ISSN 0956-9979 MARCH 1996

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **The voice of authority.** This month, *Virus Bulletin* undertakes a review of several disk authorization products. For the results of the tests, and a detailed analysis, turn to p.12.

• **More to shout about.** The first *Ami Pro* macro virus has appeared: although it is not in the wild, *VB*'s reviewer is worried about the implications. An analysis is on p.11.

• **Swift research.** Jimmy Kuo is a relative newcomer to the anti-virus world, yet in the three short years he has been working in the field, he has risen to head of *McAfee*'s anti-virus research group. What makes the man so successful? Turn to p.7 for an insight.

CONTENTS

EDITORIAL

Boza: The Circle of Life

It has been compared to the Michelangelo scare of 1992, it has generated a frenzy of media interest the like of which has not been seen for quite some time, and it has computer users all over the world in something of a frenzy. What is it? Perhaps a radically new type of virus with an incredibly destructive payload which is all over real-world computers? Erm… not quite.

Boza (for this is the virus at the centre of it all) is none of these things – it is not in the wild, it is largely uninteresting, and apart from infecting programs, it does not destroy data. It is simply a perfectly normal non-resident, direct action infector – dozens of new viruses of this type are seen every month. There is a catch, however, and a big catch at that: Boza is a *Windows 95* virus [*see VB, February 1996, p.15*]. As soon as that is mentioned, everything is different – this is big news, hold the front page!

“ Windows 95 does not in fact act as some form of magical virus deterrent ”

It is often the case that the important things are those which come first. These are the ones that grab the public's attention. This is especially true in the field of computing – the first product of type X for system Y always gets disproportionately more notice than any subsequent products in the same genre. Apart from anything else, it proves that something is possible.

This is exactly what Boza has done – it has grabbed the attention of the mass media simply because it is the first *Windows 95* virus. It would appear that people were unaware that *Windows 95* was vulnerable to viruses in the same way as other operating systems: one journalist said to me: ‘But isn't *Windows 95* protected?’. It is not clear exactly what he meant by ‘protected’, but others have come up with similar statements, to which the only answer is: ‘It's not protected, just different’.

And not even very different, at least not at the level at which Boza functions – the main differences between *Windows 95* and DOS, as far as this particular virus is concerned, are the new file format and the method by which system calls are made.

Meanwhile *VLAD* magazine (produced by the people behind Boza) seems somewhat miffed – not at the fact that it has received gratifyingly little publicity from the whole affair, but at the name. The fact that this virus has been assigned the name Boza (a piece of Bulgarian slang meaning something which has been badly put together, and also the name of a slightly alcoholic drink made from millet), ignoring the message within the virus referring to it as Bizatch, has not gone down well. However, the anti-virus industry can choose to call viruses whatever it wishes, and in cases such as this, where publicity is bound to come along sooner rather than later, it seems a good idea to deny virus authors at least some measure of infamy by shunning the name of their choice and opting for a different one.

VLAD's work on Boza is only really noteworthy for the fact that they got there first. Now Boza is available on several Internet sites and numerous BBSs, it must be assumed that the floodgates will open, and we will see more of the same, at least at first. It is also reasonable to believe that the development will follow much the same course as viruses for DOS – gradually, this new breed of *Windows 95* viruses will become more advanced, complex, and dangerous, until the cycle repeats all over again with the next big new operating system, a few years from now.

Or will it? Perhaps, for whatever reason, things will be different this time. That would be nice – but it is ambitious to speculate on something so dependent on such a variety of factors, including how popular the new operating system becomes.

Anyway, Boza has had its fifteen minutes of fame, the press has packed up and left for greener pastures, and the world now knows that *Windows 95* is not virus-proof after all. From that point of view, the flurry of interest may well prove to have had a useful effect: the public is at least aware that *Windows 95* does not in fact act as some form of magical virus deterrent.

Welcome to the new age.

NEWS

InocuLANding

In mid February, testing of *Cheyenne's InocuLAN for Windows NT* by UK-based *PC Week* magazine revealed an bug in the product – when installed on a system, it silently creates an account, 'InocuLAN', with administrator privileges, whose password is constant across all installations.

InocuLAN uses this account to provide many of its more advanced features, such as cross-server product management. The fault is not that the account has been created, but that the user is neither informed of, nor offered the chance to change, the password associated with the program. The administrator can change the password, just as with any other account on the system, but the chances are high that he will not even notice the existence of the new account.

