful, professional and polished every year, were joined by newcomer Sarah – all three must be congratulated for their invaluable and efficient help. Thanks also to 'Big Rich' for driving all the conference material over and helping to set it up. The conference organizers would like to voice their appreciation for the boys from Gearhouse, who were responsible for the audio-visual equipment and presentations. They managed another superb job and their group rendition of 'Yellow Submarine' at the gala dinner will become the stuff of future legend.

Last but not least, thank you to all the staff at the Munich Park Hilton, especially Andreas and Martin who managed to maintain their composure and patience despite the logistics of getting two hundred and fifty delegates up fifteen floors twice a day.

Special mentions to – *ICSA's* Scott Markle and his remarkable talent for spotting biscuits at a thousand paces. Marta Olafsdottir for her professionally rendered Lloyd Webber numbers in the piano bar, Svein Meland and his wife for being dazzling in authentic Norwegian costume, and last but not least, to Stephen Trilling for his very generous offering of a million *Symantec* dollars to shave the editor's beard off for charity. [*No joy I'm afraid. Ed.*]

VIRUS ANALYSIS

The Marburg Situation

Péter Ször
Data Fellows

While the number of 32-bit *Windows* viruses is not rising quickly, it is alarming that three have reached the WildList recently. We first saw Win95/Anxiety (see *VB*, January 1998, p.7), then different variants of Win95/CIH (*VB*, August 1998, p.8) and now Win95/Marburg is on the list. The latter is the first in the wild, polymorphic virus to infect only Portable Executable applications. Few *Windows* viruses have been this successful – most have been full of bugs and thus unlikely to become widespread.

Where Will Viruses Want to Go Tomorrow?

The appearance of Marburg shows that *Windows 9x* viruses do have the potential to spread as quickly as, or even faster than, boot viruses. [*Although this analysis appears after that of CIH, Marburg predates CIH by several months. Ed.*] DOS is not the PC-dominating platform it was, having been replaced with different implementations of *Windows*. When the dominant platform on popular computers changes, some virus types will die out but this will not mean an end to the virus problem. When a new platform begins to dominate, we see thousands of viruses for it in just a few years. This is likely to continue on Win32 platforms – first on *Windows 9x*, later on *Windows NT*.

A year ago we did not see more than one *Windows 95* virus per month. By the middle of 1998 this had changed to an average of about one per week. Nowadays, there is more than one variant per week. During 1999 we may see a new *Windows 9x* or *NT* virus almost every day. So far this is what we have seen with other virus types and nothing suggests it should be different with 32-bit *Windows* viruses. This is not helped by some virus writers realizing just how easy it is to develop viruses in a high level language such as C or even Delphi. [*…and VBA, of course! Ed.*]

There are now several viruses that are around 100 KB to almost half a megabyte long. This is not the case with Marburg, however. Marburg is written in assembler and was probably the first polymorphic *Windows 9x* virus. Its author also wrote Win95/HPS (*VB*, June 1998, p.13). The Marburg polymorphic engine is very similar to that of HPS, but looks like an earlier development. While HPS hooks system functions, Marburg is a direct action infector. The balance of probabilities suggests Marburg must have been written and released long before HPS for it to be in the wild. HPS has not been seen in the field so far.

Marburg's current claim to fame is that files infected with it were found on several magazine cover CDs. First, it was included accidentally on the cover CD of the July 1998 edition of UK-based *PC Gamer* magazine. A utility program that was automatically executed if you chose to watch any of the preview videos from the CD was infected. Localized versions of *PC Gamer* exist, in addition to the UK edition. The Swedish and Slovenian editions also carried infected files.

Then, in August, Marburg was included on the master CD of the popular *MGM/EA* game *WarGames*. It also had widespread circulation on the cover CD of Australian *PC PowerPlay* magazine in August 1998. [*A further incident is listed in the October issue Editorial. Ed.*]

This seems to be the most 'successful' distribution of a virus, at least in such a short period of time. Unlike many viruses, Win95/Marburg has few bugs but, fortunately, one of them is fatal enough (if very small) to prevent the virus replicating under *Windows NT*. The virus contains the text '[Marburg ViRuS BioCoded by GriYo/29A]', hence the name Win95/Marburg.A. The .B variant is also unable to replicate under *Windows NT*.

Why did Marburg get the opportunity to spread so widely and end up on so many commercial CD-ROMs? As it happens, most scanning engines had to be changed in order to detect this virus reliably. Such major changes always require a longer development time than simple database updates. There are still only a few products which can detect Win95/Marburg.A in all circumstances.

The virus utilizes a slow polymorphic replication mechanism. Further, the infection method differs slightly in some files. This small difference may not have been apparent to some virus analysts at first glance. A few missed samples on each PC can be enough to keep the virus circulating over and over. Deliberately targeting screen saver (SCR) files may also have assisted distribution.

Executing an Infected Application

Win95/Marburg.A is a PE infector. When an infected 32-bit application is executed, the virus code takes control. When the host program does not have relocation for the first five bytes at its entry point, the virus places a jump instruction there and does not modify the entry point field in the PE header. Otherwise, if it is really necessary because there is a relocation for the first instruction, it modifies the entry point in the PE header and the code at the original entry point remains the same.

Then comes the trickiest case. When there are no relocations for the first 255 bytes from the entry point, the virus not only places a jump instruction in the code at the entry point of the host, but builds a random garbage code block first and puts the jump to the virus' polymorphic decryptor at the end of it. The size of the junk block

together with the jump will be less than 255 bytes. The jump instruction or the entry point field of the PE header points to the very end of the real virus body which is always attached to the last section of the host program. Then the polymorphic decryptor decrypts the virus body which precedes it.

The size of the virus body (without the decryption code) is a constant 5793 bytes but infected files will grow by around 7900 bytes. This is because the size of the polymorphic decryptor and the constant virus body is 7841 bytes and the virus pads itself out to make the infected file size exactly divisible by 101. Several viruses written by members of the 29A group use the same self-recognition technique to prevent multiple infections.

Marburg uses several techniques with similar functionality to that of Win32/Cabanas (see *VB*, November 1997, p.10). Marburg attempts to save pointers to the import addresses of the GetModuleHandleA and GetProcAddress APIs during the infection process. Once the virus body is decrypted and control passes to it, if these API addresses are available in the host program's import table, Marburg can work easily.

When the address of GetModuleHandleA is not available, as in Cabanas, Marburg tries to use the ForwarderChain field of the import table. At that moment the virus knows the base address of the loaded KERNEL32.DLL in the virtual address space of the process. When the host program does not have imports for the GetProcAddress API, the virus simply searches the export table of KERNEL32.DLL and picks up the address from there.

After that, the virus is in a position to obtain all the API addresses it needs from KERNEL32.DLL. Altogether there are nineteen APIs of interest to the virus (including CreateFileA, CreateFileMappingA and GetSystemTime) and it gets all of them by calling the GetProcAddress API in a loop. If an error should occur, the virus executes the host program. Before executing the host, Marburg checks whether fixes are needed at its entry point. If so, the virus replaces the code at the host's entry point with the original code, then passes control there.

If no errors occur, Marburg allocates memory and copies itself there, passing control to that copy of itself. This mechanism is needed because of the virus' polymorphic engine. During infection, Marburg saved the current date and hour in its body. At this point of execution it checks whether the infected program is being run three months after its infection and during the same hour. Whenever these conditions are met, Marburg calls its payload.

Spots Before Your Eyes?

The payload routine needs the addresses of three APIs from USER32.DLL. The virus first ensures this library is loaded, then uses GetProcAddress three times in succession to call the required APIs.

Marburg's payload is not damaging. This mess can be cleared simply by maximizing then restoring an application window.

The first API is LoadIcon. Marburg loads the standard IDI_HAND (0x7F01) icon resource which *Windows* uses in the case of serious error messages – a white cross on a red circle in the case of *Windows 9x*. Then it gets a handle to the desktop with the GetDC API. Finally, it draws up to 255 icons (depending on the screen resolution) at random positions on the desktop.

Windows 9x will gradually redraw the desktop area as window sizes change, causing Marburg's icons to disappear. However, the same infected program will display some icons again in the same hour, as will other infected applications, should the payload trigger conditions be met.

Infection

Marburg is a direct action virus. The virus tries to infect one file with an EXE or SCR extension in each of the current, *Windows* and *Windows* System directories. Marburg is a retro virus, deleting known checksum files of different anti-virus products such as 'ANTI-VIR.DAT', 'CHKLIST.MS', 'AVP.CRC' and 'IVB.NTZ' in every directory it attempts to infect. The virus avoids infecting many anti-virus programs. It also avoids infecting any program whose name contains the letter 'V' or strings 'PAND', 'F-PR' or 'SCAN'.

Before infecting a file, the virus changes the file attribute to normal. Thus, it can easily infect read-only files. The virus uses file mapping functionality which makes the infection process much shorter than it would be otherwise. Marburg always places itself into the last section of the host. However, it does not infect the file if the last section has shared characteristics. It sets the last section characteristic to include the writeable attribute.

The infection procedure is protected with Structured Exception Handling (SEH), thus the virus will execute the host program if a GP fault should occur in its own code. Viruses using this technique can be very stable (and more successful). During infection, the virus checks if the host

program is 386-compatible and only infects it if that is the case. It tries to save references to the GetModuleHandleA and GetProcAddress import addresses, as this makes the initialization less complicated later on. Then it checks the relocations and according to those uses different entry point strategies as described above.

