ISSN 0956-9979 JUNE 1998

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **May madness?** The News pages are straining at their seams this month. With reports of a new Macintosh worm (or virus) and the first PowerPC-specific one to boot, major product and technology shuffles among three large US anti-virus vendors, and the retirement of the popular Macintosh anti-virus program *Disinfectant*, there is bound to be something of interest on pp.3–5.

• **Cross dressing:** The first macro virus that tries to infect *Microsoft Office* files other than those of its host is examined in the first of this month's virus analyses, starting on p.11.

• **Regarding Romania:** Another in our occasional series of regional reports on the virus scene around the globe. *GeCAD's* Costin Raiu outlines Romania's experiences with computer viruses on p.8.

CONTENTS

EDITORIAL

NEWS

FEATURE

OPINION

VIRUS ANALYSES

CONFERENCE REPORT

PRODUCT REVIEWS

EDITORIAL

What's in a Name?

An interesting question I'm sure will not be resolved here! Perhaps Shakespeare felt there was little in names – what else may have moved him to write 'A rose by any other name would smell as sweet'? Opinions vary though. Far removed from the world of the romantic sonnet writer (and probably at the opposite end of several relevant spectra) the philosopher of science Lakatos was fond of saying 'Labels should not matter, but they do'.

Roses, sonnets, philosophy – what have these to do with viruses? Admittedly not much. However, they do nicely illustrate a wide divergence of opinion on the importance of names. Similar divergence of opinion can also be seen in the anti-virus industry. What we commonly refer to as 'the naming problem' has, in various ways, been at the forefront of my thoughts the last few weeks.

> *" developers seem quite resilient to changing the names they have selected "*

When many of the 'established' sciences were 'new' (or, more correctly, at the time that historians of the 'successful' knowledge-seeking endeavours pin-point as that disciplines' beginning), the pioneers were often heavily into cataloguing. Devising naming schema and categories, into which everything about the subject matter could be pigeon-holed, was a major focus of study.

There are those in the anti-virus industry who feel that such activities are important to the anti-virus effort. Highly structured naming schemes stand in testimony to this age-old categorization effort. Witness names such as Win95/Memorial.12413, WM/Npad.A (through to .HS as of this writing), WM/Rapi.A (and its .A1 and .A2 offspring), Stoned.Standard.A and so on. Whether there is any long-term value in extant naming schemes *per se* cannot be decided this early in the history of anti-virus research, but there are those among us who value the ideal of having *one* naming scheme.

Many, however, seem to take the approach that so long as they avoid names already in use in their own product, their name for a new virus matters little. VGrep, the virus name cross-referencing system developed by *VB's* previous editor (http://www.virusbtn.com/VGrep/), demonstrates both that this happens and the near-chaotic namespace mess that has ensued. A few minutes searching VGrep for semi-randomly chosen words will quickly throw up several examples of a single virus that is named four, five, six or more quite different things, and that is just among fifteen products.

Of course, the value of all developers using the same name for the same virus is analogous to that seen in most other walks of life. Imagine the mess if the person you contracted to paint your house white used the word 'white' for the colour we call 'lime green'. Yes…

Few anti-virus developers seem to care much that they call a virus Indonga that another calls Eater, and others variously call IND3652, CODE1289 and Pindonga (I have ignored the fact that most names in this sample from VGrep suggest different infective lengths!). This tends to not be quite so problematic with very common viruses but, in general, developers seem quite resilient to changing the names they have selected. Such issues were particularly salient when I was trying to resolve which Avispa variant was in our ItW File test-set this month, following enquiries about a test result from an anti-virus company.

The more interesting question though, arises from the appearance of Cross (see p.11) and a more complex macro virus that 'cross-infects' between *Word* and *Excel*. The naming issue here is not closely related to that of using the same name for the same virus, but more that of what 'system prefix' to use for such viruses. Most virus naming schemes prefix the name so as to indicate the platform where it is infective (PC boot and DOS file infectors tend to have no such prefix).

Cross- platform viruses could simply be treated as two (or more) separate platform-specific viruses, each taking the prefix appropriate to its 'native' form. The unfortunate offshoot of this is that the multi-platform nature of the virus is then not immediately apparent from its name, and faced with detection of one form of such a virus, I feel you should be warned that a form of the virus for some other platform may also have been released on your machine. I am looking forward to the resolution of this issue, if in fact there will be one!

NEWS

Disinfectant Retired

This letter from John Norstad of Northwestern University was posted on *Disinfectant's* official distribution site, ftp://ftp.acns.nwu.edu/pub/disinfectant/, on 6 May:

> I regret that I am officially retiring *Disinfectant*, our free anti-viral utility for the Macintosh. The current version, 3.7.1, is the last version. *Disinfectant* will not be updated for the new Autostart 9805 worm [*See next item. Ed.*] or for any future viruses, worms or other Macintosh malware.
>
> I made this decision not because of the new Autostart 9805 worm, but rather because of the widespread and dangerous *Microsoft* macro virus problem. I believe that there are now well over 1000 of these viruses, and many new ones are discovered every month. They are now a much more serious problem for Mac users than are the classic system viruses. I simply do not have the resources to combat a problem which is this huge in scope and complexity.
>
> I am aware that some Mac users do not use *Microsoft Word 6* or *Excel 5* or later versions, and hence have still found *Disinfectant* useful. These people seem to be a minority, however. The majority of Mac users need a commercial anti-viral product. *Disinfectant* is not adequate protection, and hasn't been for several years. For this reason, I feel that there is little point in updating the program for the new worm. Doing so would, in fact, only provide a false sense of security, and result in more harm than good.
>
> The following commercial anti-viral utilities are currently available for the Macintosh. All *Disinfectant* users should switch to one of these products:
>
> - *Anti-Virus Toolkit – Dr Solomon's*
> - *SAM – Symantec*
> - *Virex – Dr Solomon's*
> - *VirusScan for the Mac – Network Associates*
>
> I began working on the Mac virus problem and *Disinfectant* ten years ago, in the spring of 1988, when the first Mac viruses began to appear. *Disinfectant 1.0* was released to the public on 18 March, 1989. I have been enormously gratified by the success of the program and its very kind reception by the Macintosh community. I'd also like to thank my many users for their support and encouragement over all these years. I'd also like to express my appreciation to the other members of the Mac anti-viral research community for their outstanding spirit of cooperation and public service which has made all of our products possible.
>
> Nine years is a long run for any kind of computer software. It's time to move on ∎

Prevalence Table – April 1998

Virus	Type	Incidents	Reports
Cap	Macro	68	16.8%
Laroux	Macro	42	10.4%
AntiExe	Boot	28	6.9%
Concept	Macro	20	5.0%
Form	Boot	17	4.2%
AntiCMOS	Boot	16	4.0%
Parity_Boot	Boot	16	4.0%
Empire_Monkey	Boot	14	3.5%
Ripper	Boot	13	3.2%
Wazzu	Macro	11	2.7%
DelCMOS	Boot	9	2.2%
NYB	Boot	9	2.2%
Dodgy	Boot	8	2.0%
Npad	Macro	7	1.7%
Muck	Macro	6	1.5%
Junkie	Multipartite	5	1.2%
Mental	Macro	5	1.2%
Stealth_Boot	Boot	5	1.2%
Angelina	Boot	4	1.0%
Appder	Macro	4	1.0%
Eco	Boot	4	1.0%
Galicia	Multipartite	4	1.0%
Rapi	Macro	4	1.0%
Showoff	Macro	4	1.0%
WelcomB	Boot	4	1.0%
CSV.5536	File	3	0.7%
Exebug	Multipartite	3	0.7%
MDMA	Macro	3	0.7%
Schumann	Macro	3	0.7%
Stoned	Boot	3	0.7%
USTC.7680	Multipartite	3	0.7%
Others [1]		59	14.6%
Total		404	100%

[1] The Prevalence Table includes two reports of each of the following viruses: ABCD, Cruel, Goldfish, Imposter, Jimi, NiceDay, Niknat, One_Half, Quandary, Sampo, Spanska.4250, Swlabs and Wolleh; and one report of each of: Alien, Baboon, Bandung, Colors, Demon, Die_Hard.4000, DZT, Edwin, Elvira.239, Flip, Hare.7786, Int12, INT-CE.2560, Johnny, Joshi, Jumper.B, LBB_Stealth, Leonardo.2085, Lunch, Maverick.2048, MPCa.1204, MZBoot, Nomenklatura, Nottice, Nov17th-8556, Obscene.2374, Rehenes, Shell.10634, Stoned.Stonehenge, Tequila.2468, Thery, Trackswap and V-Sign.

New Macintosh Worm

It is not every month we receive news of a new piece of native Macintosh malware (as opposed to new macro viruses that run in the *Word* or *Excel* environments provided on both *MacOS* and *Windows* operating systems). Thus, the discovery of a new worm, and then two further variants of it, during the first three weeks of May must seem like an outright frenzy of activity to Mac malware analysts.

AutoStart 9805 was discovered early in May and there are reports of it being in the wild in Asia. It takes advantage of the Quicktime 2.0 AutoStart feature, whereby the act of simply mounting a volume – be it inserting a diskette or some form of removable cartridge – can be made to cause virtually anything to be executed from the just-mounted media. In the case of AutoStart 9805 the 'anything' is a small program that copies itself to the system's Extensions folder, thus ensuring it is run at each subsequent system startup. When executed, AutoStart 9805 runs as an invisible background application, named Desktop Print Spooler. Mac users should not confuse this process (which is not visible in standard task listings, but can be seen with tools such as *Process Watcher* and *Macsbug*) with the quite legitimate Desktop Printer Spooler *file* on their systems.

Once active, AutoStart 9805 regularly checks all mounted HFS and HFS+ volumes (including server volumes in the initial version) and spreads to any that are not already infected. [*That makes it a virus to me. Ed.*] Unfortunately, this is not all – AutoStart 9805 has a malicious payload. It searches out files with names and sizes matching various criteria and corrupts them by overwriting the beginning of the data fork with up to about 1 MB of random data.

The AutoStart 9805-B variant has many minor differences from the original. It uses slightly different file and process names, triggers its spread and payload mechanisms more often (and limits the number of files it will damage in one payload activation) and avoids server volumes. It also deletes copies of the original version, if found when spreading to other media. Lastly, it will stop spreading completely after 24 December 1998.

The AutoStart 9805-C variant uses file and process names that are different again. It replaces other AutoStart 9805 variants with itself, and after 8 June 1998, will attempt to disinfect itself from all visible, infected volumes.

Macintosh anti-virus products have been updated to deal with the AutoStart family, and Macintosh owners should check the web sites of their anti-virus vendors. Users of Macintoshes without anti-virus software installed should note that AutoStart 9805 is a native PowerPC application, so cannot affect older, 680*x*0-based machines. If you use a PowerPC-based Macintosh and currently have no anti-virus software, you can protect yourself, if not already infected, by disabling the CD-ROM AutoPlay option from the Quicktime Settings Control Panel. *VB* will publish a detailed analysis next month ▮

Big Blue for Symantec?

The anti-virus productscape has undergone some significant changes through the years. Vendors and products have come and gone; sometimes through marketing or product failure, sometimes through acquisitions. Some of these changes have been all but predictable, whereas others have been 'out of the blue'.

Recent announcements by *IBM*, *Symantec* and *Intel* are certainly of the latter nature. No-one in the anti-virus industry with whom *VB* discussed this was expecting such moves and all seemed surprised, although this passed as the implications of the move were analysed.

On 19 May, *IBM* and *Symantec* made a joint press release announcing a slew of product and support changes, technology sharing, cross-licensing deals and the like. To summarize, *Symantec* and *IBM* have agreed to jointly develop virus detection technologies (particularly *IBM's* much vaunted 'immune system') and market them in *Symantec's Norton AntiVirus* (*NAV*) product line. The entire *IBM AntiVirus* (*IBMAV*) product range has been withdrawn and *Symantec* has taken over support and OEM contracts on those products – *IBM* will recommend its customers use *NAV*. *Intel* will incorporate current *IBM* virus detection technology into the anti-virus components of its *LANDesk Virus Protect* and related network management products.

For *IBMAV* customers, the biggest news must be the withdrawal of that product line. *IBM* will continue to provide virus signature updates and defect upgrades until the end of this year, but no feature upgrades beyond the currently shipping version (*IBMAV v3.0.2*) will be forthcoming. This means that if a new form of virus is discovered, requiring completely new detection methods, *IBMAV* will be unable to detect this type of virus. Traditionally, this would be a small concern, but with the deployment of VBA5 as the macro language of ever more applications, we may be at the begining of a steep rate of increase in the appearance of new forms of virus.

With both *IBM* and *Symantec* urging existing *IBMAV* users to move to *NAV*, it is not surprising the two developers claim to be cooperating on a migration tool to ease the rollout of such a significant software changeover. The complete removal of *IBMAV* from marketing may worry those preferring to make the changeover to *NAV* later rather than sooner, who need additional *IBMAV* licences in the meantime. To cover this transitional period, *Symantec* will (until the year's end) fulfil orders from existing *IBMAV* customers for additional licences.

Seizing a marketing opportunity, *Dr Solomon's* announced a free 'cross-grade' for corporate *IBMAV* users. This offer pushed the line that these potential new *Dr Solomon's* customers have been 'abandoned by *IBM* and handed over to *Symantec*'. Network Associates followed the next day with a dollar-for-dollar credit offer towards its Net Tools Secure anti-virus and security suite.