Cheyenne states that the company informed all licensees of the product as soon as the problem was discovered, and immediately stopped shipping that version. A new release, version 1.01, was shipped on 15 February: this, when installed, prompts the user for a password to assign to the new account. Administrators are advised to upgrade, but in the short term they may simply change the password on *InocuLAN's* account to one of their own choosing.

For more information, visit *Cheyenne's* WWW site at http://www.cheyenne.com/, or contact the company on Tel +1 516 484 5110, fax +1 516 629 1853 ■

Infectious Bard

Readers will note that in the update to the IBM PC tables of viruses this month there appear a large number of viruses with a Shakespearean theme. These viruses have evidently all been written by the same author(s), the credits within them being very similar. *Virus Bulletin* hopes to bring you more information on these viruses in the near future ■

To Hoax or not to Hoax

Growth in the use of the Internet has brought a corresponding growth in the number of hoaxes placed there, in any one of a variety of different forms. With the sheer number of such hoaxes, it is not worthwhile to report every one in these pages; however, every now and then one is either sufficiently serious or sufficiently amusing to mention.

The latest was sent to a couple of virus-related lists and newsgroups on the Internet on February 2. Purporting to be from Alan Solomon, the message claims to describe a 'very advanced polymorphic virus with a Trojan side effect' [*! Ed.*] within the latest version of *Free Agent*, a popular mail and Usenet news reader for *Windows* from *Forté*. It went on to state: 'In order to clean your hard disk of this virus you must first do a low level format'.

Prevalence Table – January 1996

Virus	Incidents	(%) Reports
Concept	58	15.8%
Form	41	11.2%
AntiEXE.A	33	9.0%
Parity_Boot.B	31	8.5%
AntiCMOS.A	28	7.7%
Empire.Monkey.B	17	4.6%
Sampo	17	4.6%
Junkie	13	3.6%
Ripper	12	3.3%
BuptBoot	8	2.2%
NYB	8	2.2%
Stoned.Angelina	8	2.2%
Jumper.B	7	1.9%
V-Sign	7	1.9%
Quandary	6	1.6%
Stealth_Boot.C	6	1.6%
Natas.4744	5	1.4%
Telefonica	5	1.4%
Unashamed	5	1.4%
Feint	4	1.1%
EXEBug	3	1.0%
Stoned.NoInt	3	1.0%
Mange-Tout	2	0.6%
Manzon	2	0.6%
One_Half	2	0.6%
Parity_Boot.A	2	0.6%
She_Has	2	0.6%
W-Boot.A	2	0.6%
Yankee_Doodle.TP-44.A	2	0.6%
Other *	27	7.4%
Total	366	100%

* The Prevalence Table includes one report of each of the following viruses: AntiCMOS.Lixi, Barrotes, Boot.437, Bye, Colors, Delwin, Diablo, Die_Hard.400, Empire.Monkey.A, Green_Caterpillar, Green_Caterpillar.B, Helloween.1182, Hidenowt.1747, HLLC.Happy_Monday, Int40, Invisible Man, Maltese_Amoeba, MtE.?, Necros, November_17th.?, Nuclear, Stoned.Kiev, Stoned.Manitoba, Stoned.No_Int, Stoned.Standard, TaiPan, Tornado.

Needless to say, the alert is false, and was so unconvincing – riddled as it was with spelling mistakes and factual errors – that it should have fooled almost no one. As far as is known, there is no malware in any past or present version of *Free Agent*, and it should never be necessary to perform a hard drive format in order to remove a virus or a Trojan horse ■

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 21 February 1996. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

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

Burglar.1150

ER: An appending, 1150-byte virus with stealth capabilities, which contains the text: 'AT THE GRAVE OF GRANDMA' and 'Burglar/H*.*'.

```
Burglar.1150        A900 0E1F 33FF B935 05FC F3A4 8ED9 FA8C 8784 00C7 8782 0058
```

Caesar.867

CN: An appending, 867-byte, direct infector which infects up to four files at a time. When an infected file is run, it displays the famous 'Friends, Romans, countrymen' speech from Shakespeare's *Julius Caesar*, Act 3 scene iii. Another plain-text string inside the virus code reads: 'Julius Caesar Thespian I-EAS Virus Creation Centre v0.19a [JC] [Th] [IE-VCC v0.19a]'. The virus marks all infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
Caesar.867          ED0B 00EB 02CD 208D B6F7 02BF 0001 A5A5 0E1F 8D96 6903 B41A
```

ExeHeader.NLA

ER: A 383-byte virus with stealth capabilities, which inserts its code into the file's EXE header. The virus signature is visible at the end of a header (offset 01FDh) and reads: 'NLA'. From the operating system point of view, all infected files become COM files.