Before infecting a file, the virus calls its polymorphic decryptor generator. This engine implements slow polymorphism, thus several infected files on any infected PC will have the same polymorphic decryptor. Further, the number of different combinations is limited compared to what the engine could generate.

In any case, the virus' polymorphic engine is powerful, using several different encryption methods and keys. The size of the polymorphic generator is 1872 bytes and, as already noted, is similar to that of Win95/HPS, but somewhat limited in comparison. When the polymorphic decryptor is generated, the virus encrypts its main body and writes everything into the host. At the end of each infection, the virus changes the host's file attributes back to their original state. Finally it executes the host program.

Conclusion

The wide distribution of Win95/Marburg shows that a well written, complex, polymorphic virus can be successful. Such complex viruses have not appeared in the wild before, but that situation may be about to change dramatically.

Anti-virus researchers have to invest lot of energy and time into a complete analysis of such new viruses in order to design correct detection and disinfection methods for them. DOS viruses with direct action infection mechanisms do not usually spread too far. This situation seems to be different in case of multi-tasking environments and that makes the job of virus writers even easier.

Marburg

Aliases:	None known.
Type:	*Windows 9x* direct action, PE infector targeting EXE and SCR files.
Self-recognition in Files:	Files whose size can be divided by 101 without remainder are assumed to be already infected.
Hex Pattern in PE files:	Not possible.
Payload:	Displays the default error icon at random positions on the desktop during the hour of initial infection, three months after that date.
Removal:	Recover infected files from backup or replace with original.

FEATURE

The Biggie

Peter Morley
Network Associates

Back in 1992, when virus authoring packages first appeared, people who worked in virus labs became aware of a potential nightmare. What if someone used such a package to produce many thousands of new viruses? How would we cope? Anti-virus people were wary of discussing it in print.

It took a while to happen, but during the last week of September this year, Dmitry Gryaznov told me that it had. From the virus writer's point of view, it had got to be a case of checking the stable door well after the horse had bolted over the hill!

On Friday 2 October, we received the largest collection of viruses which has ever come our way. It consisted of almost 15,000 viruses that we had never seen before. It had been generated in the field and sent to several anti-virus developers and testers.

We completed our processing of this collection within a week. This article explains how we did it, and some of the thoughts which went into deciding the process, given that in our case, the result had to be not only detection of the 15,000 files, but also detection and repair of anything which is produced from them.

First Thoughts

How about:

- Ignore it, and hope it will go away.

  Well, I couldn't get away with this, even if some of our smaller competitors could.

- Issue a Press Release, claiming we already detect most of them, and do no work at all. Just continue processing macro viruses. Anyway, they are not in the wild so they will never be a problem, will they?

  I think you already know my opinion of anti-virus people who hide behind an imaginary wild list!

- How about the classic approach? Replicate each one which will replicate, and process it normally. We can do all the replication without manual intervention. But what do we do then? We will have a lot of files!

  We do not have an *IBM* automatic processing system, and even if we did, might this possibly clog it up?

Second Thoughts

Hang on a minute. We would have to do the unavoidable. Examine the problem, and *think*. The first three stages are standard, starting with the elimination of all duplicate files.

We did so, but there were not that many. Secondly, we ran *Findvirus*, all switches on, and made a listing. Thirdly, we had to examine the listing (no matter how big!).

The Collection

A lot of work had been done to produce it. There were 22 subdirectories, named A through V. The first, and largest, contained 1470 files, but none of the others contained more than 1000. There were 14,843 files in all, of which 891 were not detected at all. Those which were detected had mainly been produced using generator packages, like PS-MPC and IVP, and were droppers, rather than infected files. There were a few oddments.

Most of those detected were MPC1, MPC2, MPC3, MPC4, MPCa and IVP, so were already covered by six of our largest existing drivers.

I classified the elements into five groups, which I intended to process in this order:

- The oddments.
- The checksum coincidences.
- Files already detected and repaired using existing generic repair techniques.
- Files already detected as 'is like', and not repaired.
- The 891 which were only detected with heuristics.

The Oddments

There was a Jerusalem.Pipi in the collection. I double-checked. It definitely was. We repaired it successfully. The question is – how did it get there? Or, at least, why?

There were several Bacteriological Warfares. I checked, and we repaired the ones we identified. I processed two new variants we had never seen before.

The rest was business as usual.

The Checksum Coincidences

The term 'checksum coincidence' requires some explanation. In order to identify a virus, we calculate a checksum of some of its constant areas. When we calculate a checksum there is a one in 65,535 chance it will wrongly match a file which is later added to the same driver. In the normal course of events, we see these coincidences about once every two years. When it does happen, it is serious, because it can cause misrepair, as a result of misidentification.

When you add 3000 new viruses to one of our drivers which already has 100 checksums in place, the probability of such a coincidence is multiplied by 300,000. So we would expect four or five coincidences, on such a driver.

Fortunately, dealing with these checksum clashes is fairly straightforward. We reduce the area being checksummed on the old virus causing the problem, and recalculate the checksum. In this instance, there were slightly fewer of these coincidences than we had expected, and I managed to change them all.

Generic Repairs

This third point also needs explanation. We have been using generic repair techniques for some time, so that some new viruses are detected and repaired on customer sites, without us ever seeing them. It happens when we have already had several variants, and have catered for future similar ones. We have employed these techniques in all six of the big drivers covering this collection.

For those terribly concerned about numbers, the downside is that some new viruses often do not get counted, as we never see them.

There was no work to do, to process this category. You can quote me if you ever get involved in a discussion with anyone who does not approve of generic detection and repair of viruses!

Viruses Detected as 'is like' and Not Repaired

This was where the real donkey work had to be done. I had to modify each of the six big drivers to introduce new generic repair for the samples in this category. This was followed by the problem of testing that it all worked.

I followed the classic testing strategy – select samples at random and carry on replicating and testing, until you get fed up because they all work. If anything happens to cause a change, fix it and restart the testing. In this case, thank goodness, they all worked. These new generic repairs are reported as '.GR5'.

891 Only Detected with Heuristics

I expected this category might take several weeks to deal with. However, four of them were generators, which had obviously been used to prepare the collection. I decided to classify them as Trojans, and detect them.

One file was really strange. It was a DOS 3.0 file, multiply-infected by Ionkin and Homecoming, then mis-infected by Homecoming. I decided to detect it as a curio, and again, puzzled over how it got in there and why…

That left 886 'new viruses', and how lucky can you get? They were all variants of a single new morph I had never seen before. I was able to write a new driver (MPC7) to detect and repair the lot. Testing was as described above, and it worked without going back and rehacking. These generic repairs are also reported as '.GR5'.

Summary

By the time you read this, the 'problem' can be forgotten, because *Network Associates* will have released a version of *VirusScan 4*, which will detect and repair them all!

COMPARATIVE REVIEW

Opening Windows 98

Once more into the jungle that is today's anti-virus world, for a spot of behavioural observation. Here, however, extinctions occur with rather more rapaciousness than the Dodo's demise upon Mauritius.

Of the products reviewed this month yet another, the *Intel* species, was declared extinct during testing, swallowed by a large *Symantec*, while *Dr Solomon's Anti-Virus Toolkit* is destined to undergo significant evolutionary changes. However, preambles of this kind will only serve to keep the eager reader from the real purpose of the review and so the introduction ends here.

Test Procedures

The platform used for these tests was *Windows 98*, the same setup as that in the review of *Sophos Anti-Virus* last month. FAT32 disks were not used, because the sizes of the partitions employed for testing were too small. There are plans to alter this in future reviews.

The same machine was used for all the timing tests, while two other hardware-identical machines were used in conjunction for the on-demand and on-access scanning processes. In all cases the software was deployed in its standard configuration, unless this removed such useful features as on-access scanning or the ability to alter configuration of the scanners.

The August WildList was used in conjunction with the ever expanding Macro, Polymorphic and Standard test-sets, against products dated 1 September at the latest. Where possible, scan tests were run from a CD, thus removing the need to restore files after each scan as a precautionary measure against overkeen deletion or disinfection. Several products, however, produced useless report files or none at all. In these cases deletion or quarantining was used in order to obtain meaningful results.

On-access scanning overheads were tested using XCOPY to move large numbers of executables, the results being compared against a baseline with that component inactive. Floppy disk speed tests were performed upon two almost identical disks, differing only in that the files on one were all infected with Natas.4744. The hard disk scanning test, combining speed with false positives on 5500 executables, is the standard *VB* test, and comparable with results in the last *NT* comparative in September.

The complete detection tests are reported in the main tables. The results reported in the summaries are only the on-demand variety, plus the on-access result for the combined In the Wild test-sets and the Macro test-set.

Alwil AVAST32 v7.70 (Build 725)

ItW Boot	100.0%	Macro	98.2%
ItW File	100.0%	Macro o/a	n/t
ItW Overall	100.0%	Polymorphic	98.3%
ItW Overall o/a	n/a	Standard	99.7%

Still emblazoned with a horde of beetles, *Avast32* continues to sit with the better class of on-demand detectors, but remains untestable by *VB's* on-access scanning methodology.