A common reason for choosing *IBMAV* has been its availability for *IBM's OS/2*. *Symantec* has undertaken to develop a native *OS/2* version of *NAV*, and further, to provide scanners for *Lotus Notes* and *Domino* servers running on *IBM's AIX*, *OS/2*, *OS/400* and *S/390* platforms. This should prove an interesting challenge for the *NAV* developers, so heavily in the Wintel camp until now.

Among all the *IBM* and *Symantec* news, the *Intel* announcement of its integration of the current *IBMAV* engine into its *LANDesk Virus Protect* (*LDVP*) products seems to have been all but overlooked. This move should be good news for *Intel's* customers. Although the *LDVP* suite has been a good choice from a network administrator's viewpoint, its virus detection rate has been less than convincing. In recent *VB* comparatives, *IBMAV* has consistently out-performed *Trend Micro's PC-cillin* (whose detection engine is currently used by *Intel*). With its higher detection rates and lower false positives, incorporating the *IBMAV* engine into the *LDVP* range should strengthen *Intel's* place in the heavily managed/centralized IT market segment.

Some of the first commentators on the moves talked of *Symantec* 'buying' *IBM's* anti-virus technology – specifically the immune system. This is not the case. Apart from the withdrawal of the *IBMAV* product family, the main announcements involve technology and product licensing agreements. Thus, *Symantec's* products will be the first to incorporate *IBM's* immune system technology, but this is not an exclusive agreement. The joint press release hints at *Intel's* interest in incorporating the immune system into *LDVP*, when that technology is further developed ▮

McAfee UK PR Company in Virus Distribution Shocker!

In their day-to-day business, *VB* staff receive many email messages, and, occasionally, these contain attachments. Being a magazine, this is probably not that surprising. Most of these attachments are 'documents' containing copy for the publication, but suspect files for analysis and the odd spreadsheet make their way here as well.

Thus it was not unusual, in the course of researching the *McAfee NetShield* review in this issue, to receive an *Excel* spreadsheet from *McAfee's* UK PR company, *Copithorne & Bellows*. What was surprising was the interloper travelling with the requested pricing information.

It was not some new, unknown or uncommon virus, but the original *Excel* macro virus – XM/Laroux.A. One wonders which anti-virus program *Copithorne & Bellows* use… ▮

Not to be Outdone…

In light of the previous item and considering that *Microsoft* has recently decided to package *McAfee VirusScan* with *Windows 98 Plus!*, it is ironic (perhaps 'fitting') that just before going to print *VB* received a report from Woody Leonhard (*Microsoft Office* guru and producer of the WOW Newsletter) that *Microsoft* had sent an infected *Excel* spreadsheet to its New Orleans TechEd conference speakers. The spreadsheet was a schedule and initial information suggests the virus was a Laroux variant. At press time the exact identity of the virus had not been confirmed ▮

Blitz those Viruses

The May issue of the *Armed Forces Communications & Electronics Association* newsletter, *Signal*, published an article that aroused some interest in the anti-virus industry. In breathless terms it claimed that in 'a significant Internet breakthrough that could enhance electronic commerce and protect sensitive corporate and government data, computer scientists have developed a new virus that automatically launches a lethal counter offensive against hackers.'

Variously describing this Blitzkrieg server as 'a radical digital life form', 'a self-programmed, fault-immune, ubiquitous virus-like system' and 'an offensive weapon for information warfare', the piece read more like poor science fiction than serious journalism. Anti-virus researchers regularly in touch with *VB* suggested that it may have originally been intended for the previous month's issue!

The full story behind this is even better – interested readers should look at Crypt Newsletter's coverage of it at http://www.soci.niu.edu/~crypt/other/blitz.htm ▮

Errata

The April review of *eSafe Protect* claimed, incorrectly, that *eSafe* detected 90.9% of the Polymorphic test-set. It is now clear that 337 samples were missed from the 13,500 in the Polymorphic test-set, giving a simple detection rate of 97.5%. Using the calculation scheme normally employed in *VB* comparative reviews, the results produce a detection rate of 95.4%. *Virus Bulletin* apologizes for this error.

Two errors from the May comparative review also need correcting. The Technical Details box at the end of the review incorrectly claimed the test machines were running *NT*, rather than *Windows 95 (SP1)*.

Further, *Sophos* queried *VB's* Avispa.D samples – the virus that caused *SWEEP* to miss a VB 100% award. It transpires that the samples *SWEEP* missed are not Avispa.D. They are viral and replicate, and all other products in the review detected them as some form of Avispa, as *SWEEP* itself has done in the past. However, they are not samples of the same virus as the Avispa.D in the *WildList Organization's* 'reference set'. Genuine Avispa.D replicants have been generated from a reference sample supplied by the *WildList Organization* and these will replace the Avispa samples in our In the Wild File test-set. In re-testing, the reviewed version of *SWEEP* detected these samples, thus *Sophos* has been granted a May VB 100% award. No other results are affected ▮

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

Bachkhoa.4426

CER: An encrypted, stealth, 4426-byte appender containing the texts ‘Ha Noi University of technology Your PC was infected by Bách Khoa virus’, ‘CHKLIST.MS’, ‘CHKLIST.CPS’, ‘FILESIGN.SAV’, and ‘FILE_ID.DIZ’. Infected files have their time-stamps set to 62 seconds.

```
Bachkhoa.4426       2790 33D9 538B D583 C402 8BEC 83ED 0231 4E00 83EC 028B EAC3
```

Cafe.667

PER: A companion, 667-byte virus containing the texts ‘NET$ERR.$$$’, ‘LOGIN.EXE’ and ‘Enter your password:’. The ‘Are you there?’ call, Int 21h, AX=0DEADh, returns AX=0CAFEh.

```
Cafe.667            3DAD DE75 04B8 FECA CF3D 004B 7403 E9F3 00E9 F500 FC57 5606
```

Cleaner.937

CER: An encrypted, 937-byte appender containing the text ‘☺ HDD-CLEANER Version 2.0 ☺ ☺ Copyright (c) 1997 (1st JAN) ☺ ☺ Made in Hungary, Sopron ☺ DESTRUCTION IN PROGRESS...’. The payload, which triggers on 9 August, overwrites the first 16 sectors of the first physical hard drive.

```
Cleaner.937         B440 BA00 01B9 A904 81E9 0001 CD21 E8C1 FFB8 2125 BA50 02CD
```

CodeRed.7428

CN: A slightly polymorphic, encrypted, appending, 7428-byte direct infector containing the texts ‘(θd∊ ϒΣd is (c)97 By THe GaBBeR’ and ‘The Code_Red Virus Has Stopped Your System’. These templates cover all infected files.

```
CodeRed.7428        8DB6 861B B938 0181 34?? ??46 46E2 F8C3
CodeRed.7428        8DBE 861B B938 0181 35?? ??47 47E2 F8C3
```

Djin.133

CER: A 133-byte overwriter containing the text ‘[DjiN_DjiN] iS MaDe By THe GaBBeR’. Infected files have their date and time-stamps all set to zero.

```
Djin.133            B800 4233 C999 CD21 B985 00B4 40BA 0001 CD21 9933 C933 D2B8
```

Dune.728

CR: An encrypted 728-byte appender containing the text ‘DUNE 1.3-ROM COMMAND.COM’. Infected files have their time-stamps set to 62 seconds and the word C350h at offset 0003h.

```
Dune.728            BD36 02B9 A200 2630 4600 2600 4E00 45E2 F52E 8B1E 2402 2E8B
```

Evul.109

CN: An overwriting, 109-byte, fast direct infector containing the texts ‘*.C*’, ‘SHIT’, and ‘EVUL’.

```
Evul.109            B440 B96D 00BA 0001 CD21 B43E CD21 FE06 5601 803E 5601 0A74
```

Evul.264

CN: An overwriting, 264-byte, direct infector containing the texts ‘*.COM’, ‘John & Connie by: EvuL 1997....’ and ‘I Love You, Connie... john 1997’.

```
Evul.264            BA00 01B4 40B9 0801 CD21 90B8 0157 8B16 B801 8B0E BA01 CD21
```

Evul.436

CN: An encrypted, appending, 436-byte, direct infector which infects five files at a time. It contains the text ‘Hello, World... If you like this virus, E-Mail me at COMPKILLER@JUNO.COM EvuL FuKKiN RuLeZ.. I will live forever...   You loose!’. Infected files have the byte 43h (‘C’) at offset 0003h.

```
Evul.436            CD16 E800 005D 81ED 1801 E803 00EB 0F90 B907 01BE 3101 8032
```

Evul.480

CN: An encrypted, appending, 480-byte, direct infector containing the texts ‘ha ha ha ha ha... EVUL RULEZ’, ‘EvuLz Dancer has waltzed into your puter Dance-Macabre Dream EvuL’, ‘Dance Macabre’, ‘EvuLz Nightmare’, ‘dreamevul’, ‘[dm]’, ‘[de]’ and ‘[Dance MAcabre v1.01]’. Infected files have the byte 43h (‘C’) at offset 0003h.

```
Evul.480            E813 00B4 40B9 E001 8D96 0601 CD21 E805 00B4 3ECD 21C3 8DB6
```

Exorcist.212

CN: A 212-byte overwriter containing the texts ‘[RED MEASLES]’, ‘Bad command or file name’ and ‘Exorcist[DC]*.com’. Infected files have the word 4344h (‘DC’) at offset 0003h. The payload, which triggers on the first of each month, tries to overwrite the first 999 sectors on the A:, B: and C: drives.

```
Exorcist.212        B802 3DBA 9E00 CD21 93B4 40B9 D400 BA00 01CD 215A 59B8 0157
```

F117.1079

CR: An encrypted, 1079-byte appender containing the texts ‘[F-117] STEALTH’, ‘(c)1997 The Netherlands’, ‘THe GaBBeR’ and ‘ANTI-VIR.DAT’. The virus corrupts EXE files starting with ‘ZM’.

```
F117.1079           EB02 EB4C B937 058D BE63 01BA 0100 33C0 EB06 32E4 CD1A 92C3
```

Feliz.1060

CEN: An encrypted, 1060-byte appender containing the texts ‘≡ Virus XAVIRUS HACKER, parido en Paraguay. ¡Feliz cumpleaños Javier! ≡’, ‘*.EXE’, and ‘*.COM’. Infected files have the byte 58h (‘X’) at offset 0003h (COM) or the word 4858h (‘XH’) at offset 0012h (EXE).

```
Feliz.1060          B440 B924 048B D5CD 21E8 8700 E8A7 00B4 40B9 1A00 8D96 2004
```

Glew.4245

ER: A mildly polymorphic, stealth, 4245-byte appender containing the texts ‘KEY.SIG’, ‘TB’, ‘FV’, ‘F-’, ‘VS’, ‘AV’, ‘VIR’, ‘HIE’, ‘OOLK’, ‘UARD’, ‘SCAN’, ‘CLEA’ and ‘MMAN’. The payload overwrites the contents of a first physical drive, displaying the message ‘A la Memoria de Cevallitos-= RATA de GLEW virus =-’. Infected files have their time-stamps set to 62 seconds. This template detects the virus in memory only.

```
Glew.4245            B8FF 35CD 213D CACA 747B B800 16CD 2F3D 0400 750A B835 2186;
```

Hell.797

CR: An encrypted, 797-byte appender containing the texts ‘▌H▌E▌L▌L’, ‘(c)1997 THe GaBBeR’, ‘????????.COM’, ‘*.COM’ and ‘*.*’.

```
Hell.797            368B 2D81 ED05 012E 803E 6501 B974 5DB9 1D04 8DBE 6501 BA01
```

Ifor.1427

ER: A polymorphic, stealth, encrypted, 1427-byte virus containing the texts ‘ANTI-VIR.DAT’, ‘CHKLIST.MS’, ‘CHKLIST.CPS’, ‘MSAV.CHK’, ‘AVP.CRC’, ‘CHKLIST.TAV’, ‘[BodyCount] version POLY-B by iFOR’, ‘SMARTCHK.CPS’ and ‘IVB.NTZ’. This template detects the virus in memory only.

```
Ifor.1427           3D89 6374 0690 9090 E918 02B8 8963 8BD8 CA02 00E8 1202 727C
```

Oksana.1843

CN: An encrypted, appending, 1843-byte virus containing the texts ‘WARNING Good Morning !!! I’m virus - TERRORIST ! Your FAT and boot and other fuckin first sectors on disk C: was D E L E T E D !!! Only I can restore it. So , don’t turn off your computer till 3 am and I will restore all shit that you keep there. Otherwise we’ll DIE together - I and YOU ! ! ! GOOD NIGHT ! ! ! (isn’t it ?) (Crazy)’, ‘HAPPY BIRTHDAY OKSANA’, ‘COMMAND.COMc:\command.com’, ‘CRAZYCRAZY ! I~m here !’ and ‘*.com’. The virus infects one file at a time. The payload tries to overwrite the first 16 sectors on the C drive, disable creation of new files and intercept key-strokes.

```
Oksana.1843         BEF7 04BF 1907 2BFE 8BCF 2E8A 0434 AA2E 8804 46E2 F5BE 0500
```

Omed.544

ER: A 544-byte appender containing the text ‘Demo’ (at the end of infected files).

```
Omed.544            B8F0 FFCD 213D 49E6 7450 B821 35CD 2131 C08E D82E 891E FA00
```