```
ExeHeader.NLA       CD7D 7210 9C50 2EA0 0301 3C02 7409 3C03 7405 589D CA02 0053
```

Funky.325

CN: An appending, 752-byte direct infector containing the plain-text message: 'AV Funkware Evaluation League of [NuKE'94]*.c?m'. All infected files have their fourth byte set to 20h.

```
Funky.325           5E83 EE03 56FC 81C6 1101 BF00 01A5 A55E 8D94 1A01 B41A CD21
```

Gisela.99

CER: An overwriting, 99-byte virus containing the text: 'GISELA'. Since the virus does not check whether or not it is already active in memory, it installs the new Int 21h hook each time an infected file is run (the amount of available memory shrinks every time an infected program is executed).

```
Gisela.99           B440 B963 008E 1E41 01BA 0001 2EFF 1E3D 01B4 3E2E FF1E 3D01
```

Hallo.749

CN: An appending, 749-byte, direct infector which contains this plain-text message, displayed when the virus is run: 'Hallo! I have got a virus for you!'.

```
Hallo.749           02B8 0042 CD21 7239 BA10 01B9 ED02 8B1E CC02 B440 CD21 7229
```

Hallo.812

CN: An appending, 812-byte, direct infector containing this message displayed during infection: 'Hello! There is a new virus in your computer! Good luck!'. It carries a dangerous payload: when triggered, it overwrites the contents of the current disk.

```
Hallo.812           02B8 0042 CD21 723D BA10 01B9 2C03 908B 1EE3 02B4 40CD 2172
```

Hamlet.1144

CN: An appending, 1144-byte, direct infector which infects up to four files at a time. When an infected file is run, it displays the well-known speech from Shakespeare's *Hamlet* (Act 3 scene i), which begins: 'To be, or not to be: that is the question'. Another plain-text string inside the virus code reads: 'Hamlet Thespian I-EAS Virus Creation Centre v0.19a [Hl] [Th] [IE-VCC v0.19a]'. The virus marks all infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
Hamlet.1144         ED0B 00EB 02CD 208D B60C 04BF 0001 A5A5 0E1F 8D96 7E04 B41A
```

Henry.IV.857

CN: An appending, 857-byte, direct infector infecting up to six files at a time. When an infected file is run, it displays a speech from Shakespeare's *Henry IV Part 2* (Act 3 scene i). Another plain-text string inside the virus code reads: 'HenryIV Thespian I-EAS Virus Creation Centre v0.19a [Hy] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') in the fourth byte from the start of the file.

```
Henry.IV.857        ED0B 00EB 02CD 208D B6ED 02BF 0001 A5A5 0E1F 8D96 5F03 B41A
```

Henry.V.735

CN: An appending, 735-byte, direct infector infecting up to seven files at a time. An infected file, when run, displays the speech from Act 3 scene i of *Henry V Part 2*, which begins: 'Once more unto the breach, dear friends, once more; Or close the wall up with our English dead!'. Another plain-text string inside the virus code reads: 'HenryV Thespian I-EAS Virus Creation Centre v0.19a [HV] [Th] [IE-VCC v0.19a]'. The virus marks all infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
Henry.V.735         ED0B 00EB 02CD 208D B673 02BF 0001 A5A5 0E1F 8D96 E502 B41A
```

Henry.VI.592

CN: An appending, 592-byte, direct infector which infects up to four files at a time. An infected file when run displays the 'corrupted youth' speech from Shakespeare's *Henry VI Part 2*, Act 4 scene vii. A plain-text string inside the virus code reads: 'HenryVI Thespian I-EAS Virus Creation Centre v0.19a [H6] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C'), placed in the fourth byte from the start of the file.

```
Henry.VI.592        ED0B 00EB 02CD 208D B6E4 01BF 0001 A5A5 0E1F 8D96 5602 B41A
```

Henry.VIII.1263

CN: An appending, 1263-byte, direct infector which infects up to five files at a time. An infected file, when run, displays the 'long farewell' speech from Shakespeare's *Henry VIII*, Act 3 scene ii. Another plain-text string inside the virus code reads: 'Henry8 Thespian I-EAS Virus Creation Centre v0.19a [H8] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') in the fourth byte from the start of the file.