This is not the problem it might seem – the on-access detection of viruses is dependent on an attempt to execute, which makes the testing of this function a task too epic to undertake in one lifetime. Nevertheless, *Alwil's* product remains reliable and stable, giving little cause for anything but pleasant comment.

CA Cheyenne Inoculan AntiVirus v5.0.4.13

ItW Boot	100.0%	Macro	98.2%
ItW File	100.0%	Macro o/a	98.2%
ItW Overall	100.0%	Polymorphic	99.1%
ItW Overall o/a	99.6%	Standard	100.0%

As ever *Inoculan* was frustrating to the degree that it endangered the reviewer's mortal soul as he invented new curses to lay upon *Cheyenne* programmers. The log file problem remained the greatest single obstacle – by all appearances, the program creates log files in memory which causes it to become ever more hungry for resources as scans of large numbers of viruses progress.

The act of attempting to print the log to file is enough to crash *Inoculan*. On-access scanning, meanwhile, is beset by a similar problem of resource leakage, which resulted in frequent hangs and the speed of the machine degenerating to that of an arthritic sloth.

With all of this laggardly behaviour *Inoculan* also manages to throw in a streak of capricious disobedience too. No amount of changing instructions could provide a setting where the on-demand boot infector tests did not produce a choice of actions to take. Such obvious settings as 'log only' had some mystical significance quite at odds with their literal meanings. There was also an impressive ability for the program to report a virus in memory when scanning of boot disks had just occurred – only likely to be true if *Inoculan* has masochistic code designed to activate boot viruses if detected.

Nevertheless, *Inoculan* was able to detect well in all categories which were testable – on-access polymorphic testing could not be completed without inducing catatonia

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Alwil Avast32	84	100.0%	738	100.0%	100.0%	1685	98.2%	14385	98.3%	1017	99.7%
CA Cheyenne Inoculan	84	100.0%	738	100.0%	100.0%	1684	98.2%	14433	99.1%	1026	100.0%
Command AntiVirus	84	100.0%	726	99.6%	99.6%	1715	99.5%	14176	97.3%	1017	99.7%
Cybec Vet NetSurfer 98	84	100.0%	726	99.6%	99.6%	1686	98.1%	14086	96.6%	1008	98.9%
Data Fellows FSAV	84	100.0%	738	100.0%	100.0%	1700	99.1%	14415	99.8%	1017	99.7%
DialogueScience Dr Web	0	0.0%	738	100.0%	89.8%	1683	98.1%	14394	99.7%	1017	99.7%
eSafe Protect	83	98.8%	708	98.2%	98.3%	1518	90.4%	13456	91.5%	1007	99.1%
ESET NOD32	84	100.0%	738	100.0%	100.0%	1711	99.1%	14381	99.5%	1026	100.0%
GeCAD RAV	82	97.6%	738	100.0%	99.8%	1706	99.4%	13865	95.4%	980	95.7%
Grisoft AVG	83	98.8%	686	94.8%	95.2%	1337	79.5%	12796	88.5%	883	87.0%
H+BEDV AntiVir	82	97.6%	659	95.5%	95.7%	1545	92.3%	11558	79.1%	980	96.9%
Intel LANDesk Virus Protect	81	96.4%	716	99.2%	98.9%	1578	94.0%	13611	94.0%	1013	99.5%
iRiS AntiVirus	84	100.0%	738	100.0%	100.0%	1688	98.4%	14433	99.1%	1026	100.0%
Kaspersky Lab AVP	84	100.0%	738	100.0%	100.0%	1700	99.1%	14415	99.8%	1026	100.0%
NAI Dr Solomon AVTK	84	100.0%	738	100.0%	100.0%	1692	98.6%	14287	97.6%	1026	100.0%
Norman TBAV	84	100.0%	730	99.7%	99.7%	1607	95.4%	14083	94.8%	997	98.2%
Norman Virus Control	84	100.0%	738	100.0%	100.0%	1617	96.0%	14294	99.0%	1017	99.7%
Sophos Anti-Virus	84	100.0%	738	100.0%	100.0%	1640	97.2%	14273	98.8%	1015	99.5%
Stiller Integrity Master	82	97.6%	559	86.1%	87.3%	1050	63.7%	5081	30.7%	769	81.9%
Symantec Norton AntiVirus	84	100.0%	738	100.0%	100.0%	1719	99.8%	14443	98.7%	1017	99.7%

upon the test machine. This detection rate is the only saving grace for *Inoculan* and the only part of the program which is not produced by *CA* programmers.

Command AntiVirus for Windows 95 v4.52

ItW Boot	100.0%	Macro	99.5%
ItW File	99.6%	Macro o/a	99.5%
ItW Overall	99.6%	Polymorphic	97.3%
ItW Overall o/a	99.6%	Standard	99.7%

The monitor lizard is a particularly close relative to *Command AntiVirus* (*CSAV*), both being slow lumbering creatures yet very effective in their respective hunting niches. No false positives were recorded during the scan of the Clean test-set – a 'suspicious' warning was the limit.

This conclusion was reached at a lethargic rate – only two products were slower. On-access overheads were of a more strolling nature, slowing affairs by a factor of four or more. Floppy disk speeds alone were an area where *CSAV* approached the median in terms of velocity.

On-demand tests resulted in good levels of detection – against the ItW test-set only Marburg and TVPO.3783.A were missed, the former being a worrisome creature given its current wide domain. It was also the sole virus missed in the Polymorphic test-set. Macro misses were due to AccessiV.A and B, which are not scanned in the default setting due to the large extra overhead incurred by default scanning of MDB files. Against the Standard test-set, Navrhar falls into the same category of unscanned files but here VxDs are ignored by default – a precaution particularly necessary in this less than swift scanner.

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Alwil Avast32	84	100.0%		n/t	n/a		n/t		n/t		n/t
CA Cheyenne Inoculan	81	96.4%	738	100.0%	99.6%	1684	98.2%		n/t	1026	100.0%
Command AntiVirus	84	100.0%	726	99.6%	99.6%	1715	99.5%	14170	96.4%	1017	99.7%
Cybec Vet NetSurfer 98	83	98.8%	738	100.0%	99.9%	1691	98.2%	14340	98.0%	1008	98.9%
Data Fellows FSAV	84	100.0%	738	100.0%	100.0%	1701	99.1%	14444	100.0%	1026	100.0%
eSafe Protect		n/a	703	97.4%	n/a	1511	90.0%	13456	91.5%	1026	100.0%
ESET NOD32	84	100.0%	738	100.0%	100.0%	1711	99.1%	14381	99.5%	1026	100.0%
Grisoft AVG	49	58.3%	416	61.6%	61.2%	1140	68.8%	1102	7.5%	614	68.1%
H+BEDV AntiVir	24	28.6%	685	96.6%	89.7%	1548	92.5%	12178	84.1%	994	98.0%
Intel LANDesk Virus Protect	78	92.9%	366	56.4%	60.1%	180	9.9%	515	3.5%	608	68.4%
iRiS AntiVirus	81	96.4%	738	100.0%	99.6%	1688	98.4%	14419	95.5%	1026	100.0%
Kaspersky Lab AVP	84	100.0%	738	100.0%	100.0%	1700	99.1%	14415	99.8%	1026	100.0%
NAI Dr Solomon AVTK	83	98.8%	738	100.0%	99.9%	1688	98.4%	14287	97.6%	1024	99.7%
Norman TBAV	60	71.4%	657	89.3%	87.5%	1242	73.9%	14444	100.0%	1008	99.0%
Norman Virus Control	82	97.6%		n/t	n/a	1628	96.2%		n/t		n/t
Sophos Anti-Virus	84	100.0%	738	100.0%	100.0%	1636	96.9%	14273	98.8%	1015	99.5%
Symantec Norton AntiVirus	84	100.0%	714	97.7%	98.0%	1646	97.7%	13500	93.5%	1017	99.7%

On-access scanning was much the same, though a handful of Cryptor samples evaded the snapping jaws of *CAV* in addition to those already noted. The status of the on-access scanner was rather difficult to ascertain at first – what appeared to be a tray icon for on-access scanning was in fact connected with the management console.