Paraguay.367

CR: A 367-byte appender containing the text ‘ ► MACONDO ◄ by Int13h ≡ PARAGUAY’. The virus resides in the Interrupt Vector Table (at address 0000h:0200h). Infected files have the byte 9Dh at offset 0003h and at the end of their code.

```
Paraguay.367        BA00 02B4 40B9 6F01 CD21 E862 FFB4 40BA 6B03 B904 00CD 21B4
```

Revenger.505

CR: A 505-byte appender containing the text ‘REVENGER RiCK BRACY’. The payload, which triggers on the 29th of each month, corrupts the CMOS data and tries to overwrite the contents of the hard disk. Infected files have the byte 00h at offset 0003h.

```
Revenger.505        B80D F0CD 213C 0D75 49B0 07E6 70E4 713C 2974 0468 0001 C3B9
```

RS

CR: Two variants of a slightly polymorphic, encrypted, appending virus. Both contain the texts ‘Divide Overflow’, ‘COMMAND.COM’ and ‘CHKDSK.EXESCANDISK.EXENDD.EXE’. The 1470-byte variant also contains the text ‘*.com’. Infected files have the word 5353h (‘SS’) at offset 0003h.

```
RS.1254             ??90 BF39 0003 FDB2 ??2E ??15 2E28 0D90 ??90 2E28 1547 E2F1
RS.1470             90BF 3700 03FD B2?? 512E 2815 2E28 0D90 ??90 2E28 1547 E2F1
```

SmallEternity.156

CER: A 156-byte overwriter containing the text ‘[SMALL ETERNITY] iS MaDe By THe GaBBeR’. Due to a bug, the virus only creates a successfully replicating copy when a target file has the byte 27h at offset 009Ch. Infected files have their date and time-stamps all set to zero.

```
SmallEternity.156   B99C 00B4 3080 C410 BA00 01CD 2199 33C9 33D2 B801 57CD 21B4
```

Suit.1167

CR: An encrypted, 1167-byte prepender containing the text ‘(c) Red Hacker’. Infected files start with the word 03EBh. The system reboots if a loaded program contains the text ‘Marek Sell’ (author of the Polish anti-virus program MkS_Vir). This template may be used to detect the virus in memory only.

```
Suit.1167            BA8F 05B4 40CD 2172 9BA1 0301 3D30 F277 933D 8F04 728E 0E07
```

VRN.2276

CER: A polymorphic, stealth, encrypted, 2276-byte appender containing the texts ‘CHKLIST.TAV’, ‘ /m /h’, ‘draziw@usa.net’, ‘CHKLIST.MS’, ‘SMARTCHK.CPS’, ‘->#ThE_WiZArD’, ‘AVP.CRC’, ‘IVB.NTZ’, ‘!! S70p K1ll1n 0uR B4byS 0r w3 w1ll d3s7r0y y0ur d474 4g41n !!’, and ‘ÄÄ   VrN vIrUs coded by ThE_WiZArD in Spain (1998) ÄÄ   !! DEDICATED TO VeRoNica !! The injustice and ignorance can only end by force If we must end this ourselvers, we will stop at nothing This is one Strike, in what will soon become MANY You may stop this individual, but you can not stop us all...’. The second template should be used for in-memory detection only.

```
VRN.2276            ??26 8035 ??40 ???? EB04 90EB 0490 A4EB FA3B C372 ECC3
VRN.2276            3DBB 3075 04B9 B0BE CF93 80FF 1174 2680 FF12 7421 80FF 4E74
```

Zwickau.505

ER: An appending, 505-byte virus containing the texts ‘*.exe’ and ‘GRUESSE AUS ZWICKAU’. Infected files have the word 0002h at offset 0014h (initial IP).

```
Zwickau.505         B800 42CD 21B4 4033 D2B9 F901 CD21 5A59 B801 57CD 21B4 3ECD
```

FEATURE

Between East and West

Costin Raiu (craiu@gecad.ro)
GeCAD srl, Romania

Romania is sandwiched between Bulgaria and Russia, two of the greatest computer virus scenes in Europe. You might expect the same situation over here – well, yes and no…

In the Beginning

Before the Great Revolution replaced the Communist Party with a democratic government, there were practically no PCs, and hence no computer viruses here. However, in 1989 foreign companies started to invest in Romania, bringing with them computers as part of their office equipment. Within a year, the first Romanian virus, Jos.1000, was discovered. This virus contained a political message – when someone typed ‘jos’, the virus’ Int 09h routine automatically inserted ‘Iliescu’, the name of Romania’s President at the time. (‘Jos’ means ‘down’ – the complete message translates as ‘Down with Iliescu’.) Some anti-virus products still detect this virus as Jabberwocky, or Jabb.1000 after the following string:

```
JABBERWOCKY (), the first Romanian Political
Virussian
```

Although Jos.1000 was only able to infect COM files, the second Romanian virus Alexander.1843 (named from its encrypted string ‘Alexander-Constanta, Romania’), was also able to infect EXEs. This virus contained code similar to that found in Dark Avenger’s Eddie viruses. Alexander’s author took some routines – for example, the memory residency routine – from the Bulgarian virus, and added his own payload. Incidentally, Constanta is a Romanian city and is suspected to be the author’s home town. Today, there are three versions of this virus; the standard 1843-byte version, and two others of 1951 and 2104 bytes each.

The first Romanian semi-stealth virus – BadSectors.3428 – appeared next. It was the summer of 1992 when I first saw it. I spent about five hours analysing it, and wrote a short detector/cleaner. This virus was extremely contagious, infecting files by hooking an impressive number of DOS functions. For example, with the virus resident, a simple DIR command could result in the infection of all the executable files in either the current directory, or the directory targeted by the command. Combined with its limited stealth abilities, this explains why the number of computers affected was so high. The author of the BadSectors virus also created other versions. While the 3428-byte variant carried the ‘BadSectors 1.2’ string, three others, of 3150, 3422 and 3626 bytes (versions 1.0, 1.1 and 1.3 respectively), were consequently found. These were not as successful as the 3428-byte version.

Modified versions of Mannequin.778 and Jinx.846, which none of the contemporary anti-virus products could detect, were the first viruses many Romanians saw. Soon afterwards, there were reports of the first boot sector viruses. 6 March 1992 was a day to remember for some of us. Michelangelo received major attention from the Romanian media – warnings were broadcast on the national TV channel, and many companies chose not to use their computers on that day. I can happily announce that there have been no Michelangelo reports this year in Romania, and we do not consider it to be an issue any more.

Back in the ‘bad old days’, Michelangelo was soon followed by Parity_Boot (both A and B variants), Ripper and Stoned in 1993, along with Hi.460, all of which were the most reported viruses at that time. The Mr family, with its large number of members, was also very common, but no match for Hi. Actually, the latter was so common that a handful of other viruses based on Hi’s code were found in the wild in 1993 and in the years to follow.

The Polymorphics

The first Romanian polymorph was discovered in 1994. Prodigy.1200, (known as UU.1200 by several anti-virus products), used simple polymorphic techniques to avoid detection by conventional string scanners. It was soon followed by Alia.1023, another polymorphic virus to be actively reported in the wild at that time. Alia.1023 was also the first Romanian virus to include anti-emulation/anti-heuristics code. The variable co-processor instruction embedded in its polymorphic decryptor, while not a problem for today’s code emulators, proved to be a very powerful defence. A 1300-byte version is also known, but did not have the same success as the 1023-byte version, most probably because of its numerous bugs.

While Prodigy and Alia used only simple polymorphism, the much larger Dumb.4722, discovered at the end of 1995, was almost impossible to detect using signatures, wildcards or crypto-analysis. Dumb.4722 was uploaded to several BBS systems (very popular in those days) in the form of a dropper claiming to be a graphics viewing utility. Its name, after the string ‘I dunno what you think about me but i am NOT dumb !’, was also the author’s nickname. In a recent interview with a local IT newspaper, he accused Romanian anti-virus developers of making money from his work and claimed to have stopped writing viruses.

The next polymorphic virus was Breath.3457. This one is a little bit more complicated than Dumb.4722, and was strongly promoted by its author via BBS using various shareware programs as droppers. Another Romanian polymorph is Calu.2429, which includes some anti-emulation code and tricks. It was discovered in mid-1996, and so far it is the latest Romanian polymorph. There are some (unconfirmed) reports of a new advanced polymorphic virus floating around, but it might just be a story – hopefully, the number here will remain low.

The ItW Viruses

There are constantly reports of numerous viruses discovered in the wild, including Leonard – of which both 1194 and 1176-byte variants are quite common right now. One_Half.3544 is still one of the most widespread viruses in Romania and we usually receive several infected (or damaged) hard-disks each month. Some months ago, we heard a few reports of infections by the 3544.G strain, which is not very different from the base version. Unsnared.814 (see *VB*, November 1996, p.10) was also one of the most reported viruses, but now that most anti-virus products detect and clean it, this is not a problem any more. Other common viruses are Burglar.1150.A, DS.3783 (see *VB*, March 1997, p.8), Pieck.4444 and the recent Romanian viruses Teapa.1609, Scapny.795 and Equinox.855.

You may know that Romania is probably the only country which has a local WildList. Gabriel Pislaru, a member of the International WildList Organization started the RoWildList in the summer of 1996. In early 1997, he handed over maintenance of the list to a fellow researcher, and it is still active today. Relevant information can be downloaded from ftp://main.gecad.ro/pub/RoWildList/. Recently, the RoWildList was used as a test-base in an anti-virus comparative review supported by the Romanian publication PC-Report.

In Romania today, the leading viruses are not COM, EXE or Macro infectors, but boot sector viruses. RP, a prolific Romanian virus author, wrote about 20 boot viruses, including the so-called RP family. RP.Dec_17th, or more precisely, its payload, is constantly reported every year. RP created the less successful RP.Bugs, RP.Remember and RP.New_Gen viruses, which are also reported in the wild.

RP also wrote Dodgy, one of the few boot viruses to replicate under *Windows 95*, using the same method as Hare to intercept floppy accesses. With more than four sub-versions, Dodgy is still reported in Romania, and world-wide. Dodgy is also one of the few Romanian retro viruses, as it contains specific activation routines targeted against our anti-virus program, *RAV*. Dodgy's payload, which is pretty effective, still brings three or four damaged disks a month into our data-recovery division.

Recently, a local IT publication distributed a Dodgy dropper in the form of a program called RENEDEMO, on one of its CDs. An apology was published in the next issue, along with instructions on how to remove the virus from infected systems. This was probably the magazine's second mistake, as most of their readers lacked the computer skills to accomplish (or understand) the removal instructions. Moreover, the fact that the CD had been scanned by a well-known (not Romanian) anti-virus package helped neither users (who must have lost faith in that program) or the editors, who eventually decided to stop including free software sent by their readers on their CDs.

Another common boot sector virus here is Multi_Ani, suspected to be of Romanian origin ('Multi ani' is Romanian for 'Happy New Year'). We very often see computers infected by both Multi_Ani and an RP variant. So far, we have no reports of Win32 viruses, or viruses able to infect PE files, such as the Win95.Anxiety family, known to be in the wild in Russia and the Czech Republic.

Macro Viruses

We first saw macro viruses in Romania late in 1995 when an IT publication printed the source code for Concept. This resulted in five or six variants, depending on the number of tabs or spaces used by the person who typed in the source. Four years later, we are still getting reports of Concept infections, but not as many as two years ago. Nowadays, NPad and CAP lead the field, but the number of *Excel 97* reports (mostly XM97/Laroux) is also showing a small increase. So far, no macro viruses appear to have been written in Romania.

In the Future

Now the virus scene is calm, compared to a few years ago. Boot viruses are still a problem, and Macros (especially *Office 97* macro viruses) are showing a notable increase over the past few months. In Romania, there are three major anti-virus products; *RAV*, developed by *GeCAD*, *AVX*, developed by *SoftWin* and *SUMI Software Developments' AspVirin*. Foreign products are available, but do not share the same success, probably due to established markets.

In Romania, there are no laws against virus writing, and the copyright law is only a few months old. Worse still, virus-writing discussion lists are prevalent. Virus-l@pcnet.ro is a common place for meeting virus writers like Lord Julus, author of a recent polymorphism-oriented article in the virus e-zine 29A#2. His polymorphic engine, called Lord's Multiple Opcode Fantasies, is probably fantasy – I have not seen any viruses based on it yet. The author of the RP viruses is well-known as he signs his sources with his real name and then distributes them amongst his friends. RP came to our stand at several computer fairs, bringing some of his new viruses to prove that known anti-virus software cannot detect them! As most virus authors use pirated anti-virus software to make their viruses undetectable, heuristics are probably the only defence we have against them.

As for the future, I expect a fall in the number of boot sector infection reports, along with a small decrease in reports of macros due to the advanced macro virus detection engines used in today's anti-virus software. If nothing new emerges (like a new kind of virus, or a new, highly insecure virus platform from a large software vendor) the virus situation in Romania should be easy to keep under control for many years.

OPINION

The 'W95CC' Problem

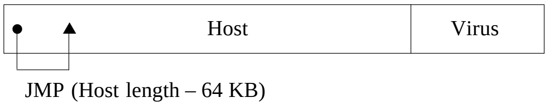
Peter Morley
Dr Solomon's Software

Recently, we have received a spate of infected *Windows 95* COMMAND.COM files from the field. The customers who send them are unhappy, because we often fail to detect viruses in them, or if we do, the files are still not repaired. The viruses in question tend to be old favourites like Junkie, Green_Caterpillar, and Dutch_Tiny. The inevitable comment from customers – 'We thought you had this virus or that virus well under control' – is not entirely inappropriate. Here, I attempt to explain both the COMMAND.COM file problem, and the actions we take to deal with it.