```
Henry.VIII.1263     ED0B 00EB 02CD 208D B683 04BF 0001 A5A5 0E1F 8D96 F504 B41A
```

Kali.641

CN: An appending, 641-byte, direct, fast infector. It contains a plain-text string: 'KALI-4 / Köhntark' and an encrypted string: '*.COM *.EXE'. Two variants are known, both detected by the following template.

```
Kali.641            B906 008B FE81 C799 0281 C664 02A5 F755 FEE2 FA5E 8D94 9902
```

Kali.655

CN: An appending, 655-byte, direct, fast infector. It contains the following plain-text string: 'KÆ£¡-5 ≡ √ïŸä£ ¢hück [BΣ£]' (which reads: 'KALI-5 = ViraL chuck [BEL]'). It also contains an encrypted string: '*.COM *.EXE'. All infected COM files have their fourth byte set to 20h; all EXE files have the value of the checksum in the header (offset 12h) set to 77h.

```
Kali.655            B906 008B FE81 C7A7 0281 C672 02A5 F755 FEE2 FA5E 8D94 A702
```

KingJohn.667

CN: An appending, 667-byte, direct infector infecting up to eight files at a time. When an infected file is run it displays a speech from Shakespeare's *King John*, Act 5 scene vii. Another plain-text string inside the virus code reads: 'King John Thespian I-EAS Virus Creation Centre v0.19a [KJ] [Th] [IE-VCC v0.19a]'. The virus marks all infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
KingJohn.667        ED0B 00EB 02CD 208D B62F 02BF 0001 A5A5 0E1F 8D96 A102 B41A
```

KingLear.917

CN: An appending, 917-byte, direct infector infecting up to five files at a time. When an infected file is run it displays a speech from Shakespeare's *King Lear*, Act 5 scene iii, which begins: 'Come, let's away to prison'. Another plain-text string inside the virus code reads: 'King Lear Thespian I-EAS Virus Creation Centre v0.19a [KL] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
KingLear.917        ED0B 00EB 02CD 208D B629 03BF 0001 A5A5 0E1F 8D96 9B03 B41A
```

Kobrin.489

CN: A prepending, 489-byte, direct, fast infector containing the plain-text strings: 'BrPI-Kobrin' and 'Andy said:Zarin is dangerous'. It contains a payload which triggers on eleventh and 23rd days of each month and includes a routine which overwrites the hard disk boot sector.

```
Kobrin.489          BAE6 02B8 2425 CD21 B42A CD21 80FA 0B74 0780 FA17 7402 EB0B
```

Kobrin.491

CN: A prepending, 491-byte variant, direct, fast infector containing the plain-text strings: 'BrPI-Kobrin' and 'Andy said:Zarin is dangerous'. It contains a slightly modified payload which triggers on the same days as the previous variant, and also includes a procedure to overwrite the first hard disk.

```
Kobrin.491          24BA C102 B425 CD21 B42A CD21 80FA 0B74 0780 FA17 7402 EB11
```

Loves.1198

CN: Appending, 1198-byte, direct infector infecting up to eight files at a time. An infected file when run displays the 'qualities of love' speech from Shakespeare's *Love's Labour's Lost*, Act 4 scene iii. Another plain-text string inside the virus code reads: 'Love Thespian I-EAS Virus Creation Centre v0.19a [LL] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') in the fourth byte from the start of the file.

```
Loves.1198          ED0B 00EB 02CD 208D B642 04BF 0001 A5A5 0E1F 8D96 B404 B41A
```

Mackbeth.753

CN: An appending, 753-byte, direct infector infecting up to four files at a time. When an infected file is run it displays the speech beginning: 'Be innocent of the knowledge, dearest chuck, Till thou applaud the deed' from Shakespeare's *Macbeth*, Act 3 scene ii. Another plain-text string inside the virus code reads: 'Macbeth Thespian I-EAS Virus Creation Centre v0.19a [Mb] [Th] [IE-VCC v0.19a]'. The virus marks all infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
Mackbeth.753        ED0B 00EB 02CD 208D B685 02BF 0001 A5A5 0E1F 8D96 F702 B41A
```