Cybec Vet NetSurfer 98 v9.8.5.0

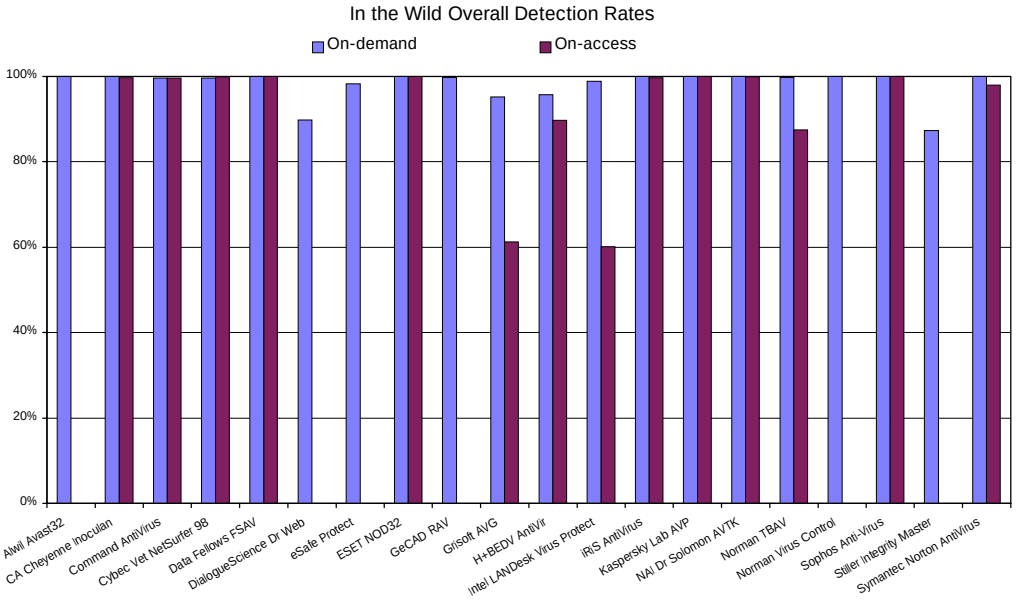
ItW Boot	100.0%	Macro	98.1%
ItW File	99.6%	Macro o/a	98.2%
ItW Overall	99.6%	Polymorphic	96.6%
ItW Overall o/a	99.9%	Standard	98.9%

Vet remains devoted to aardvarks in its manual and were an aardvark's tongue a little swifter in motion there might be some useful comparison to be made. Combining the buzzwords 'net', 'surfer' and '98' it might be expected that this product will appeal to the more gullible of middle management, who would on this occasion at least have purchased a reasonably effective and stable product. *Vet* also remains the speediest of the products reviewed here, with low overheads from its on-access component as well as impressive throughput in both the hard disk and diskette speed tests.

The main disappointment will therefore be the lack of detection of all In the Wild viruses, especially because there is a simple method of overcoming this failing. In the on-access tests *Vet* achieved a full detection rate, on-demand it missed only the screen savers infected with Marburg; the problem clearly being a simple omission from the default scanned extensions (SCR) or fixed programmatically with automatic file type detection. Users of *Vet* would be well advised to add SCR to the list of scanned files – especially if Marburg has been detected elsewhere or is making unexplained returns after disinfection.

Data Fellows F-Secure Anti-Virus v4.02

ItW Boot	100.0%	Macro	99.1%
ItW File	100.0%	Macro o/a	99.1%
ItW Overall	100.0%	Polymorphic	99.8%
ItW Overall o/a	100.0%	Standard	99.7%



with the Clean set. Coaxed through several partial runs, it produced two false positives, but could not be made to scan all of the test-set.

Elsewhere, however, results were good and there was a noticeable speed increase when scanning files on both floppy and hard disks in comparison with the *NT* testing. Detection, too, reached admirable levels, with all file categories recording detection percentages in the high nineties – in the wild files topping this at full detection. All in all, the results can be considered to represent a two-headed calf and act as an extreme example of the perils facing companies when they submit a new, superficially improved, but not quite fully tested, product for review.

eSafe Protect v2.0

ItW Boot	98.8%	Macro	90.4%
ItW File	98.2%	Macro o/a	90.0%
ItW Overall	98.3%	Polymorphic	91.5%
ItW Overall o/a	n/a	Standard	99.1%

The trickiest part of dealing with this product is its serpentine user interface. Once mastered, detection is respectable, though poor against the Macro set and especially on-access. During the overhead tests the inbuilt heuristics were sufficiently oversensitive to trigger upon the execution of XCOPY32. The overhead ratings thus do not include this particular part of the standard protection regime.

It is unlikely that any user would opt for virus protection which prevented any file copies due to their suspicious nature, and considered, as *eSafe Protect* did, that COMMAND.COM should be prevented from executing. The controls for the scanning methods to be used on-access and on-demand are praise-worthily comprehensive, allowing this niggle to be disabled simply.

ESET NOD32 v1.09

ItW Boot	100.0%	Macro	99.1%
ItW File	100.0%	Macro o/a	99.1%
ItW Overall	100.0%	Polymorphic	99.5%
ItW Overall o/a	100.0%	Standard	100.0%

Sadly the pulsating alien heart motif has departed *NOD32* but the rest of the program continues to please. Detection remains at an impressive level, with the sole problem area

Past reviews of the 4.*x* version of *FSAV* have shown it to be fearsomely painful to reviewers due to its instability and an initial problem when faced with on-access boot viruses did nothing to inspire confidence. On this occasion scanning halted after the first sample, giving an apparent detection rate of one. This turned out, happily, to be akin to a bee sting attack – once and once only – the program behaving impeccably thereafter and gaining a detection rate of one hundred percent for both boot sector tests.

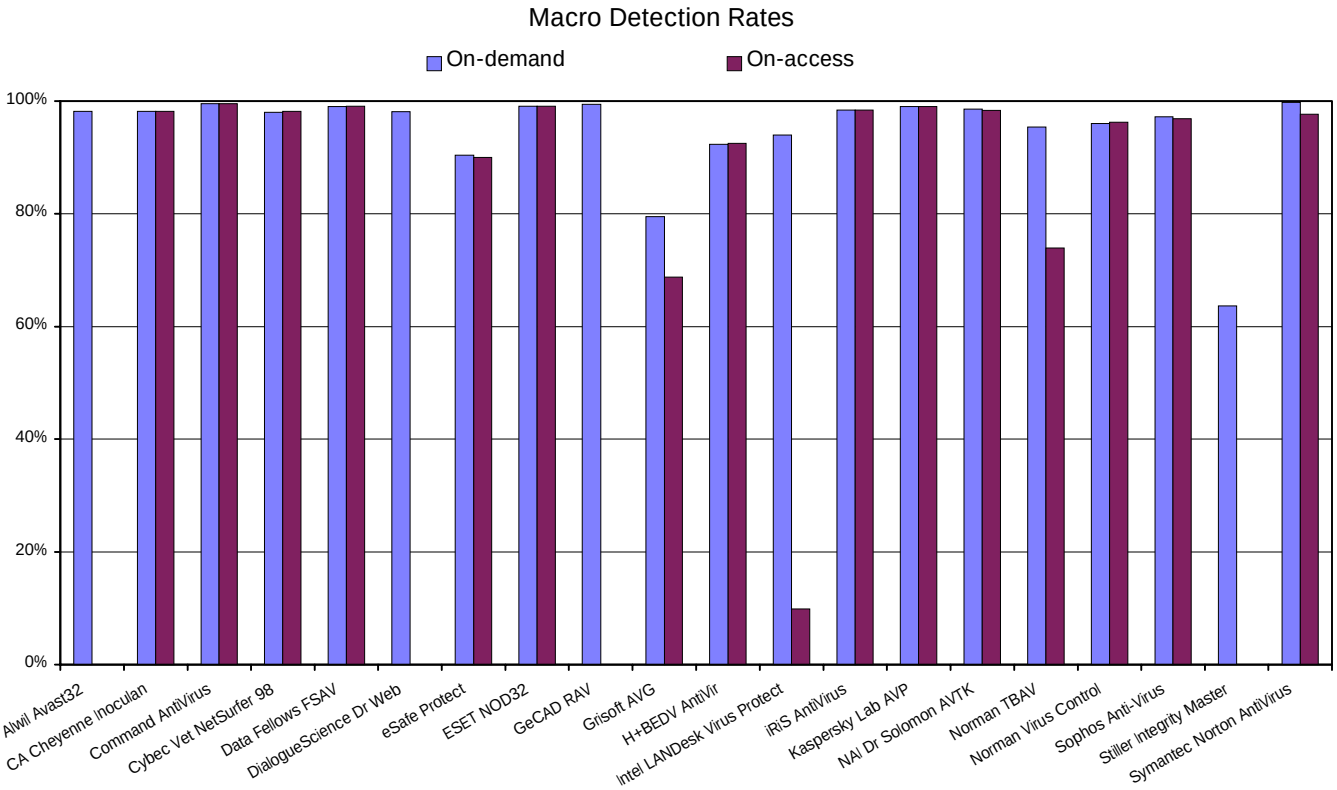
Detection in other areas was admirable too – MDB and VxD files undetected for reasons of speed, and macro viruses, including the almost universally problematical XM/Compat.A, provided the remainder of the misses. It was a notable feature of this test that macro viruses were by and large the greatest bane of the scanners involved, due, perhaps, to the problems involved in dealing effectively with the new generation of polymorphic macro viruses.

DialogueScience Dr Web for Win32 v4.02b

ItW Boot	0.0%	Macro	98.1%
ItW File	100.0%	Macro o/a	n/a
ItW Overall	89.8%	Polymorphic	99.7%
ItW Overall o/a	n/a	Standard	99.7%

In the *NT* comparative two months ago *Dr Web* proved a worthy, though rather slow, program. This slightly different version has no cosmetic changes but something under the skin has been drastically altered, and not all for the better.

The program supplied was admittedly a beta version and the suspicion must be that any release version could not be as flawed as this particular edition proved to be. Most disturbingly, detection of boot viruses dropped from near perfect to none whatsoever, a result which smacks of a botched build. The program repeatedly crashed when faced



being the treatment of the W97M/Splash.A virus. Splash is polymorphic by dint of adding random comments to itself, increasing in size with every generation. Here detection of samples in the lower range of size was perfect, but larger documents remained unflagged as infected. Whether this is a problem which will easily be rectified remains to be seen.