Technically Speaking

The *Windows 95* COMMAND.COM is not a COM file at all. It is, in fact, a normal 93 KB EXE file, which *Microsoft* sees fit to call COM for backward compatibility reasons. Normal COM files are limited to 65,280 bytes because all their code plus the 100h-byte PSP must fit in a 64 KB segment. Several viruses, including the three above, select files for infection solely on the basis of file extensions, thus *Windows 95* COMMAND.COM files are fair game.

A *Windows 95* COMMAND.COM file is infected in the following way. The virus records the victim's first few bytes in its code (these bytes form our repair data), before jumping to the end of the file and appending its code. Then it overwrites the first three bytes of the host with a near jump that would usually point to the virus code. However, this initial jump is miscalculated – it is 65,536 bytes too short. A misinfected COMMAND.COM file, which will neither run nor infect, is the result. Inevitably, the PC becomes the subject of a technical support call.



JMP (Host length – 64 KB)

The classic anti-virus view of this, which I cannot condone, is as follows. Since the misinfected COMMAND.COM file cannot run or infect anything else, it is not a virus, and thus not necessarily an 'anti-virus issue'. The problem becomes apparent very quickly as the machine itself will not run, or boot, and therefore further damage cannot occur. The solution is to replace the damaged COMMAND.COM file. Everyone has a copy of it, after all.

The Customers' View

Every customer I talk to rejects this view. They say 'I expect *Findvirus* both to detect and repair my misinfected COMMAND.COM file. After that, I can boot clean, run *Findvirus* in repair mode, and my problem is over in less than five minutes. If it fails to do this, I will still boot clean and run *Findvirus*, but the problem has just begun; I find my machine will still not boot, and call [internal] tech support. When they arrive, I explain the problem, but they may not know the solution. At worst, they may take my machine away and experiment on it – I may lose data. At best, if they replace my COMMAND.COM file first thing, I will have lost about an hour, instead of five minutes.'

My View

I accept the customer view, and have made serious efforts to provide detection and repair. One fact I have kept from you is that providing detection and repair for a specific virus is easy. After that incorrect initial jump, if you skip another 65,536 bytes, you are in the correct position in the file to do the detection and repair. The problem is knowing which viruses to do it for.

It is out of the question to review each known virus to check whether it misinfects. Instead, field samples are processed immediately, with any modifications extended to all other variants of that virus. The client is sent an extra driver (in case the problem recurs) and told which release of *Findvirus* will incorporate the fix.

We have added a *Windows 95* COMMAND.COM file to the set of bait files we use, with every new virus we process. So, if the problem occurs, we fix it not only for the new sample, but for all other variants of that virus. There will be a period during which we receive more from the field. Please keep them coming, rather than treat the problem as a trivial one-off, and not bother.

Virus Construction Packages

All the above raises the question of what to do about old viruses produced using virus construction kits. This is particularly relevant in the US, where several of these packages were written, and are still widely used by virus creators ('authors' is not the right word!). Well… the news is good. As far as I can see, viruses produced with IVP, (perhaps the most commonly used kit) infect *Windows 95* COMMAND.COM files as normal EXE files, and the usual repairs work just fine.

MPC is a mixture. Most *Windows 95* COMMAND.COM infections are normal EXE infections, and we repair them. However, one morph (MPCa) is a COM file misinfector. We have fixed it for all the samples we have, but we expect more will come. The other common construction kit – VCL – does misinfect as I have outlined (or at least the variants I have tested do). As a parting shot, I propose that the people who test anti-virus products for magazine review articles should consider this subject.

VIRUS ANALYSIS 1

Crossing the Office

Andrew Krukov
Kaspersky Lab

'This could easily be done with *Excel* and *Word* but I don't have *Excel* loaded so it didn't happen that way.'
Author of the Cross virus

Early in their development, computer viruses evolved 'vertically'. When the first viruses appeared in the late 1980s, different viruses infected different objects (disk boot sectors *or* files). The next generation – memory resident, stealth and polymorphic viruses – infected in more complex and clever ways, but they did not target other objects. Parasitic viruses infected only files, boot viruses infected only disk boot sectors.

In early 1990, the first multi-partite virus appeared and surprised anti-virus researchers. Anthrax infected DOS files and hard drive MBRs. Thus, viruses started to cross 'platforms' and their evolution took a new direction. They could be seen to progress 'horizontally' – infecting one type of object, jumping to another, and often being able to infect the first kind of object again.

When *Windows* viruses first made an appearance, one of them was capable of infecting both DOS and NewExe executable files. Then macro viruses came along, and several of them were able to drop DOS or *Windows* parasitic viruses. In a short time, the Anarchy multi-partite virus (see *VB*, October 1997, p.6) was able to infect *Word* documents from a DOS environment, and vice versa.

We are now observing yet another phase of virus evolution. The 'vertical' progression of macro viruses, which were first discovered in 1995, has now become a 'horizontal' development. At first, stealth and polymorphic macro viruses appeared and new infection mechanisms have been 'tested and approved'. Nowadays, macro viruses are starting to cross platforms, and the first known multi-partite macro virus has been caught in the waves of the Internet.

This extremely large infector (530 KB of Basic source code) is capable of infecting *Word 8* (*Office 97*) documents from an *Access 97* environment, as well as dropping an *Access 97* virus from infected *Word 97* documents. Fortunately, this virus is not able to spread itself more than once because of bugs. Despite this, the very concept of 'multi-office' macro viruses is quite dangerous.

Following the release of the first *Access* virus (AccessiV, see *Virus Bulletin,* April 1998, p.15), it seems many virus writers have turned their inflamed brains to this new virus writing field. It is very probable that now, after the discovery of the first multi-partite macro virus, they will switch to new ideas. In a short time we expect to see a 'multi-office' virus that jumps from one *Office* application to another and back – in effect, 'crossing the Office'.

Matreshka Set

Recently, *Kaspersky Lab* received a package containing two infected files – an *Access 97* database and a *Word 97* document. The virus code in each file is able to replicate under its native application, so, in one sense, we have two different viruses in the same package – each of them infects native objects (documents or databases) without any problems. A feature common to both viruses is the ability to infect files from other *Office* applications. Thus, the *Access* virus drops a *Word* macro virus, while the *Word* virus drops an *Access*-infected database.

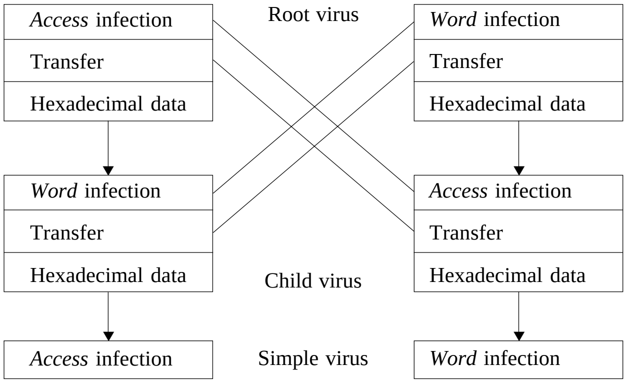
Both viruses share another commonality – a similar three-part structure. The first part is a native infection routine, the second a routine that transfers the virus to another *Office* application. The third part contains hexadecimal data that is converted into an infected file when the second part infects another *Office* application.

The Cross virus' hexadecimal data is in standard macro virus form – it is prepared to be converted by DOS DEBUG into a binary data file. This method involves the virus writing the data to a temporary file, creating a DOS batch file that runs DEBUG to convert the data into a binary disk file, and then deleting all temporary files. It is necessary to note that binary data dropped by the virus is in CAB (MS Cabinet) format – it is a compressed file that can be unpacked by the *Windows* Extract utility.

When we analysed the hexadecimal data in both infected files, we found that it contained two other viruses – for *Word* and *Access*. They are also capable of spreading themselves under their native applications and dropping infected objects to another application. The replication and transfer routines are identical, but the hexadecimal data is not the same as the parent virus! In the next layer of data we found another pair of *Word* and *Access* viruses that have neither transfer routines nor hexadecimal data. These viruses can only spread under their native applications.

So, we have a 'matreshka' (Russian doll) of viruses – each of the larger ones has another inside. Opening this package matreshka by matreshka, we found three layers. The first, or root, virus contains a dropper of the second-level or child virus, the child contains a third – a simple *Access* or *Word* virus that is not able to spread across applications.

The *Access* root virus differs from the *Access* child virus only in its hexadecimal data section (similarly, the *Word* root virus differs from *Word* child virus in this way). Both infection and transfer routines are the same command for command in both the *Access* and the *Word* pairs. So, the original ‘matreshka’ has three levels of cross-encapsulation with the same routines:



Going backwards from the simple *Word* and *Access* viruses to the root ones, we found that both pairs of *Access/Word* infection routines are very similar. The only difference is that the simple viruses do not have the code for, and calls to, the transfer routines, whereas both root and child viruses do. So, the root viruses seem like the result of a two-step extension of the simple viruses to handle another application. The effect of such an extension is the virus’ size – the source code of the *Access* virus is 370 KB, while the source of *Word* virus is nearer 160 KB. This places Cross amongst the largest macro viruses.

Spreading the Word

The infection routines in both the *Access* and *Word* samples use techniques common to already known viruses. The *Word* virus replicates by using the Import/Export facility in *Office 97* VBA. The virus saves its source code to a temporary file on the disk and then imports it to all opened documents. Although not new, this method is quite unusual and currently limited to only a few viruses (for example, W97M/AntiSR1).

The virus consists of one module named ‘X’, containing several subroutines: AutoOpen, AutoClose, AutoExit, AutoExec, FindAx, MakeBat, DropKey, DropDetox, CheckKey and Info. The AutoOpen macro contains the infection routine. Initially, this removes Tools/Macro, and Tools/Templates and Add-Ins from the *Word* menus, providing the virus with some basic stealth abilities. The routine then exports its source code to the C:\X.VIC file. Finally, the virus checks for the executing environment (document or normal template) and infects the appropriate object by importing the source code into it.

There is only one difference between the AutoOpen routine of the root and child forms and that of the ‘simple’ virus: in the root and child forms, AutoOpen calls the CheckKey subroutine that infects *Access*, if installed. The CheckKey routine tries to find files matching the C:\*.YZV mask. If such files are not found, the routine calls the FindAx routine (so, the presence of the *.YZV file means the system is already infected).

The FindAx routine tries to find the *Microsoft Access* application in C:\Program Files\Microsoft Office\Office. If *Access* is found, the DropKey routine is called, which creates a C:\AX.YZV file, then calls DropDetox (creating a script for DOS DEBUG) and MakeBat (creating a DOS batch file to unpack the DATA.MDB file then executing *Access* with DATA.MDB as a parameter). Then the FindAx routine executes the batch file in a hidden window. The rest of the auto macros just call AutoOpen.

The Info routine contains the following text:

```
Cross.Poppy Word Component
—[Cross is a blend of SexR-1 and Detox]—
by VicodinES / Sin Code IV (anagram)
```

Infected Access

In order to infect *Access* databases, the virus uses the same method as that in known *Access* macro viruses (AccessiV, Detox). The function named TheDetoxUnit finds MDB files in *Access’* current directory and ‘injects’ virus code into them with the TransferDatabase command. Interestingly, the virus disables Tools/Options... menu items and sets several properties of the infected database by calling the SetStartupProperties routine. These are simple stealth mechanisms, as employed by the Detox *Access* virus.

In a loose parallel with the *Word* part of the virus, the root and child forms contain and call a CheckKey routine, which is not present in the ‘simple’ form of the virus. This routine checks for C:\*.YZV files. If unsuccessful, CheckKey calls the FindWd routine that tries to find the *Word* application in C:\Program Files\Microsoft Office\Office. If the application is found, the FindWd routine calls the DropKey routine, creating the flag file C:\AX.YZV, then calls DropSexr1 to create a script file for DOS DEBUG. Finally, it calls MakeBat to create a DOS batch file and executes it.

Accessing the Word

In order to spread to *Word* from *Access*, the virus creates an infected DATA.DOT file in the *Word* Startup directory. It does this by writing a DEBUG script (from its hexadecimal data section), piping this through DEBUG, and unpacking the resulting CAB file with the EXTRACT utility. Once dropped in this convoluted manner DATA.DOT is copied to the C:\PROGRA~1\MICROS~1\OFFICE\STARTUP and C:\PROGRA~1\MICROS~2\OFFICE\STARTUP folders. If they do not exist, the virus fails to spread to *Word*.

If one of these copy operations succeeds, when *Word* starts it loads templates, including the infected DATA.DOT file, from its startup directory, and the virus takes control. Fortunately, the virus has a bug in the hexadecimal data and cannot infect *Word*. (The child *Access* virus does not have this limitation, and can spread from *Access* to *Word*.)

Further, the root *Access* virus is not able to replicate under some system conditions. When *Access* executes the virus macros it sometimes displays an error message about low memory. This error appeared on PCs with 24 MB of system memory installed, but there was no such error when replicating the virus on PCs with 64 MB of RAM.