Merchant.793

CN: An appending, 793-byte, direct infector which infects up to five files at a time. An infected file when run displays the text of the famous 'Hath not a Jew eyes?' speech from Shakespeare's *Merchant of Venice*, Act 3 scene i. Another plain-text string inside the virus code reads: 'Merchant of Venice Thespian I-EAS Virus Creation Centre v0.19a [MV] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
Merchant.793        ED0B 00EB 02CD 208D B6AD 02BF 0001 A5A5 0E1F 8D96 1F03 B41A
```

Midsummer.813

CN: An appending, 813-byte, direct infector infecting up to eight files at a time. When an infected file is run it displays the speech beginning: 'Question your desires; Know of your youth, examine well your blood' from Shakespeare's *A Midsummer Night's Dream*, Act 1 scene i. Another plain-text string inside the virus code reads: 'Midsummer Night Thespian I-EAS Virus Creation Centre v0.19a [MD] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') in the fourth byte from the start of the file.

```
Midsummer.813       ED0B 00EB 02CD 208D B6C1 02BF 0001 A5A5 0E1F 8D96 3303 B41A
```

MuchAdo.565

CN: An appending, 565-byte, direct infector infecting up to four files at a time. When an infected file is run, it displays text beginning: 'Sigh no more, ladies, sigh no more, Men were deceivers ever' from Shakespeare's *Much Ado about Nothing*, Act 2 scene iii. Another plain-text string inside the virus code reads: 'Much Ado About Noth Thespian I-EAS Virus Creation Centre v0.19a [MD] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') placed in the fourth byte from the file's start.

```
MuchAdo.565         ED0B 00EB 02CD 208D B6C9 01BF 0001 A5A5 0E1F 8D96 3B02 B41A
```

Othello.585

CN: An appending, 585-byte, direct infector infecting up to six files at a time. An infected file, when run, displays text from Shakespeare's *Othello*, Act 3 scene iv ('there's magic in the web of it'). Another undisplayed text reads: 'Othello Thespian I-EAS Virus Creation Centre v0.19a [Oo] [Th] [IE-VCC v0.19a]'. The virus marks infected files with byte 43h ('C') placed in the fourth byte from the file's start.

```
Othello.585         ED0B 00EB 02CD 208D B6DD 01BF 0001 A5A5 0E1F 8D96 4F02 B41A
```

Romeo.625

CN: An appending, 625-byte, direct infector infecting up to seven files at a time. When an infected file is run it displays the lines spoken by Juliet in Shakespeare's *Romeo and Juliet* ('O Romeo, Romeo! wherefore art thou Romeo?' etc; Act 2 scene ii). Another plain-text string inside the virus code reads: 'Romeo and Juliet Thespian I-EAS Virus Creation Centre v0.19a [RJ] [Th] [IE-VCC v0.19a]'. The virus marks all infected files with byte 43h ('C') placed in the fourth byte from the start of the file.

```
Romeo.625           ED0B 00EB 02CD 208D B605 02BF 0001 A5A5 0E1F 8D96 7702 B41A
```

Rubbit.3164

CER: An encrypted, stealth, appending, 3164-byte virus from Taiwan. It contains the text: '♥RuBBit ♥## RuBBit Version 2.2 Written by [P.F] in Taiwan. ## ## This idea from Dark Slayer. 1994/05/02 ##'. Infected programs contain the plain-text signature at the end of file: 'RuBBit'.

```
Rubbit.3164         E81E 0006 0E0E 1F07 8BDE 81C6 2B00 8BFE B91B 0CAC C0C8 04AA
```

PSMPC.Draculea

ER: An encrypted, appending, 520-byte virus containing the texts: 'Fuerst_Vlad_Draculea_II [Draculea]' and 'MPC\G2'. Infected files are marked with the string 'AD' located in the file header at offset 10h.

```
PSMPC.Draculea      BB?? 00B9 FD00 2E81 ???? ??83 EBFE E2F6
```

SX.731

ER: An encrypted, appending, 731-byte, fast direct infector. It contains the text: '-THUNDER STRUCK- (c) SX Written in RSA.' and 'Too many TSR programs loaded! Out of memory.'.

```
SX.731              BD?? FFB8 ???? 8DB6 1701 B962 012E 3104 D1C0 83C6 02E2 F6??
```

Trance.1688

CR: An encrypted, appending, 1688-byte virus with stealth capabilities. It contains the text: 'Trance Virus (c) 1955 by The Nuker' and '*.com'.