Under *Windows 98* it was also apparent that the 'odd boot sector' viruses had changed compared to those in previous reviews. On-demand *NOD32* declared that the directory path was not valid for ExeBug.Hooker, Michelangelo.A and Quox.A, though the viruses were detected both on-access and on-demand. In products which could not handle such oddities these three proved a particular problem.

GeCAD RAV v6.08

ItW Boot	97.6%	Macro	99.4%
ItW File	100.0%	Macro o/a	n/a
ItW Overall	99.8%	Polymorphic	95.4%
ItW Overall o/a	n/a	Standard	95.7%

Looking distinctly less attractive than its competitors, and without an on-access component, *RAV* is also relatively tortoise-like. This is particularly true of the boot sector tests, where the same age-old system of labyrinthine clicks and keypresses is required for each disk scanned. Seven false positives, similar to those noted in the *NT* test, were also thrown up against the Clean test-set.

All this said, *RAV* remains effective in the prime area of concern – that of detection. Though missing more than it should, *RAV* firmly occupies that middle ground.

Grisoft AVG v5.0 (database 20)

ItW Boot	98.8%	Macro	79.5%
ItW File	94.8%	Macro o/a	68.8%
ItW Overall	95.2%	Polymorphic	88.5%
ItW Overall o/a	61.2%	Standard	87.0%

AVG showed a variety of problems coupled with a readme file containing less than inspiring revelations. On-demand testing comprised a number of options, making the choice of scan a not entirely intuitive one.

The Complete test was chosen, and the on-demand tests performed fairly smoothly, though with a distinctly uninspiring set of detection statistics. Macro viruses proved the greatest challenge to detection, a sign of *AVG's* team being behind the times in their addition of new viruses. To this was added the scenario, drifting in the wake of the scan procedure, whereby *Explorer* refused to perform changes in the current directory – not a pleasing side-effect. Boot sector testing was almost perfect, though the inability to spot Natas.4744, an elder statesman of the virus world, must be considered disturbing.

If on-demand tests were unsatisfactory, on-access ones posed fewer problems for the operating system, but were more disappointing in terms of detection. A host of boot sector viruses passed unnoticed, those which were detected once were missed next time during a sequence of scans, a problem more common under *NT*. Detection of file viruses was similarly poor. Over 14,000 of the 17,000 samples were missed, making for a level of protection which might be considered worse than useless.

With such a low detection capability it is perhaps to be expected that no false positives were encountered, and that scanning proceeded speedily. It was surprising the readme file referred to the new addition of Laroux disinfection, but not that directories with long filenames were still unsupported in the scanning exclusion list.

H+BEDV AntiVir v5.14.0.7

ItW Boot	97.6%	Macro	92.3%
ItW File	95.5%	Macro o/a	92.5%
ItW Overall	95.7%	Polymorphic	79.1%
ItW Overall o/a	89.7%	Standard	96.9%

Whimsically fronted by a turn-of-the-century bathing photograph, *AntiVir* continues to be educational inasmuch as learning German benefits the review process. The on-access scanner was a new addition to this product in *VB* reviews, though these changes were not without concomitant changes in program stability. These manifested themselves in fatal exceptions during both on-demand boot and on-access file tests, and warped GUI antics at other times. *AntiVir* also takes the rabbit prize for timidity, finding 62 false positives in the Clean test-set.

The on-access scanner was all but useless on the boot sector tests, discovering fewer than a quarter of the virus-infected diskettes thrown at it as worthy of concern. Strangely enough, on-access scanning of files was marginally more effective than the on-demand scanner, though here detection was at least at a level which might be considered to provide adequate protection.

Disturbingly, for those folk who disapprove of macro virus upconversion, an option in the scanner triggered on occasion stating, in German, that the document scanned was of unknown format and offering to convert it to one which was known. Quite what the result of this would be is unknown, lest the wrath of the one known as Bontchev fall upon *Virus Bulletin's* unworthy collective pate.

Intel LANDesk Virus Protect v5.02

ItW Boot	96.4%	Macro	94.0%
ItW File	99.2%	Macro o/a	9.9%
ItW Overall	98.9%	Polymorphic	94.0%
ItW Overall o/a	60.1%	Standard	99.5%

Comparisons with the animal world fail with *Intel's* latest offering, since no creature as unsuited to its intended environment as *LANDesk* would ever have survived. The most heinous problem was encountered during the detection of certain boot viruses. When presented with Hare.7786, Hare.7610 or Moloch, the *LANDesk Virus Protect* simply crashed on-demand.

On-access, affairs were far worse. Scanning of these viruses turned off the on-access portion of the scanner completely, both for boot and file operations. This problem was occasionally noted at reboot with a message produced concerning debug errors but was not obvious from the actions of *LANDesk* either during or after the scan process. Since these viruses remained undetected by either on-access or on-demand scanning, this is a very serious flaw indeed.

Other problems were minor in comparison. Since *LANDesk* has no way of creating log records after scanning, infected files were simply deleted. This was fraught with problems, since it proved impossible to persuade *LANDesk* to delete read-only files. Having set all file attributes to allow deletion, there were still problems in that Cruncher was detected but the samples were not deleted.

iRiS AntiVirus v22.13

ItW Boot	100.0%	Macro	98.4%
ItW File	100.0%	Macro o/a	98.4%
ItW Overall	100.0%	Polymorphic	99.1%
ItW Overall o/a	99.6%	Standard	100.0%

A relatively little-known dark horse, *iRiS* supplies the scanning engine for *Cheyenne*, and the results of the two are unsurprisingly in accordance on-demand. Speed tests also show the expected similarities of a shared lineage and false positives are identical. On-access, however, very slight differences creep in with *iRiSAV* detecting WM/Leveller.A where *Inoculan* did not. The greatest difference is of a much more telling nature though, and is related to the stability and utility of the product.

Despite sporting some of the ugliest graphics around, *iRiSAV* produces good useful log files and no crashes occurred in these tests. This added stability is an anticipated side-effect of the *iRiS* team's use of their own virus detection code, as opposed to *Cheyenne's* aim of integrating *Inoculan* into many *CA* products.

Kaspersky Lab AVP v3.0 (build 124)

ItW Boot	100.0%	Macro	99.1%
ItW File	100.0%	Macro o/a	99.1%
ItW Overall	100.0%	Polymorphic	99.8%
ItW Overall o/a	100.0%	Standard	100.0%

Very much the pet beast of the newsgroup alt.comp.virus at the moment, *AVP* did not quite live up to its house-trained reputation in this showing. In general, detection was as good as ever, though macro viruses in general and XM/Compat.A in particular caused more problems than in the past. Boot virus testing resulted in the usual clean sweep of detection in both on-access and on-demand scanning modes.

On-access scans of the non-boot viruses were slightly more fraught. The first scan run produced a major seizure for the test machine, caused directly by an *AVP*-associated DLL. Retrying this gave no problems during the scan, yet directly

afterwards *Windows* hung when *Explorer* was run. Overheads on copy time with the *AVP* monitor were also a noticeable effect, running at close to 100%. Despite these problems detection remained exactly on a par with that shown on-demand and the possibility remains, as with other products, that some on-access problems are magnified by the sheer volume of infected files processed.

NAI Dr Solomon AVTK v7.87

ItW Boot	100.0%	Macro	98.6%
ItW File	100.0%	Macro o/a	98.4%
ItW Overall	100.0%	Polymorphic	97.6%
ItW Overall o/a	99.9%	Standard	100.0%

Rejoicing in possibly the longest name to be associated with anti-virus merchandise, this product was in fact the *Dr Solomon's* component, devoid of any *Network Associates* input. The aim of *NAI* being the selective breeding of a chimera of *McAfee* looks and *Dr Solomon's* detection, the choice of test subject comes as no surprise. Unhappily for those concerned, the slight stability worries which were apparent during boot sector testing in the past have become no better.

The first problems appeared upon installation, the screen outside the program window being transformed to a veritable kaleidoscope. The setup was its usual irksome self – the smallest changes to the on-access scanner still required a full reboot – a problem which one hopes will be not insuperable in the new generation of *NAI* scanner.

Previous problems with on-demand boot testing were in evidence again. Flame and Michelangelo.A both caused a complete hang of the test machine, offering no alternative to a potentially infecting reboot. With problems such as these appearing just as the tricky graft procedure for *NAI* and *Dr Solomon's* is occurring, there must be some doubt as to the stability of the combined program.

Norman Thunderbyte AntiVirus v4.10

ItW Boot	100.0%	Macro	95.4%
ItW File	99.7%	Macro o/a	73.9%
ItW Overall	99.7%	Polymorphic	94.8%
ItW Overall o/a	87.5%	Standard	98.2%

A product of evolution in action, *TBAV* now possesses an on-access scanner, though further changes are necessary before this new feature can be fully trusted. As ever, the prime feature of *Thunderbyte's* offering is its cheetah-like speed, though this was marred somewhat by the presence of nine false positives. These were all claimed to contain the HLLC.14795 virus. Being a high-level language virus, it seems more than likely that the part chosen to identify this virus is part of code commonly produced by the virus writer's compiler. Floppy scan rates were similarly speedy and with the new on-access scanner having minimal overheads there can be no complaints on this front.