From Word to Access

The virus uses a method analogous to that just described when infecting *Access* from *Word*. It creates the temporary DATA.COM file with DOS DEBUG, writes its hexadecimal data there, and unpacks it with the Extract utility to the infected DATA.MDB *Access* database. The virus then executes *Access* with the START command, passing the infected DATA.MDB file as a parameter. As a result, the virus takes control and infects other *Access* databases.

Conclusion

The evolution of computer viruses continues. DOS is dying, and DOS viruses will die with it, but a great number of *Windows* and macro viruses are replacing them, often repeating the evolutionary history of 'good old' DOS viruses. They occupy new 'ecological niches' and are spreading from one platform to another – from *Access* to *Word* and back in Cross' case.

We are probably fortunate that some of the more sophisticated cross-*Office* infection mechanisms available are not used in Cross. It can only be a matter of time, however, before we see viruses that utilize some of these options, breaking the shackles (and limitations) of DEBUG and spawning batch files to achieve their trickery.

Another problem is that few anti-virus vendors yet support *Access* database formats. If this kind of virus had its bugs fixed and appeared in the wild, it may be a real shock to the PC world. Let's be ready – ready for a *Word-Excel-Access-Windows* stealth and polymorphic virus that also infects MBRs and drops itself on CDs, if a CD-writer is installed…

Cross

Aliases:	None known.
Type:	*Microsoft Word 97* and *Access 97* macro infector which 'cross-infects'.
Self-recognition in files:	*Word* documents containing a module named X are assumed infected.
Intercepts:	*Word*: AutoClose, AutoExec, AutoExit and AutoOpen macros. *Access*: Autoexec macro.
Payload:	None – see text for bugs.
Removal:	In a clean *Word* environment, delete modules from Organizer. In *Access*, reset the Toolbars on the Toolbar tab of View/Toolbars/Customize, then follow the procedure for AccessiV (see *Virus Bulletin*, April 1998, p.15).

VIRUS ANALYSIS 2

HPS

Péter Ször
Data Fellows

The first *Windows 95* virus with oligomorphic capabilities was Memorial, which appeared in 1997 (see *VB*, September 1997, p.6). It caused infections in the wild and we suspected that the next step would be a fully polymorphic virus for *Windows 95*. Virus writers had to extend their knowledge significantly to reach that level of coding under *Windows 95*, but we did not have long to wait.

It is not surprising that the person to create such a virus comes from a strong DOS virus-writing background, specifically in polymorphic/multi-partite viruses. GriYo is a notorious member of the 29A virus-writing group and the author of complex viruses like Implant. As part of 29A, he has others around him, such as the author of Cabanas (see *VB*, November 1997, p.10), to help his creations work efficiently in *Windows* environments.

The pioneer of 32-bit polymorphic *Windows* viruses is called HPS (or Hantavirus Pulmonary Syndrome). The virus has been developed specifically for *Windows 98*, but is compatible with *Windows 95*. Fortunately, it does not work under *Windows NT*, because it is built on top of undocumented *Windows 9x* functionalities. HPS.5124 is a slow polymorphic, Portable Executable (PE) infector with directory stealth. Not only does it employ a new method of hooking the *Windows 9x* file system, it also has retro virus features. Its polymorphic engine is quite advanced, but worse news is that it is very stable. Despite making some extreme hacks in the heart of *Windows 9x*, the OS seems to accept the virus as part of the system. HPS is one of the first *Windows 9x* viruses with a graphical activation routine.

Initialization

When an infected PE program is executed, HPS' polymorphic decryptor takes control. The entry-point of the PE header points to the last section of the executable where the virus code is placed. The decryptor is at the end of the virus, in the last few hundred bytes of the infected program. The encrypted body precedes the decryptor with a fixed size of 5124 bytes. However, the overall size (including the decryptor) is variable, so HPS saves its size at the end of the infected program for use by its directory stealth routine.

The first byte of the virus body takes control when decryption is complete. First, the virus calculates the real entry point to the host program and saves its value to the stack. HPS' novelty is illustrated in this initial routine, which attempts to determine the address in memory of the already loaded KERNEL32.DLL in memory. Its purpose is similar to that of the Win32 GetModuleHandleA function that returns the base address of a DLL in memory. These acrobatics allow the virus to access any exported APIs from there, without thunking through itself in the host's import section. The search routine is protected by Structured Exception Handling (SEH) as it would certainly cause General Protection faults otherwise. SEH saves the function from application level debugging.

It seems that KERNEL32.DLL is unlike any other DLL, since it is presented for all processes (under Windows NT, too) even if the actual program does not have imports for it. Thus, the virus can search for it in the usual shared memory address space. Since this is different from one system to another, the virus tries different addresses, scanning the memory backwards in 64KB steps for the 'MZ' marker representative of a loaded DLL (EXE) program. When the marker is detected, the virus looks for the 'PE' identifier. If found, HPS checks that the name of the module in the export table is KERNEL32.DLL. The virus uses some complex calculations for these signatures, and thus was tricky to analyse at first glance.

If the base of KERNEL32.DLL is not detected, HPS simply executes the host program. Otherwise, it checks for the VxDCall address at the beginning of the KERNEL32.DLL export table. A VxDCall is an undocumented *Windows 9x* API. The entries of VxDCalls (there are several of them to accept different parameters from the caller) were documented in the first beta release of *Windows 95*.

Microsoft decided to remove the names of these entries from the exported function table since Andrew Schulman used them in 'Unauthorized *Windows 95*' to show how KERNEL32.DLL calls VWIN32_Int21Dispatch services. Moreover, *Microsoft* added special code to the *Windows 95* GetProcAddress API to disallow imports by ordinal values. This is part of everyday business, the usual 'Where do you HAVE to go today?' ideology. Obviously, *Microsoft* did not want to allow developers the use of VxDCalls. The rationale behind this is probably the prevention of solutions incompatible with *Windows NT*.

Since the virus cannot use GetProcAddress to obtain the VxDCall address, it picks it up from the export table of the loaded KERNEL32.DLL. Then it makes its 'Are you there?' call (a VxDCall) – VWIN32_Int21Dispatch get date services with ESI=48505321h ('HPS!') and EDI=5453523Fh ('TSR?'). If, on return, ESI=59455321h ('YES!') the virus assumes it is already resident and executes the host program. Otherwise, it checks if it is Saturday and saves a flag to use later.

The use of a VxDCall is not new. Xine used one similarly in its infection routines, but HPS is the first virus to hook a VxDCall. The implementation of the hook requires a good understanding of *Windows 9x* system internals. Since a VxDCall is used by all applications through system DLLs (like KERNEL32.DLL), the implemented hook cannot be placed anywhere but in the shared memory area for all processes under *Windows 95*.

In order to be able to place itself here, the virus uses undocumented Win32 VxD services provided by VMM – PageReserve and PageCommit – again using a VxDCall. Thus, the virus allocates four pages of shared memory. If the start address of the allocated pages points lower than the usual shared memory range, HPS uses PageFree to free them. This is because the hook would certainly hang the system if the memory allocation were not able to reserve pages from the shared memory area.

Next, HPS scans 256 bytes from the entry point of the VxDCall in KERNEL32.DLL for the signature 2EFF1Dh. These are the bytes followed by an address that can be patched to the virus' own VxDCall entry point. (Basically, the patching method is very similar to the one Matt Pietrek used in one of his sample programs in his excellent book 'Windows 95 System Programming Secrets'.) How is that patch possible at all? How can a Ring 3 program overwrite some bytes in a shared system DLL? The answer is that this area is write-enabled. Thus, the patch works without even using the WriteProcessMemory debug API.

Before the actual hook, the virus calls its polymorphic decryptor generator. Since polymorphism happens only at installation, the virus is a slow polymorphic. HPS' body is copied to the beginning of the allocated shared memory area and the decryptor is created after that. Finally, the virus hooks the VxDCall and executes the host. From then on, the virus actively monitors VWIN32_Int21Dispatch services without having any specific VxD part. This simplifies the virus structure significantly.

Infection/Own VxDCall Handler.

The new VxDCall handler monitors all the important, *Windows 95* long filename functions (7143h, 714Eh, 714Fh, 7156h, 716Ch, 71A8h) coming in as an Int21_Dispatch service code (2A0010h) and all file open requests for read or write. The virus also monitors the get date function for its 'Are you there?' call.

The virus carefully saves the host program's original attributes and its date-stamp, resetting them after infection. In EXE, SCR, BMP and SYS files, the virus looks for the extension of the accessed file. The above files are checked for their internal structure. If the first bytes are 'MZ' and the 'PE' mark is in place, the virus tries to infect the host program (if it has 386-compatible code). If the file starts with 'BM', the virus checks if the actual image is a non-compressed bitmap file. If so, the virus performs its activation routine.

Files whose sizes can be divided exactly by 101 are assumed to be infected. This is the same self-check that Cabanas uses. The 29A group may have decided to use this simple trick to avoid double infections by their different viruses. During the infection process, the virus uses only Int21_Dispatch service routines. It modifies the entry-point of the PE header to point to the end of the program. It also alters the last section header to fit its own code, and changes the flags in order to execute code in that section. Then it calls its generic encryption function and encrypts its image with this routine. After that, it writes itself to the end of the host program with the polymorphic decryptor created at installation and some additional bytes at the end to make its size exactly divisible by 101. HPS saves the entire virus size as the last four bytes of the file for later use by its directory stealth handler.

When infection is complete, HPS looks for various anti-virus checksum files such as ANTI-VIR.DAT, AVP.CRC, CHKLIST.MS, IVB.NTZ. If these are in the same directory as the host program, the virus deletes the file, after taking the precaution to clear the file's attribute. Finally, it gives control to the original VxDCall.

Directory Stealth Handler

HPS is a directory stealth virus. It does not hide file size changes from the DIR command, as it only monitors 714Eh, 714Fh LFN FindFirst/FindNext functions.

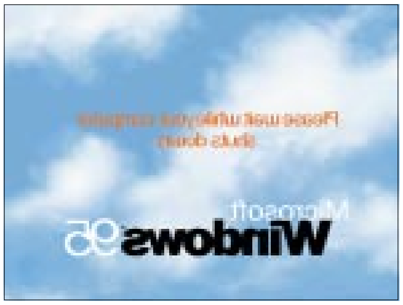
Implementation of the stealth handler is unique. On the fly, the virus patches the return address of FindFirst/FindNext functions on the stack to its own handler. This handler checks whether the file size is exactly divisible by 101. If so, the virus opens the program with an extended open LFN function. It reads the virus size from the last four bytes of the infected program and subtracts this value from the value returned by FindFirst/FindNext. This way the virus is able to hide its file size changes from most applications, despite having a variable infective length.

Polymorphic Engine

HPS has a polymorphic engine as powerful and advanced as that of Implant. It supports subroutines (using CALLs and RETs) and conditional jumps with non-zero displacement. The polymorphic engine comprises about half the virus code. There are blocks of random bytes inserted in the decryptor's generated code chain. The full decryptor is built up during the first initialization phase which makes the virus a slow polymorphic. This means that anti-virus vendors cannot test their scanner's detection rate against this virus efficiently, because the infected PC has to be rebooted in order to create a new decryptor. The decryptor consists of 386-specific instructions. The image is encrypted and decrypted by different methods including XOR, INC, DEC, NOT, ADD and SUB instructions with 8-, 16- or 32-bit keys. This reduces the range of detection shortcuts rather drastically. I am sad to say that the polymorphic engine part is well written, just like the rest of the virus. It was certainly not created by a beginner.

Activation Routine

Since the virus only checks the date during initialization, it will only activate when the infected PC is booted on a Saturday (or the virus is executed on a Saturday for the first time). If a non-compressed bitmap file has been opened, the virus allocates enough memory for the image and horizontally flips the picture. HPS patches the value DEADBABEh into the end of the bitmap header area to avoid flipping the same image again. Since non-compressed bitmap files are frequently used by *Windows 95*, this causes all kinds of weird effects.



Conclusion

I had not finished writing this article when another polymorphic virus called Win95/Marburg, also the work of GriYo, was discovered. Marburg has a similar infection method to Cabanas, but it only works under *Windows 95* due to a small bug in the replication code. It would be easy to fix this bug, thus I suspect we will see Win32 (*NT*-compatible) polymorphic viruses in the near future.

We also anticipate the emergence of a polymorphic mutation engine library, callable by Win32 viruses in a few months' time. Moreover, we will have to face more and more weird graphical effects under 32-bit *Windows* systems caused by *Windows 95* and Win32 viruses. It seems virus writing is going back to its roots. While Cascade exploited the possibilities of DOS by way of graphics, *Windows* viruses have endless resources for hair-raising activation routines. These will most certainly challenge us in 1998.

HPS

Aliases: Win95/HPS, Hanta.

Type: *Windows 95/98* PE infector, hooks VxDCall, directory stealth.

Self-recognition in Files:
Files whose size is exactly divisible by 101 are assumed infected.

Self-recognition in Memory:
VxDCall (VWIN32_Int21Dispatch get date) on call ESI=48505321h, EDI=5453523Fh returns ESI=59455321h if the virus is active.

Hex Pattern in PE files:
Not possible.

Intercepts: VxDCall, VWIN32_Int21Dispatch services.

Payload: Flips the image in non-compressed BMP files horizontally on Saturdays.

Removal: Under clean system conditions, restore infected files from backups or replace with originals.

CONFERENCE REPORT

IVPC’98: Taking the Mickey?

Richard Ford
IBM

One of the most enjoyable aspects of the anti-virus world is the chance for researchers to gather together and share battle stories. However, given the fast pace of development of new viruses (and recently, it seems, other malware threats), such opportunities usually only present themselves at conferences. As the industry has changed, such conferences have come and gone, but the two most enduring are the *Virus Bulletin Conference* (organized with military precision) and those virus conferences hosted by the *International Computer Security Association* (*ICSA* – formerly the *NCSA*).

As the *ICSA* has undergone rapid growth, these *International Virus Prevention Conferences* (*IVPC*) have in the past been somewhat hit or miss affairs, ranging from ‘excellent’ to ‘not quite up to par’, and passing all points in between. Thus, as one never quite knows what one is going to get, it was with interest that I began to travel from the teeming conurbation of North Eastern USA to the balmy [*Pun intended? Ed.*] oasis of Orlando, the setting for the rather prosaically titled *IVPC’98: Protecting the Workplace of the Future*.

Location, Location, Location

IVPC’98 was scheduled to run from Tuesday 28 April to Wednesday 29 April, and this year was being merged (or in the parlance of the conference proceedings ‘co-located’) with the somewhat more prolixly titled *Remote Access: Building and Managing the Workplace of the Future* conference organized by the *Gartner Group*.

Held at the beautiful Walt Disney World Dolphin Resort, delegates had the option to register for both conferences, or just *IVPC*. As the cost of attending *IVPC* alone was a rather stiff US$845 (all this for a single stream conference), my wallet would only stretch to the cost of *IVPC*; thus, I shall make no further comment regarding the *Remote Access* part of the conference.

Given the announcement on 16 April that the *Gartner Group* had, to quote the press release, ‘made an investment of 19.9 % in *ICSA, Inc*’ I suspect that we may see several other such ‘co-locations’ in the not-too-distant future. Several fellow delegates commented on the seeming illogicality of running an anti-virus conference as one stream of a teleworking and networking conference. It was felt that combining *IVPC* with a more security-related conference would clearly seem to make a lot of sense given the small attendance at the virus portion of this conference.

What Goes Around…

Opening the first day was Dr Peter Tippett of *ICSA* fame, speaking on ‘The Cost of Computer Viruses’. As the track summary stated that ‘Dr Tippett will also address the epidemiology of the latest threat to computer security – malicious software or malware’, it was with some interest that I cruised downstairs at 8:30am (please – conference organizers – have mercy on those who travel and are operating on different time zones!).

It is not uncommon to rework presentations, and re-use them, as a lot of work usually goes into them, so I was not surprised when the opening presentation seemed very similar to a talk I heard last year from Dr Tippett. For those who have not heard the original talk, the thesis is simple and fundamentally solid: most anti-virus software works, but many people are not using it, or use it incorrectly. By creating a model of what computer viruses cost, Tippett went on to show that using anti-virus software is much more cost-effective than not using it.

While the kernel of this is solid, the model appeared to be less robust in the details. Sadly, there was no paper or even a copy of the slides included in the conference proceedings, so more directed comments are hard to make. Perhaps the main surprise was Dr Tippett’s conclusion: that scheduled scanning ‘is not worth it’ and that updating one’s anti-virus software monthly is also not terribly useful.

The next session was a presentation by Rob Stroud, from *Ontrack Data International*, which included a written paper (of the eleven talks given at *IVPC*, the conference proceedings contained only four papers and no copies of slides). Stroud painted an interesting picture, stressing the rapidly changing threats, and urged users to re-assess their policies; a system that was effective two years ago may be in serious need of an overhaul today. To quote Stroud, ‘in summary, the effective implementation of virus prevention procedures you have implemented today may not be the correct procedure for tomorrow.’

In terms of practical advice, Stroud advised that it is vital to keep software right up to date (a different conclusion from that given in the preceding talk), and to keep a level head. While viruses are a definite threat, they are far from the leading cause of data loss according to *Ontrack*’s in-house statistics – hard drive failure was much more common. ‘Keep regular backups’ he concluded – sensible advice.

The next scheduled talk was to be by Jason Khoury (*ICSA*) on legal issues. I was looking forward to this session, as Jason is one of the *ICSA’s* most interesting presenters, and the topic was a timely one. However, Jason was unable to attend the conference and so Dr David Stang was slotted in at the last minute (thus being the only person there with a legitimate excuse for not providing a paper!), presenting what seemed to be much the same material as he did at last year's *Virus Bulletin* conference.

After lunch (a lovely sit-down meal, well done *ICSA*!), another speaker with no paper or slides in the proceedings presented: Dr Steve White of *IBM Research*. With the intriguing title 'Taming the Demons of Cyberspace', White discussed the new threats that increased connectivity and 'ubiquitous e-mail' pose.

Always an animated speaker, White painted an accurate picture of the current threat, pulling in new technologies like Java and ActiveX and presenting another set of statistics for virus prevalence which were different from all others presented at the conference. White believes the virus problem will increase and that the only way for organizations to stay on top of things is to use anti-virus software that travels faster than the viruses can. He showed how the *IBM* immune system technology has been designed to move this way, although there was no demonstration like the one at the *Virus Bulletin Conference* last year.

Malware Malarkey

Keeping on the subject of new threats and the demons of cyberspace, Dr Igor Muttik of *Dr Solomon's Software* spoke about Trojans on the Internet. Muttik opened with the real date of the original Trojan problem and continued onward from this point, placing today's Trojan threat in a nice historical perspective.

He divided history effectively into what I shall call the 'Before Internet' and 'After Internet' ages. Muttik argued that the Internet has completely changed the face of the Trojan threat by supplying a number of what he refers to as 'replication mechanisms': Usenet, Spam and the WWW being prime amongst these. Thus, he reasons that in a connected world, the replicative properties of an actual virus are much less important, as widespread distribution of a pathogen can be accomplished directly.

Muttik sees a technological solution to this problem, rather than a sociological one. Education as to the risks of running 'untrusted software' and the 'correct' response to receiving such, he argues, 'is not very effective and takes ages'. While there is some truth to this, it seems unlikely (and undesirable) that technology can ever replace common sense. However, Muttik does convincingly argue that we need more robust operating systems and WWW browsers, and, of course, anti-trojan software.

This move toward promotion of anti-trojan software was echoed in several of the other talks, but most notably in those by Shimon Gruper (*eSafe Technologies*) and Igor Grebert (*Trend Micro*).

Gruper posited that a system which could 'learn' the normal behaviour of software on a machine and applied a sandbox to those files downloaded from the Internet could provide protection against 'software vandals' (aka malware). However, such an approach seems fraught with problems, similar to those which beset more traditional anti-virus behaviour blockers. Furthermore, while some companies have successfully made an effective behaviour monitor (e.g. *Norman Data Defense*) for viruses, this task is considerably easier than detecting generic malware. A virus is constrained in that it must infect something; malware has no such constraint. Thus, creating a behaviour monitor for malware on the face of it looks like a much more difficult task. Has *eSafe Technologies* succeeded in developing a technological solution that is effective and not prone to unreasonable constraints on system performance? Only time will tell… I, however, remain to be convinced.

This focus on more generic forms of malware is of some concern to me. While the threat of Trojanized software is obvious, the correct approach is far from clear. In the rush to get things to market, it is possible that the industry may both raise false expectations from users as well as provide a false sense of security. Additionally, as there is little hard data concerning the way in which Trojans are targeted outside limited populations like AOL (with its infamous, if not numerous, password stealers), it is easy to claim that a program protects against Trojans, but hard to test such a claim in a meaningful way.

From an end-user's perspective, this is clearly a dangerous combination, but perhaps 'opportunity-filled' if you are a software developer. More research is badly needed to gain a better understanding of the threat types that we expect to see, and the ways in which those threats will be realized. In turn, this should lead to a better understanding of how to address the concerns raised by non-viral malware

In Conclusion

As I sat back on my plane leaving Orlando, sipping my Coca Cola and enjoying my Rold Gold pretzels, (ex-editors of *Virus Bulletin* suffer from a noticeably lower level of stress than they did when they held such a hallowed position), I reflected on the last two days.

On a positive note, a trip to Disney is always a fascinating experience for anyone involved directly with customers: the Dolphin staff were very conscientious in their attention to detail and overall standard of care. The same was true of the workers at the Disney parks too… despite the general high level of support in the anti-virus world, there is still a lot which we as an industry could learn from watching Disney's staff! Another positive note was the strength of the conference organization – definitely an improvement upon that of recent years.

That said, this was not enough to redress the concerns I have expressed in the preceding sections, making the 1998 event a little disappointing. The *ICSA's* annual conference has the potential to be an important and thoroughly worthwhile event. Hopefully, with the infusion of the *Gartner Group's* experience in conference organization, a new lease of life will be breathed into *IVPC'99*.

PRODUCT REVIEW 1

Disknet for Windows 95

Disknet from *Reflex Magnetics* functions primarily as a data security tool – at the heart of its operation is the principle that if all incoming programs are checked for content, and the machine is virus-free, then the PC will remain virus-free indefinitely. While the main attraction is that scanning can be limited to new files – far less onerous than the traditional anti-virus system – there are, of course, potential drawbacks. Chief among these is the possibility that viruses which slip through the initial incoming file scan could also be missed by the behaviour monitor and macro interceptor.

This 'isolationist' policy relies on the product's preliminary scanning engine – a slightly modified version of the long-established *Norman ThunderByte AntiVirus*, renamed *Sherlock*. This was tested thoroughly using the standard *VB* methods, while the more security-based features were subjected to a less exhaustive examination. The additional Macro Interceptor, a *Word* add-on of sorts designed to check for Macros while *Word* is running, was also tested with a limited subset of the *Virus Bulletin* Macro test-set.

Packaging and Documentation

The review product came in a sturdy box, decorated in a very blue fashion, containing a shrinkwrapped manual, and two *Disknet* 3.5-inch floppies. Also included were a Macro Interceptor disk along with its own demonstration disk, and a *Sherlock* disk. A single sheet of installation instructions for the Macro Interceptor was more ominously joined by an erratum sheet for *Disknet* itself, about a problem with a DLL which might not work correctly under certain versions of *Windows 95*. The reassuring advice states that 'The fault only occurs occasionally and may not need to be fixed on a machine unless the problem occurs'. It did…

Diskent Floppy Disk Administrator
File
Options
Help
3½ (A:)
reflex
diskent
Select Floppy Drive to Check Diskette
Exit

The manual is a complex and potentially confusing document, though with the multitude of options available in *Disknet* this came as no real surprise. It contains a few mistakes, most notable being a list of five questions, clearly labelled as ten. The introduction turned out to be the clearest part of the documentation, initially explaining *Reflex's* argument for installing *Disknet*, and then discussing typical setup options with their pros and cons. The following pages proved less helpful than a course of educated experimentation. On-line documentation does not exist for *Disknet* in any way, shape or form, which was irritating when attempting to work out the precise effects of the many different options. The Macro Interceptor does have limited, if cumbersome, help available on-line.

Installation

Disknet is one of those increasingly rare applications which recommend the use of the control panel's 'add/remove programs' at installation. The program failed to install correctly at first, but, apparently, not due to the requirement of an updated DLL mentioned above. A reboot and another installation attempt resulted in the expected error, which vanished with the addition, in DOS mode, of the updated COMCTL32.DLL.

This file cannot be found on the *Disknet* disks for what *Reflex* claims are legal reasons. It is, however, readily available with current versions of *Internet Explorer* (*IE4*). As the latter is freely distributable, there may well be a case for *Reflex* including *IE4* with future releases. [*Although this may drastically increase the number of diskettes needed to distribute the product! Ed*].

More seriously, this raises the question of why a developer would utilize a user-interface feature for which it cannot supply a necessary component other than by requiring the user to install a particular product from yet another developer. *Microsoft* now makes a redistributable COMCTL32 updater available, and its installation process can be incorporated into a product's own installation routine. Adoption of this update removes the necessity of insisting that your users also install *IE4*. [Reflex *informs us that the COMCTL32.DLL update is now shipped with* Disknet. *Ed.*]

There was a recommended option for a pre-install virus scan of all local drives with the *Sherlock* engine. After this, the installation was simplicity itself, though the custom install option would have complicated matters. The 'anti-virus' configuration was chosen as a default, for obvious reasons, with all local files protected from alteration, as suggested in the manual. The installation chugged a little whilst processing these files, and prompted for a reboot, advising that the disks be removed from the A: drive.

Scanning Results

The *Sherlock* scanner was tested against the usual *Virus Bulletin* test-sets as a standalone product. Its *ThunderByte* parent differs in that checksumming is not an integral part of Sherlock, and there is no associated on-access scanner. There were, in fact, very few options available in comparison with most scanners reviewed in these pages.

The scanning results were passed to a log file, though not without some idiosyncrasies. As *Disknet's* scanning is designed to be part of a gateway computer, it is only possible to scan entire disks, and only floppies at that. The LS120 and CD- ROM drives on the test machine were ignored. The recent introduction of re-writable CDs (CD-RW) opens another interesting possibility for products that should detect 'removable media', but no CD-RW drive was available for testing.

More unusually, the report file was processed in a novel interpretation of alphabetical order. The sequence was not based upon the names of files scanned, their paths, nor the order in which they were encountered. Rather imaginatively, the developers have settled on the name of the supposed infection. However, *Sherlock* is not unusual in giving different names to replicants of the same virus, resulting in a rather shambolic report file.