On-demand scanning remains at the usual, reasonable level for *TBAV*, though a sprinkling of CIH misses is an area where improvements are a priority, and the detection of Marburg was far from perfect. Macro viruses too proved a particular weakness. *TBAV* does, in its defence, include an integrity checking component which might lessen the impact of these misses.

The on-access portion, however, exhibited instability and a bizarre detection pattern with DOT and DOC files. The first of each three DOC samples of most macro viruses was not detected. Viruses missed on-demand were again missed completely, and these should have been fully detected due either to age or simplicity.

There was also a number of spontaneous reboots and crashes during attempts to instigate on-access scanning. The on-access boot scans also proved a little unsatisfactory, with a significant number of misses, and poor detection of disk changes. In other areas on-access detection was very similar to that achieved with on-demand scanning, a few extra misses being in accordance with most other such scanners' performances.

Norman Virus Control v4.52

ItW Boot	100.0%	Macro	96.0%
ItW File	100.0%	Macro o/a	96.2%
ItW Overall	100.0%	Polymorphic	99.0%
ItW Overall o/a	n/a	Standard	99.7%

Norman Virus Control (*NVC*) remains its usual stable self, a beast which has found its habitat and stays there. The on-access part of the program remains something of a nonesuch, consisting of a standard macro virus detector, combined with an entirely heuristics-based, pre-execution 'behaviour blocker' for other file viruses. Boot viruses are also detected by pattern-based methods. For this reason only Boot and Macro test-sets were employed for on-access testing – attempting to execute all the samples would have been infeasible.

As ever, *NVC* was on its best behaviour, and testing was without any mishaps or adventures. The largest number of misses came in the macro virus collection, the polymorphic varieties proving problematical. Oddly, XM/Compat.A was detected on-access but not on-demand, possibly reflecting a difference in the databases used by both functions. Boot virus detection showed a couple of misses on-access but none on-demand, and time tests showed *NVC* to be just faster than average.

Sophos Anti-Virus v3.13

ItW Boot	100.0%	Macro	97.2%
ItW File	100.0%	Macro o/a	96.9%
ItW Overall	100.0%	Polymorphic	98.8%
ItW Overall o/a	100.0%	Standard	99.5%

Sophos already stables a selection of corporate beasts – a zebra, a rabbit and a penguin. Following an established tradition the *Sophos Anti-Virus* (*SAV*) tests were performed with no crashes or untoward happenings, log files being produced with no great stress on the reviewer's part. The same version of this program was featured in last month's review and, as might be expected, the only real difference was in the non-detection of some of the newer macros added to the test-set in the intervening month.

Stiller Integrity Master v4.01a

ItW Boot	97.6%	Macro	63.7%
ItW File	86.1%	Macro o/a	n/a
ItW Overall	87.3%	Polymorphic	30.7%
ItW Overall o/a	n/a	Standard	81.9%

Stiller Integrity Master (*IM*) is something of an oddity in these tests and reviewing it here is akin to comparing an elm tree to a variety of marsupial. As the name suggests, *IM* is primarily an integrity checker – in some ways not unlike *In-Defense* (see p.20). There is little point in having an integrity checker which is installed upon an already infected machine, however, and to this end *IM* pre-scans for known viruses before it produces its first integrity checksum database for a machine.

The scanner may also be utilized on-demand. However, *Stiller Research* clearly considers this scan to be of far from vital importance, providing updates to the virus list relatively infrequently, and trusting in its integrity checking to detect viral activity.

This lack of regular updates shows in the scan results, with polymorphic viruses proving a particularly problematical area for *IM*, code emulation presumably not being present in its repertoire of detection tricks. Against the In the Wild test-sets matters were better, though clearly date-related – the more recent samples remaining mostly undetected. Detection of boot viruses gave the best performance, not a surprise as this is an area where new viruses appear with much less frequency and the *Virus Bulletin* test-set is limited to those in the wild.

Speed-wise *IM* proved in the faster portion of the middle running, producing only one false positive in detecting a boot virus in a file (culled from an ancient virus scanner) that contains unencrypted scan strings. One of *IM's* companion virus detection heuristics is somewhat problematic when *Windows 98* itself installs both SCANDISK.COM and SCANDISK.EXE in the same directory.

Symantec Norton AntiVirus v5.00.01

ItW Boot	100.0%	Macro	99.8%
ItW File	100.0%	Macro o/a	97.7%
ItW Overall	100.0%	Polymorphic	98.7%
ItW Overall o/a	98.0%	Standard	99.7%

Resplendent in fine scarlet plumage and replete from the devouring of *Intel*, the question is whether *NAV 5's* image is the only difference. *NAV 5* is usually bundled with a host of nest-fellows but, possibly for legal reasons, the review copy arrived without them. This isolation may or may not explain the presence of a warning upon installation that *NAV* was 'unable to load auto-protect agent, logging… will not be available'.

This was not an ill omen, however, since the logging options available had, in fact, increased from those notable by their absence in the 4.*x* versions. On the whole, *NAV 5* showed improvement, with detection more worthy of *Symantec's* market share. One Marburg sample, the Navrhar VxDs and the macro virus W97M/Encr.A were the only samples missed on-demand.

On-access these joined a motley collection of mostly polymorphic macro viruses and the complete set of Marburgs. The macro virus misses here are presumably a result of the quest for low overheads, currently standing at about one hundred percent. However, the missing of Marburg is more of a disappointment, since it is not unlikely to be 'supplied' in archived material on CDs. In such cases on-access detection is of great importance.

Conclusion

The half-expected rash of new problems associated with *Windows 98* failed to materialize, though some differences in behaviour were apparent in comparison with the previously used operating systems. More disturbing, however, were the persistent problems remaining in an environment now several years old. Stability remains difficult to find in some well-established programs – this is becoming worse rather than better in more than one of the products tested.

The recent rise in polymorphic macro viruses caused by far the greatest percentage of misses. So far the polymorphism seen in macro viruses is quite simple, yet, for many products, dealing with it adequately will require some major redesign of internal macro handling functions. What the future holds is presumably more complexity in the viruses and perhaps a drop in detection while anti-virus companies get to the root of the problem.

Technical Details

Test Environment: Three 166 MHz Pentium-MMX PCs with 64 MB of RAM, 4 GB hard disk, CD-ROM drive and a 3.5-inch floppy, running *Windows 98*. The workstations could be rebuilt from disk images and the master copy of the test-set was held on a CD-ROM. All timed tests were run on one workstation.

Speed and Overhead Test-sets: Clean Hard Disk: 5500 COM and EXE files, occupying 546,932,175 bytes, copied from CD-ROM to hard disk.

Virus Test-set: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Win98/199811/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

PRODUCT REVIEW

In-Defense v2.10

[*This review digresses somewhat from a typical* Virus Bulletin *standalone review – apart from not covering a scanner, it has had to be split into two parts. The December issue will include details of our tests of* In-Defense's *virus detection and repair capabilities. Ed.*]

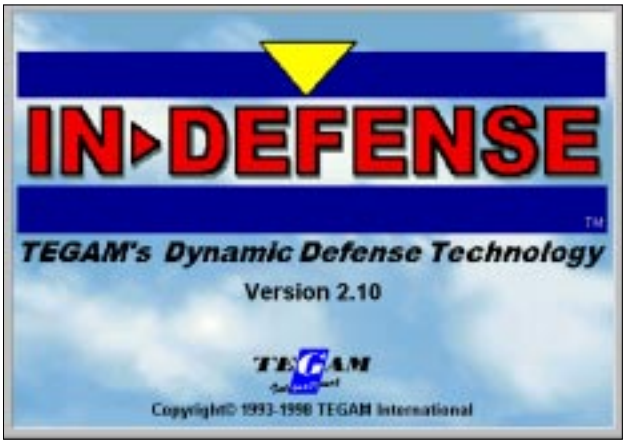
The latest entry in the 'we have completely redesigned anti-virus technology' stakes, *Tegam International* makes some very bold claims about universal virus detection and disinfection for its product, *In-Defense*. *Tegam* claims to have been marketing the product in France successfully and is now striving to break into the US and other predominantly English-speaking markets.

Claiming to represent a revolutionary new approach to anti-virus software, *In-Defense* is being marketed with some strong claims to infallibility. The 'generic' approach it takes to virus detection also claims to obviate the need for regular updates, although functionality upgrades are made available from time to time.

Packaging and Contents

The standard single licence product arrived in a typical software carton. Its fluffy white cloud background is somewhat reminiscent of the *Windows 95/98* logo screens, and the bold claims of 'Total Virus Protection' and 'Repels All Viruses – Known or Unknown' are presumably meant to attract the curious, if not the adventurous. Various information about other products in the range is presented on the sides of the box, where the system requirements for *In-Defense* on the various supported platforms are also clearly spelled out.

It was the back of the box that was to be marvelled at. Such excess of marketing hype had not made its way to this reviewer in quite some time (at least, not in the form of packaging adornment). 'Eliminates all known and unknown viruses', 'Eliminates all risk of infection', 'Prevents viral damage to hard drives', 'Does not slow system down' and 'No false alarms' are just some of the claims made for this product. Either the word 'all' had been roundly abused here or *In-Defense* was to prove truly wondrous.