After processing the report file, it became more comprehensible. The ItW File test-set was close to perfectly detected, with Morphine alone proving difficult, with only two of the nine samples detected. The larger set of polymorphics was more efficiently dealt with, however – only three samples of Mad.3544 and one of SMEG_v0.3 slipped through the net. The *Word 6/95* macro viruses were not a great problem, with the little-known WM/MortalKombat the sole undetected virus. The Laroux variants were easily detected amongst the *Excel* macro viruses, though XF/Paix and XM/Robocop were not. A total of ten samples, across four viruses, were missed from the Standard test-set.

Boot Sector viruses proved something of a mixed bag, since a number of the samples were registered as unformatted disks. All the other samples were successfully identified. With the disk authentification system in place (described later), the 'unformatted', and thus unauthorized, disks should not be able to be used, and boot protection should also prevent booting from them.

Sherlock's detection in the preceding categories was satisfactory, if not perfect, which leaves the *Word 97* macros as the major area where it had difficulties. In this sub-set of the Macro test-set, eleven viruses were detected either imperfectly or not at all. In total, 41 samples of the 178 *Word 97* infections were missed.

Disknet's Macro Interceptor software was tested on an infectable machine, specifically with those *Word* macros which had remained undetected in the initial scan. Results for this test are included later.

Scanning speed was to be tested both against the *Virus Bulletin* Clean test-set and using floppies containing infected and clean files. The former proved tricky to perform under real conditions, since, with *Disknet* running, only the A: drive was scannable from within the installed components. A standalone scan with *Sherlock*, while not meaningful from a timing standpoint, showed a gratifying lack of false positives. The disk scans showed a slight increase in scanning times on infected disks, from 35 seconds on a clean disk to 48 seconds on a disk containing the same files, each infected with Natas.4744.

BrownWright Ridiculous?

Reflex Magnetics have made a great deal of reference to the BrownWright Report, which focuses heavily on the effectiveness of virus scanners. *VB* subscribers have asked whether their chosen anti-virus solution is as poor as the report insinuates. Laying aside *Reflex Magnetics'* inferences from it, there appear to be many flaws in the BrownWright test methodology.

BrownWright claims to have obtained a large number of viruses from freely accessible sites on the Internet. A 'sample' was accepted as a virus if one of the scanners used in the tests 'detected' it. This must be considered flawed. Many scanners tend to false alarm on improperly disinfected files, virus fragments or simply clean files. There is also some debate in the anti-virus industry as to whether droppers, 'joke' programs, Trojans, worms, mIRC scripts and so-called 'virus simulations' should be detected by virus scanners – it is, however, accepted that such things should not be in *virus* test-sets, although some developers will elect to detect some of them. Thus, it is likely that the 6,301 'viruses' in the BrownWright tests contain many non-viral files, despite some products 'detecting' them. No attempt was made to replicate these files, yet this is the only criterion in deciding eligibility for inclusion in the *VB* test-sets.

The report is scathing about both the consistency and quality of detection of the scanners tested. Scanners are primarily designed to detect viruses, so when presented with a collection of odds and ends intermixed with real viruses, non-detection is not only expected but may be admirable. Similarly, products reported to have detected samples in their early versions which were 'missed' by later versions may, in fact, have reduced their proneness to false-positive, rather than have decreased their detection, as the report suggests. Without having run replication tests on the 'samples', there is no basis for deciding.

VB's opinion is that the BrownWright Report conclusions are invalid, partly because of its interpretation of the facts, but largely because these are based on seriously flawed methodology.

Other Scanners

Recently, *Reflex Magnetics* has been at the centre of a major debate about the effectiveness of scanners in the fight against computer viruses. Thus, it seems strange that *Reflex* suggests supplementing *Sherlock* with one or more extra on-demand scanners for the gateway scan. The manual states that *Disknet* is compatible with *Cybec VETAVS*, *Data Fellows F-PROT*, *Dr Solomon's AVTK*, *Symantec Norton AV* and *McAfee VirusScan*, all of which were duly tested up to installation. This proceeded smoothly, but if a non-installed scanner was chosen, there was no warning that a missing product had been selected.

Additional scanners must be selected from the supplied list – there is no facility for user-specified scanners. This could be a disadvantage to potential customers who have already invested in unsupported scanners. Adding scanners to *Disknet* is possible, but requires a command line version of the scanner and cooperation between its developer and *Reflex*. [Reflex *tells us they are happy to incorporate other scanners in* Disknet *on these terms. Ed.*]

F-PROT is mentioned specifically, implying that the reference is to this product only, rather than to the modern version, which includes the *AVP* engine. Trials with *F-SECURE* demonstrated no problems, but if a virus was detected by the first scanner on the list, the disk was rejected and no other scanner triggered. If no viruses were detected by the first scanner, then the second scanner was triggered. During the scan process there were points at which the odd option to scan another disk was presented.

The Macro Interceptor

This program is a standalone application, which also interacts with *Word* and *Excel*. Whether on-demand or on-access, it is designed to detect viral macros within documents or spreadsheets, in what seems to be a heuristic fashion. Since the program can be run on its own to scan directories, this was the first option used for testing. The interceptor detected all 51 *Excel 6/95* files, and all 16 *Excel 97* files, which is good enough for any purpose. *Word 6/95* files were well detected, with only three misses out of 904 samples.

As with *Sherlock*, it was with *Word 8* files that the Macro Interceptor had problems. The program missed at least a third of the 179 samples, before crashing when faced with a file infected with W97M/Wazzu.DG. This reproducible flaw does not bode well for those who might wish to use the Macro Interceptor with complex custom macros.

Security Features

The security features of *Disknet* are its *raison d'être* and were tested, but not to the same extent as the anti-virus portion of the software. What follows must be considered an overview of the efficacy of each of these measures.

Boot protection prevented access to the C: drive when booting from another drive. The boot process progressed as usual, but the C: drive remained unusable from this clean-booted state. The Program Security Guard was supposed to protect all programs on the local drives from unauthorised tampering. This did not, however, protect all files necessary to secure its own installation. In its default configuration AUTOEXEC.BAT could be altered, including deleting the boot protection applications, allowing for a boot from A: giving access to C: without disk authorization or boot protection. As this was not the prime area of this review, there was no further investigation into this problem.

Data authorization is *Disknet's* major activity, and the program proved efficient in detecting whether a disk had been scanned and marked with an authorization code (recorded in the boot sector). This was bizarrely signalled by the message that no disk was present. After this message, the option to scan was given if an authorization code was not found. A virus-free disk was then authorized, or if read-only, authorized temporarily. The 'permanent' authorization was removed if the disk's contents were used in another machine. Thus, theoretically, only scanned disks could be used on a *Disknet*-protected machine, and no viral material could find its way onto such a machine.

Conclusion

Considered as a purely anti-viral tool, this version of *Disknet* can be seen to have both strengths and significant weaknesses. The primary weakness lies in the likelihood of *Word 8* Macro virus infections, which are detected relatively poorly by both the gateway *Sherlock* scanner and the resident Macro Interceptor. Other scanning is much more efficient, yet not perfect enough to allow *Disknet* to be used entirely on its own. Although the reasoning might be that on-demand scanning of non-removable media will not be required, it is annoying that there is no option for performing this task with *Sherlock*, except at installation.

Adding another scanner may combat these shortcomings, yet this increases the financial cost of the protection, and the time overhead for the introduction of new media. These disadvantages aside, it must be stressed that *Disknet* is primarily a security program. Such overheads may be more than compensated for by the need for security, with the advantage of a built-in anti-virus solution, integrated and less liable to be unstable than a mixed bag of products.

Technical Details

Product: *Reflex Disknet v3.04* for *Windows 95.*

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

Availability: This program requires a 386DX or higher CPU and 30 KB of free disk space.

Version evaluated: v3.04 Administrator.

Serial number: None visible.

Price: 26–100 users, £49 per node; 1000 users, £24 per node.

Hardware used: 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disk, CD-ROM drive and 3.5-inch floppy running *Windows 95 (SP1).*

Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Win95/199805/test_sets.html.

PRODUCT REVIEW 2

NetShield for NT

This review started sadly, as we had a great deal of difficulty obtaining a copy of the software! The holiday season had sent the usual *Virus Bulletin* contacts off to sunnier climes, and as a result, installation of *Network Associates' McAfee NetShield* took a slightly unusual path (albeit one familiar to users in the real world) since it involved updates and upgrades. The 'real world' being notoriously more prone to disaster than an aseptic lab, it was with some degree of trepidation that the review process commenced.

Packaging and Documentation

The *NetShield* box was at the half-full level typical of a software product, most of the contents being small – the CD and bulky manual the only exceptions. One of the slimmer manuals was the Quick Start Guide, and, seeking an easy life, this was the first of the documents to be investigated.

There was some confusion when requirements for the *NT* server product did not include *NT*, with the Guide claiming that *NetWare* was vital. The manual turned out to be a *NetWare* Quick Start Guide. Denied this useful resource, the review started with a lengthy reading session of the manual, which is quite a verbose affair. This said, the *NetShield* interface is not Byzantine in its complexity, and most users would only need the manual as a reference to what can be done, rather than how to do it.

As well as the step-by-step instructions on the best way to write-protect a disk, all the anti-virus functions provided by *NetShield* were described in detail. There were also sections concerning the update feature, general hints for the prevention of virus spread and a short virus primer.

A further section described the two-year product support, signature update and full product upgrade provisions to the standard corporate licence. Chargeable PrimeSupport options are also detailed. The last appendix provided command line options for *VirusScan* (here referred to as *scan32)*, and options for use in the *NetShield* configuration (.VSC) file (having the familiar syntax of *Windows* INI files). The other, less vital, contents of the box were the licence agreement, the BBS and service guide, the registration card and a card containing an incentive for corporate customers to register, together with a reminder of the *Network Associates* web server address.

Help files and documentation within *NetShield* itself were good; context-sensitive help is available throughout the program. This was certainly more practical than the User Guide, and on the occasions when clarification was required, the on-line help was more than sufficient to bring enlightenment.

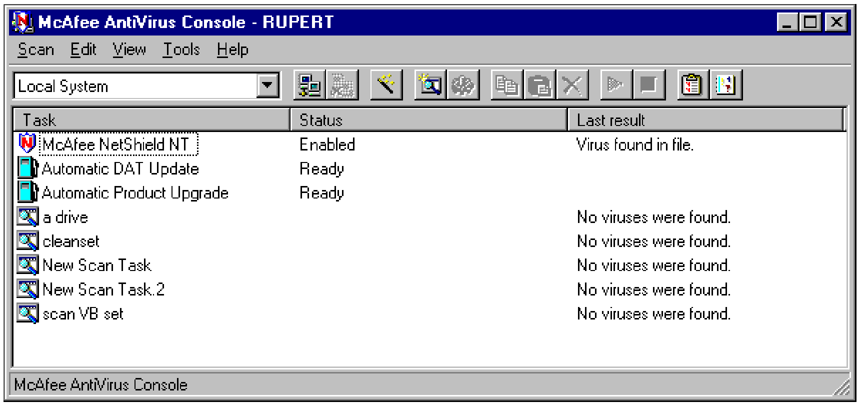
Installation

The software was installed on the server as administrator, from a CD-drive shared from a workstation. For a program subject to such variability in working conditions, the number of options during installation was remarkably small. The supplied product turned out to be antiquated – v3.03 from September 1997 – and its detection rate far worse than expected. Even with new virus data files there was a distinct sense of under-performance, so an upgrade seemed in order. Installing upgrades is not an unusual event in the administration of anti-virus products, so it seemed reasonable to test this procedure. *NetShield* v3.1.4a was obtained from *Network Associates* web server and the installation procedure was repeated, with more options apparent on this second pass.

Before, the choice would have been whether to install locally or to a remote computer. This new version of *NetShield* recalled the presence of an earlier incarnation and gave three options. The choices available were to uninstall this relict completely, to install with default settings within *NetShield*, or to install while retaining customizations. The last of these was selected, and next came a prompt for custom, compact or typical installation. Custom installation was chosen, which offered the choice of installing the alert manager, and options as to which features be activated immediately upon installation. Again, all of these were selected, and installation proceeded without hitches.

The Interface

The manual states that 'most of the functionality' of *NetShield* lies within the console. This is easy to believe, considering that there are only three relevant executables installed onto the machine after the above choices – the *VirusScan* front-end and alert manager being the others.



Task	Status	Last result
McAfee NetShield NT	Enabled	Virus found in file.
Automatic DAT Update	Ready	
Automatic Product Upgrade	Ready	
a drive		No viruses were found.
cleanset		No viruses were found.
New Scan Task		No viruses were found.
New Scan Task 2		No viruses were found.
scan VB set		No viruses were found.

Also accessible from the program group are an uninstaller, a readme file, a 'what's new' document and a file containing details of product resellers. A mini-console is also installed in the task bar, giving access to the Properties tab, as well as statistics for on-access scanning, control over the messaging feature and the ability to activate or deactivate the on-access scanner. The main console is also available from here, together with the ubiquitous 'about' tabs of so little use in day-to-day operations.

On-demand scanning is available through two paths – the console, where reusable scheduled scans are arranged, or the *VirusScan* module, used for immediate scans where simplicity rather than flexibility is required. The *VirusScan* module allows immediate scans to be performed upon selected files, drives or folders. It also functions as a command line scanner.