Opening the box revealed a soft-covered, spiral-bound manual, a registration card and the software envelope. The last is interesting in that the seal over the flap reads 'By breaking this seal, you agree to read the License and Warranty Agreements' whereas the usual (and possibly unenforceable) practice is to suggest that opening the envelope indicates your acceptance of those 'agreements'. Client versions of the software for DOS, *Windows 3.1x*, *95*, *98* and *NT* were on the CD. Also included was a high-density, 3.5-inch 'Rescue Diskette'.

Convenient, pre-printed labels on these diskettes carry the *In-Defense* toll-free support number – presumably 'international' versions, as they become available, will have other than the US number. The rescue diskettes are machine-specific (which is normal) and fundamental to some recovery situations. Given this, a little more space should have been provided on the label to write the identification information for the machine to which the disk belongs. This could easily be achieved by reducing the *In-Defense* logo, which consumes more than a third of the label's area.

Documentation

Divided into five sections, the User Guide might normally be 150 pages long – there being 146 uniquely-numbered pages in the one received and at least four pages missing from the end of section five. It also, however, had an additional forty-odd pages, comprising a repeat of part of the third and fourth sections. If the guide normally has an index, that too was missing in the particular incarnation received for review.

Setting aside these binding problems, the layout is effective and easy to follow. In order, the sections are; Quick Start, User Guide, Network Administrator's Guide, Technical Notes, and Evaluating InDefense. This last section name is indicative of a small, but annoying, point – the developers and marketeers do not seem to be in agreement as to the product's name. Almost universally referred to as *InDefense* in the manual and packaging, the programs and help files equally adamantly refer to the product as *In-Defense*.

Diagrams and screen shots are generally used well in describing the installation routine and the functionality of the programs. Although the screen shots from the *Windows* programs are clear, those from the DOS programs are largely indecipherable. This should be easily fixable.

The manual is infused with a general tone of wonderment at *In-Defense's* 'new' approach to virus detection. Such an attempt to distinguish a 'different' product is understandable, but in places it verges on religious fervour, and many of the assertions about the novelty of the techniques used show a poor appreciation of the history of anti-virus software development. For example, integrity methods and behaviour monitors or blockers were amongst the very first anti-virus tools – the *In-Defense* view of history ignores this, claiming scanning was the first anti-virus technology.

Further, in the effort to establish the superiority of the *In-Defense* approach, some sections of the manual portray a misleadingly simplistic view of modern implementations of the other approaches, tending to caricature in places. One is left with an overall feeling that many of the claims surely are not meant to be taken seriously. Some are unintentionally humourous, such as the sub-heading 'What distinguishes it [*In-Defense*] from Anti-Virus technology?'.

Product Philosophy

The *In-Defense* manual plays long and hard on the inadequacy of existing anti-virus methods; especially scanners. It also takes great strides to differentiate itself from scanners. To understand the intensity of conviction required to buy into this belief, a couple of extended quotes from the *In-Defense* manual are presented here.

'"Known-Viruses Scanning" proved useful during the first few years of the virus scourge, but today it is completely ineffective.' The argument continues that with up to seven new viruses per day [*probably more like twenty in recent months! Ed.*], you have to update continually to remain protected. In theory this is true, but the reality is that an extremely high proportion of those new viruses will never make it to real world computers, let alone yours. Thus, we are told, 'the known-viruses scanning method is outdated and no longer offers a serious solution for people interetsed in security'. The article in this issue by Peter Morley (p.10), describing how detection and disinfection of in excess of 14,000 new viruses was added to his company's product in less than a week's work should be illuminating reading for the *In-Defense* developers…

In-Defense offers a 'new' approach 'based on the fact that there are only three families of viruses each having common characteristics. Using a set of powerful artificial intelligence analyzers and strategic logic protection layers, InDefense immediately detects any of these virus families and can then remove the detected viruses. This means that InDefense prevents your computer from being infected with viruses whether known or not. After all, prevention is better than cure.' The manual lays especially heavy importance on prevalence of macro virus incidents among infections.

Whilst naïvely appealing, the discussion of *In-Defense's* 'protection' methods raises a number of difficult problems for those versed in anti-virus technology. Behaviour blockers, integrity checkers and heuristic analyzers are not new as *Tegam* would have you believe, neither is the combination of them present in its product. The problems that such products have traditionally suffered are also well-understood. Detecting change is not the same as detecting a virus. Detecting 'usually bad' things means you will sometimes detect 'normal' things (disks do sometimes have to be reformatted).

Standalone Installation

Both the standalone version of *In-Defense* and the Administrator's Toolkit were supplied. The former is the main product, whereas the latter supplies a set of utilities for easing installation, management and upgrading of the software from a server.

The Administrator's Toolkit is usually bundled with a server licence and a number of client licences in what *Tegam* calls a Workgroup Starter Pack. For this review, a ten client pack was provided. No enforcement or built-in monitoring of licences was apparent, nor mentioned in the documentation. The Workgroup Pack included everything supplied for the individual licence (CD, rescue diskette, etc.) plus the Administrator's Toolkit on a write-protected diskette. Documentation for the latter is povided in section three of the standard manual. Installation and configuration options for the Administrator's Toolkit are covered in some detail later in this review.

Installation of the standalone product was straightforward. Once the displayed licence conditions were accepted a choice between a custom or typical installation was offered. As *In-Defense* includes an option to install a resident file access monitor, it recommends the prior uninstallation of any other anti-virus software already on the machine. Selecting a typical installation, the option of confirming the default program installation directory or the chance to specify another was offered.

With these preliminaries completed, it is simply a matter of confirming the installation options by clicking the Install button. At this point, less experienced users may become confused, if following the installation part of the User Guide section in the manual. It covers the Custom installation route, but without so much as a hint that this is the case or that there is another option.

Regardless of how the final confirmation stage is reached, once there, the Back button steps through the configuration options, in the reverse order that they would have been offered had a Custom installation been chosen. The additional options provided include electing not to make a rescue diskette and creating a second vaccination file with a user-supplied name.

Although the manual claims making a second 'vaccination file' is the recommended choice, the default installation type ('Typical') neither creates one nor presents the option to do so. The final options the Custom install offers are installing the resident protection and the macro virus

protection modules (both are installed by default under a Typical installation). The installation procedure seems to progress identically under *Windows 95*, *98* and *NT*.

Another quibble with the installation procedure revolves around the making of rescue diskettes. Should you decide not to make one after setting an installation rolling, your only option is to abort the whole installation process. Surely it cannot be that difficult to allow an opt-out at this point yet continue with the rest of the setup?

'Rescue diskette' is possibly too grand a term for the facility *In-Defense* provides. Under DOS and *Windows 9x*, its 'make rescue diskette' routine simply runs the SYS command and copies some files. It is good that the installation procedure all but forces the user to do this, as a clean boot diskette is the most basic recovery tool (and probably the most often missing) from a typical PC.

Unfortunately, depending on MS DOS for the emergency bootstrap means that a virus using the well-established (but fortunately not common) circular-partition 'trick' would lock you out of your machine despite you having prepared a 'rescue diskette'. Some vendors work around this problem by licensing or using OSes that do not suffer the fatal bug that renders MS DOS useless in the face of such partitions; others have written their own mini-OS to provide the disk and file-system services needed to run their products.

To be able to remove a virus utilizing circular partitions, *In-Defense* must be able to get the PC semi-operational. Its dependence on the user's OS for this means it will fail to meet its claim of recovering from all virus infections on PCs running operating systems based on MS-DOS (that includes *Windows 9x*).

Network Installation

The network server setup was tested using several combinations of *Windows 9x* and *NT* workstations and *NetWare 3.12* and *4.10*, and *NT*, servers. As chance would have it, the first combination tested was *NT 4* workstation and *NetWare 3.12* server. This proved surprisingly troublesome.

Being cautious, the standalone version of *In-Defense* was first installed on the workstation. Once this procedure was completed and the machine restarted, the *In-Defense Administrator's Toolkit* (*IDAT*) diskette was inserted into the A: drive. Imagine the surprise when, upon running NETSETUP.EXE, the resident protection module popped up a dialog box warning that the file was infected with a virus!

The best part of half a day was spent disassembling and analysing the program. From this investigation, the file neither appeared to be self-replicating nor harbouring anything that was. Attempts to prompt replication failed.

So much for the claims of no false alarms…

Assured, independently of *In-Defense*, that the test workstations were not about to be infected, the on-access file monitor was disabled and installation restarted. Faith in the product already being seriously shaken, the most polite description of the sequence of events that followed is 'most perplexing'.

From the off, the *IDAT* setup installed a local copy of *In-Defense* on the workstation. Maybe it was because the *IDAT* installer could not detect that the same version of was already installed? Regardless, the *In-Defense* installer itself detected that it was already installed. However, as the first thing it did was run the uninstaller, it is unclear whether it was aware it was uninstalling the exact same version as was about to be installed.

Anyway, under *NT* workstation uninstalling the existing version of *In-Defense* involved the stopping and unloading of the *In-Defense* service and the removal of the software. Then *In-Defense for NT* was re-installed. Things seemed to be running well. A prompt to replace the rescue diskette with the *IDAT* diskette was obeyed and *IDAT* started copying the software distribution directories from the CD to the chosen directory on the server.