The console is the heart of *NetShield*, and a complicated organ to explain in text. The default view shows all the pre-prepared tasks, along with toolbar and options tabs and a short summary of the task's last scan. From here, the tabs available are Scan, Edit, View, Tools and Help. New scan tasks are configured from the Scan tab and a 'task wizard' provides step-by-step guidance through the process.

A Browse option makes the selection of network drives or sub-directories simple, but, following selection, the Browse location resets to the root on the local machine. Happily, several files, drives or folders may be selected in combination, thus providing flexibility when producing complex scheduled tasks. Tasks may also be edited and duplicated here, and the log inspected. Most importantly, the 'properties' sub-console may be entered from here.

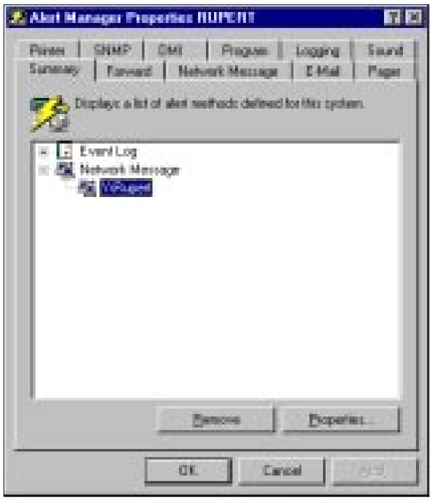
If the console is the heart of *NetShield*, the 'Properties' area is the soul. Here, almost all the most important options within scanning tasks can be selected. These include the actions to take when infections are detected; all the standard options are available. As expected, the location and contents of report files, and the size limits upon them, may also be controlled. Furthermore, exclusions from scanning may be selected, the default being PAGEFILE.SYS and the quarantine directory created by *NetShield* itself. The extensions to be scanned may also be chosen. Again, the standard extensions are included, with the addition of the somewhat more obscure SYS, BIN, RTF and OBD.

The inclusion of alternative document file formats shows a certain cunning. WM/CAP, which is very common, intercepts attempts to save documents as RTF format. It forces them to save in native *Word* format but with the RTF extension, ensuring a wider spread from users naïvely mistaking file extension for file format. The option of scanning within compressed files is toggled from the console too, CAB files being supported in addition to the more commonly supported compression formats. Only one obvious problem seemed liable to occur from the default settings – the option for skipping the scanning of bootsectors was selected by default.

The other console tabs are less likely to be accessed on a regular basis. For the purposes of labour-saving it is possible for tasks to be copied, imported and exported from the Edit tab. The View tab is for the control of more aesthetic options, while the Help tab accesses not only the general help, but also the 'Online Virus Information Library'. Finally, the Tools tab is used for the alteration of alerts, the configuration of autoupdates and the connection to other *NetShield*-installed servers, in order to perform remote administration.

Reports and Alerts

There is an impressive variety of ways in which *NetShield* is able to alert administrators to infections: this should cater to the needs of even the most paranoid. The alert manager is installed as a separate program which may be accessed from numerous places within the *NetShield* suite. Alerts may be configured as to wording, ultimate destination and delivery method. Messages can be sent to other designated computers or as general network messages. Target computers may have a priority level associated with them, so as to display messages exceeding the chosen level. Since alerts may be issued for *NetShield* internal errors as well as the detection of viruses, this feature can be used to filter information to the relevant people. Alerts may also be sent by email, to a printer or via SNMP traps or pagers. If these choices are not sufficient, there is the option of calling a specified application.



Alert Manager Properties RUPERT

Displays a list of alert methods defined for this system.

Updates and Rollouts

AutoUpdate allows the automatic retrieval and installation of new virus definition files and software upgrades. This is relatively simple from within the network, and the earlier (somewhat clumsy) scripts are no longer necessary for automatic updates from the *Network Associates* web site or other external site. A great improvement in this feature is that it now handles client scanner upgrades as well as virus data file updates. There are many configuration options for frequency of update availability checks and the like.

Results

The on-access and on-demand scans achieved identical results against the In the Wild test-set, both managing a clean sweep in this vital area. This uniformity was also apparent against the Macro test-set, where both scan settings failed to detect 43 of 1150 samples – among them the *Word 97* macro viruses W97M/AntiSR1 and a good selection of W97M/Blee variants. The relatively recent *Excel* formula virus, XF/Paix, was also missed. This was somewhat surprising, as the reviewed version was released sometime after *Network Associates* first claimed to detect this (then new) virus.

ItW Boot test results were impressive, with a 100% detection rate, although there were problems with on-access scanning, as was the case in last month's comparative review. On several occasions, the background operations of *NetShield* proved sufficient to lock *NT* Explorer into a resilient state of catatonia, and blue screens resulted more than once while scanning the Paula boot virus.

As we have seen in several products in recent *Windows 95* and *NT* comparatives, *NetShield* seemed to have trouble detecting diskette changes. This resulted in disks being detected upon their first scan only if they were left in the drive and scanned repeatedly. On one occasion, an alert occurred when there had been no apparent disk activity for some minutes. On-demand, the boot sector viruses were more easily detected, but there was no user-friendly method for speedily scanning many floppies.

It was against the Standard test-set that slight variability began to emerge between the on-access and on-demand scanning results. The on-demand scan missed 18 of the 906 virus samples (98.0%). The on-access scanner missed the same 18, with the addition of one more (97.9%).

The 13,500 sample Polymorphic test-set, was the area where the results became widely disparate. The on-demand scanner missed 59 samples (all of Cryptor – 99.6%), while the on-access scanner missed 225, (Cryptor, DSCE.Demo, Giraffe:TPE, Gripe.1985, PeaceKeeper.B and Sepultura – 98.3%). These results are exactly the same as those seen in the *Windows 95* comparative in the May issue. This implies a weakness in the on-access scanner, rather than buffering problems caused by an excessive input of tricky, time-consuming polymorphics.

A look at the on-demand results for the earlier v3.0.3 engine, supplied with the same virus definition file, showed some interesting changes. It seems that this engine was incapable of fully detecting Spanska.4250, despite being supplied with the same definitions as the 3.1.4a engine. This resulted in 332 misses across all test-sets for the former combination, as opposed to 119 for the latter.

The moral here must be that upgrading virus signatures alone is not enough to maintain the highest possible efficacy of an anti-virus product. Since *Network Associates* offers free signature updates and product upgrades to the main *NetShield* program throughout the two-year licence period, this should not be problematic for users who are already investing in regular data file upgrades. One of the hypotheses of the BrownWright Report (see p.19) – that newer versions of virus scanners fail to detect viruses that were detected by older versions – was in no way supported by these results.

Scanning the Clean set of 5500 executables on the server disk took 1213 seconds – a far from impressive result. This test was repeated over a network link, scanning the Clean set CD from an *NT* workstation share, which took 1850 seconds. Copying the CD's contents to the server took 1290 seconds. Happily, there were no false positive reports.

Overhead times for the on-access components were checked for comparison against timings for file transfers when no on-access scanner was operating. Relevant options available are whether files scanned are incoming or outgoing, and whether the scan cache is enabled. In comparison with the baseline, the scanner showed no overhead when checking only outward files, though this was to be expected under the circumstances of the test – the files were only moved within the local machine. However, the transfer was enough to trigger the routine for inbound files, as this registered a 100% overhead in copy times. The same was seen when both inbound and outbound traffic was being monitored, which is the default setting for the scanner.

There was also an option to disable scan file caching, which upped the overhead to a massive 240%. These results are sluggish at best, and will be seen by many users to be approaching the unendurable. In contrast to this slothful performance, the scanning rate of infected disks matched that of uninfected ones, where the same files were used.

Conclusion

NetShield is a feature-filled bundle, which performs well on-demand. On-access, however, affairs are less satisfactory. Large overheads will not cheer system administrators, polymorphic detection is degraded and a possible server crash when accessing infected diskettes may be worse than the 'disease'. This is another example of a product where many good details are let down by a few easily-fixed flaws.

Technical Details

Product: *McAfee NetShield v3.1.4a for NT Server.*

Developer/Vendor: *Network Associates Inc*, 2805 Bowers Ave, Santa Clara, CA 95051, USA, Tel +1 408 9883832, fax +1 408 9709727, email ordermaster@nai.com, WWW http://www.nai.com/.

Availability: This program requires *NT Server v3.51* or later with 4 MB free disk space to install.

Serial number: None visible.

Price: 10 users, $257; 50 users, $1109; 100 users, $1984. Prices for other numbers of licences are available from the vendor.

Hardware used: Server: *Compaq* Prolinea 590, 80 MB of RAM, 2 GB hard disk, running *NT Server v4.0 (SP3).* Workstations: Two 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disk, CD-ROM drive and 3.5-inch floppy drive running *Windows NT v4.0 (SP3).*

Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Win95/199805/test_sets.html.

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

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park,
Abingdon, Oxfordshire, OX14 3YP, England

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

US subscriptions only:

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

Tel (781) 9377768, Fax (781) 9320251

END NOTES AND NEWS

***Dr Solomon's* is hosting a live virus workshop** at the Barns Hotel, Bedfordshire, UK, from 14–15 July 1998. The intensive, hands-on course costs £695 +VAT. For more details, contact Caroline Jordan; Tel +44 1296 318881 or email Caroline.Jordan@drsolomon.com.

Reflex Magnetics Ltd* announces two new 32-bit releases of the *Disknet Security Suite for deployment throughout *Windows NT* and/or *Windows 95* – the Administrator is for use at the server, and the Client for use on remaining network PCs. The Administrator is equipped with Sherlock (a virus scanner based on the *Norman ThunderByte* engine) and a Macro Interceptor. There is a standalone review of the Administrator version on p.18 of this issue.

***Sophos* is hosting a practical *NetWare* security course** at its training suite in Abingdon, UK on 9 July 1998. The one-day, intensive course costs £325 +VAT. From 15–16 July, *Sophos* will also run a **live virus workshop** in Abingdon, which costs £595 +VAT. For details, contact Karen Richardson; Tel +44 1235 544015, fax +44 1235 559935 or visit the company's web site http://www.sophos.com/.

***Network Associates Inc* is to acquire *Secure Networks Inc* (*SNI*)**, providers of security assessment software and computer security research and consulting. *SNI's* security auditing scanner, Ballista, is to be incorporated in *Network Associates' Net Tools Secure Suite* for retail at £65 per user for 1000 users. Contact *Network Associates* for more information; Tel +44 1753 827500 or see the company web site http://www.nai.com/.

***Data Fellows* is establishing a worldwide network of certified AntiVirus Centers (CACs)**. Organizations of this kind already function in Belgium, Estonia, Finland, Germany, Hungary, Hong Kong, Italy, Japan, the Netherlands, Norway, Slovenia, South Africa, Sweden, America and the United Kingdom. There are plans for ten more countries to be added to that list in 1998. The concept is based on a certification program for the technical staff of *Data Fellows* partner companies. Each CAC has at least one Certified Anti-virus Expert (CAE) whose job it is to 'monitor potential virus threats, predict future hazards and detect and destroy computer viruses on a daily basis', according to *Data Fellows'* Product Manager Mikko Hypponen; Tel +358 9 859900, fax +358 9 85990599 or email Mikko.Hypponen@DataFellows.com.

On Thursday 11 June 1998, a seminar entitled 'Defence beyond the firewall' is to be co-hosted by *Content Technologies* (formerly *Integralis*), developers of *MIMEsweeper*, and *Sophos*. The event commences at 9am at the Cavendish St James Hotel, London, UK. To book your place at the seminar contact Fiona Melville; Tel +44 118 9301300, or email seminar@mimesweeper.com.

Compsec'98, the fifteenth World Conference on Computer Security, Audit and Control, runs from 11-13 November 1998, at the Queen Elizabeth II Conference Centre in London, UK. For details and a registration form, contact Amy Richardson; Tel +44 1865 843643, fax +44 1865 843958, email a.richardson@elsevier.co.uk, or visit the conference web site http://www.elsevier.nl/locate/compsec98/.

***Trend Micro Inc* announces the release of *Trend Virus Control System* (*VCS*)**, a web-based management console for the central control of configurations and updates of multiple anti-virus products on multiple platforms across the network. Notification of viral activity is via standard SNMP trap, email or *Windows NT* event logging. The product requires *Windows NT Server v4.0* or later and is available for download from http://www.antivirus.com/. Contact Peapod, *Trend's* UK distributor, for details; Tel +44 181 6069990.

Research shows that the most popular concern at *Infosecurity* last year was Internet security. **60% of visitors to *Infosecurity '98* sought products in the network security field, while 38% were concerned with virus protection.** At this year's show in April, 60 new products were launched and two government announcements made. Plans are under way for *Infosecurity '99*, to be held at National hall, Olympia, London from 27–29th April. For more information about next year's show contact Yvonne Eskenzi; Tel +44 181 4498292 or email yvonne@eskenzi.demon.co.uk.

Symantec Corporation* anounces the release of *Norton AntiVirus* (*NAV*) *for Microsoft Exchange Server, the installation and configuration of which is from a single central console. *NAV* customers can now receive **weekly updates** by downloading Intelligent Updater directly from the *Symantec* web site http://www.symantec.com on a Thursday evening, or using the company's exclusive LiveUpdate feature. For more information, contact Gunilla Larsson; Tel +44 1628 592384 or email glarsson@symantec.com.