Then the installation stopped running so well. In fact, the installation stopped. A dialog box warned that the folder INDEF32 could not be copied to the server and the only chain of options that did anything resulted in the *IDAT* installation aborting.

Attempting to restart the *IDAT* installer gave a clue as to what had happened. Double-clicking NETSETUP.EXE resulted in a dialog box warning that 'a device attached to the system is not working'. This warning was received earlier, after clicking away the sequence of 'Infected File!' false alarms. But why was there no 'Infected File!' warning as seen earlier? The penny dropped – the setup process had installed, loaded and started the service that does the real-time file system monitoring, but the helper application that handles alerts from, and user interaction with, this service would not load until the next restart.

So, no informative warning and the OS saw a timeout. But why had the process stopped at all? Comparing the contents of the INDEF32 directories on the server and CD, it became apparent. All the files on the CD had been copied, down to MACREMOV.DOT (whose presence and purpose, by the way, are not explained anywhere). The on-access service was not happy about this *Word* template file and depended in some way on the 'missing' interface component to deal with it. The failure of that component to respond presumably caused the service to block access to the file and the copying process eventually timed-out with an OS error.

Installation of *IDAT* was eventually achieved by disabling the resident file monitor, starting the installation (which uninstalled then re-installed *In-Defense*), then stopping the service that had just been re-installed and started. Following that, swapping the rescue diskette for the *IDAT* diskette resulted in the files being copied from the CD to the appropriate place on the server and the administration software being installed on the workstation. A more tortuous installation procedure is difficult to imagine, and it is certainly not something the less than expert should attempt.

Network Administration

The Administrator's Toolkit was a fairly simple program, which allowed configuration of the standard settings of *In-Defense*. These settings are those that will be used in copies of the program installed from the server. Various functions of *In-Defense*, when installed from a administrator's central setup, can be password-protected (including preventing the alteration of crucial configuration options). The Toolkit provides no mechanism for treating users or machines in groups or domains – the only way this seems possible is by maintaining a server installation with its own configuration options for each such group.

The toolkit allows setting the options used by the setup program itself. It also provides for an option otherwise unavailable in the standalone setup program – storing the rescue diskettes on the server – rather than requiring a diskette at the PC. Given that the installation will abort if the user tries to get around making a rescue diskette, use of this option seems advisable. An interesting observation in working with this part of the program was that the Default button, which should reset all options in the current dialog to their original state, sets things to a 'recommended' (and more conservative) state than the preset.

Centralized reports from *In-Defense* client machines can be viewed from the toolkit's console. There is also a facility for viewing and creating the client rescue diskettes. This latter does not necessarily work as you may suspect – only a minimum of information that would be on the physical diskettes is stored on the server, and making the diskettes for DOS and *Windows 9x* is dependent on the SYS command being available. From the toolkit console on an *NT* workstation, such rescue diskettes cannot be created.

The last element of the Administrator's Toolkit is Macro Pass. As in some other products, this allows the 'authorization' of macros developed in-house (or from otherwise trusted sources) for use on the network. As in other parts of the package, the exact functioning of this feature is not explained fully enough in the manual and systematic testing did not sufficiently clarify the processes employed either.

For example, after authorizing a harmless AutoOpen macro with Macro Pass, two side-by-side *Windows 95* workstations were logged off the server and then back on. One of them then allowed copying of a document (previously blocked) that contained this macro, but the other did not. Several logout/login cycles (including a couple of complete restarts) later the second machine updated to allow copying of the file in question. Similar peculiarities in returning the same document to 'suspicious' status occurred when the macro's authoriazation was revoked and the workstations logged out and back in.

In fact, this kind of inconsistency was apparent in many places in the product. Another example was that persistent, repeated attempts to copy a macro-bearing document (it was infected with WM/Cap.A) from a floppy to the hard drive of a *Windows 95* machine running *In-Defense* in its default standalone configuration eventually succeeded. It seems that not all file I/O is intercepted (or at least, not consistently handled by *In-Defense* when intercepted). This latter problem was especially noticeable in a DOS box where approximately one in ten attempts to copy virus-infected document files succeeded, but more persistence could also demonstrate the effect with Explorer.

Returning to the Administrator's Toolkit, the last thing in preparing to roll *In-Defense* out to the network is to modify the server's login script. NETSETUP displayed the suggested additions to make to the login script depending on the server type, but did make the changes. The contents of the dialog box displaying the suggested changes can be selected and copied to the clipboard, but a button to do this (and thus make it obvious) would be a nice addition.

The required script change causes a version checker to run on the workstation. This supposedly tests that the machine has the latest server-installed version of *In-Defense*, all critical files and the latest server-hosted configuration. If not it launches the full installer or transfers the updated configuration options and the like.

At least, that is the impression the manual gives. Testing suggests it provides somewhat less than this. On one of the test machines the helper program in the system tray that allows changing settings (if this has not been disabled by the administrator) was deleted and repeated logout/login cycles saw *In-Defense* reinstalling itself, but failing to reinstall the deleted file (AVLOAD32.EXE). Other times configuration updates caused an internal error in Explorer, which suggested a system restart as a result.

Initial Conclusions

In-Defense might suit you, but it also might cost expenditure of a great deal of time and hassle before convincing you that it is not for your environment. The apparent lack of thorough testing of common configurations (as evidenced in the initial Administrator's Toolkit installation), and hints of program instability and unreliability, call the product's quality assurance into question. This and the lack of central administration options necessary in medium scale networks means that if *Tegam* wishes to crack the corporate market, as claimed, it will have to address these issues rapidly and thoroughly. The product's performance in detecting and disinfecting viruses is detailed in the next issue of *VB*.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 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

Compsec '98 takes place from **11–13 November 1998, at the Queen Elizabeth II Conference Centre in London, UK.** The agenda includes an exhibition, a pre-conference workshop on 10 November and the Seventh Annual Directors' Briefing on 13 November. For details and a registration form, contact the conference secretary Amy Richardson; Tel +44 1865 843643, fax +44 1865 843958, email a.richardson@elsevier.co.uk, or visit the current Compsec '98 web site http://www.elsevier.nl/locate/compsec98/.

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

***Data Fellows* announce the release of *F-Secure FileCrypto for Windows NT 4.0*.** Designed for organizations with thousands of computers, it provides strong, on-the-fly encryption for confidential data. *Data Fellows* plans to release *F-Secure FileCrypto for Windows 95/98* next year. For more information, contact the product marketing manager Tom Helenius in Finland; Tel +358 9859 900, fax +358 98599 0599 or email Tom.Helenius@DataFellows.com.

The Internet Society is a non-government organization for the global cooperation and coordination for the Internet and its internetworking technologies and applications. **Registrations are now being taken for *the Internet Society's* 1999 Network and Distributed System Security (NDSS) Symposium.** The 6th annual NDSS Symposium provides a mix of technical papers and panel presentations, covering all aspects of Internet security. Associated features include pre-conference technical tutorials and sponsorship opportunities. The event takes place from 3–5 February 1999 at the Catamaran Resort Hotel in San Diego, California, USA. An early booking discount applies to all registrations taken before 6 January 1999. For more information contact the Internet Society; 12020 Sunrise Valley Drive, Reston, VA 20191, USA, tel +1 703 648 9888, fax +1 703 648 9887, or email ndss99reg@isoc.org. On-line information is available at http://www.isoc.org/ndss99/.

The 14th Annual Computer Security Applications Conference (ACSA) takes place at the Radisson Resort Scottsdale in Phoenix, Arizona, USA from 7–11 December 1998. The two and half day technical conference exploring the application of computer technology will be preceded by two days of tutorials dealing with policy matters, technology applications, etc. Introductory courses will be offered as well as advanced courses exploring specialized technology. There is a full social programme and an award for the most outstanding paper presented at the conference. For more information about registration; Tel +1 407 628 3602, fax +1 407 628 3186, or access the conference web site at http://www.acsac.org/.

Novell Inc* has entered into a partnership with *Network Associates to provide comprehensive and fully-integrated corporate anti-virus software. *NetShield* is available with the newly announced *NetWare for Small Business 4.2* at the retail price of $100 for five users.

In October *Information Security* magazine launched ***Security Wire Daily*, a daily on-line news service** devoted entirely to information security topics and issues. It is published by the *ICSA* and is available free to the public at http://www.infosecuritymag.com/securitywire/. Each weekday, *Security Wire* will feature several articles focusing on news, current affairs and events. Daily editions will also feature market news and analysis, product announcements and conference updates. The *ICSA* invites the submission of news tips and announcements to securitywire@infosecuritymag.com or fax +1 781 255 0215.

An introductory computer virus workshop on 20 January 1999 will be followed on 21 January by an advanced session at the *Sophos* training suite in Abingdon, UK. To register for a place, contact Karen Richardson; Tel +44 1235 544015, fax +44 1235 559935, or find details at http://www.sophos.com/.

The eighth annual *Virus Bulletin* conference took place at the Munich Park Hilton from 22–23 October (see the conference report in this issue, starting on p.6). **If you did not catch the conference, copies of the VB'98 proceedings are available on CD.** The price of the full proceedings is £150 or $250 (DM 450). For more information,please contact conference co-ordinator Jo Peck at the *Virus Bulletin* offices; Tel +44 1235 555139, fax +44 1235 531889, or email Joanne.Peck@virusbtn.com.