```
Trance.1688         BB?? ??8D B712 01B9 4303 2E31 1C83 C602 E2F8
```

Trance.1982

CR: A polymorphic, stealth, appending, 1982-byte virus containing the text: 'This sector has been fucked up courtesy of Trance[2] Virus (c) 1995 by the Nuker'. The limited polymorphic decryption routine has a constant length of 128 bytes. The following template detects the virus in memory.

```
Trance.1982         B830 30BB ADDE CD21 3DAD DE75 03E9 9A00 33C0 8ED8 8E06 8600
```

Turner.YooHoo

CER: A polymorphic, inserting, circa 3300-byte virus residing in Upper Memory. The virus contains the encrypted text: 'The Turner Virus Version 3.56 The Virus was written in Voronezh in July of 1993 Modified and enhanced with mutant engine & high implantation In July - November of 1995 by RingWraith TRY TO CATCH ME IF U RELY K00L CMOS …perverted MBR …perverted Whole disk …wait'. Other strings read 'EMMXXXX0', 'YooHoo' and 'SHUTEND'. The following template will detect the virus in memory.

```
Turner.YooHoo       80FC CF0F 84B8 0080 FC3D 7412 80FC 4374 0D80 FC56 7408 80FC
```

Valentine.2332

ER: An encrypted, appending, 2332-byte virus containing the text: '[A Happy Valentine To All Secret Lovers In The World] [Greetinx From BlitzBit@:TwiLightZone!] Internal SCAN V2.01 McAfee Inc. (c) Copyright, 1994 Self-check failed! WARNING! Windows.xxx Virus Found! This virus is known to be extremely harmful to your health ! Remove virus (Y/Y) ? Cleaning started… Done ! Virus removed. Have a nice day and sleep well! *.exe *.* \WINDOWS'. The virus infects EXE programs during execution. Additionally, after installing itself in memory, it infects one file in the current directory.

```
Valentine.2332      8DB6 3400 B9E8 082E 8A86 0E00 2E30 0446 49BF 75F8 7402 EBFA
```

WereWolf.658

EN: An encrypted, 658-byte direct infector, containing the text: 'Home Sweap Home (C) 1994-95 WereWolf' and '\ *.MS *.CPS ANT*.DAT'.

```
WereWolf.658        81FF 8002 72F4 C3E8 EDFF C606 8402 B8CD 21C6 0684 0281 EBDF
```

WereWolf.685

EN: An encrypted, 685-byte direct infector, containing the text: 'FANGS (C) 1994-95 WereWolf' and '\ *.MS *.CPS ANT*.DAT'.

```
WereWolf.685        BF13 002E B835 ???? 4747 81FF A302 72F3 C32E C606 A802 81EB
```

WereWolf.1208

CER: A stealth, 1208-byte virus which appends its code to EXE files and prepends it to COM files. The virus contains the plain-text strings: 'BEAST (C)1995 WereWolf' and 'CLEAN AVP TB QB SCAN COMM NAV V FINDV GUARD FV CHKDS F-PR'. The latter represents a list of files which the virus does not infect (e.g. COMMAND.COM).

```
WereWolf.1208       BB88 00FF 37FF 7702 C707 BE01 8947 02C7 0606 00F0 FF89 0E04
```

INSIGHT

cjkuo@mcafee.com

Jimmy Kuo is a Taiwanese-American who grew up in New York and New Jersey. An 'all-American boy', computers have been a passion since his high school days.

'I feel like an old man in computers,' he laughed. 'I've been at it since I met my first one as a sophomore in high school which is now approaching 20 years. At first I only played games, the game of lunar lander (on a tty, no graphics, NCR Century Basic II on a time-sharing system where the computer was 50 miles away). Then I started programming games and useful teaching tools. Some time around winning a couple of local programming contests, I decided upon it as my career.'

Kuo knew that he wanted to be a programmer; however, *Caltech* (*California Institute of Technology*), his chosen university, did not offer a course in Computer Science at that time – the accepted route for budding programmers was a degree in Engineering and Applied Science.

'But I was an enthusiastic kid at the time. I started by going after two degrees in three years (Applied Math and E&AS). As my enthusiasm waned, or reality set in, this went to two degrees in four years and then to one degree. Because of the front-end accumulation of credits, I had lots of time in my senior year to teach and work.'

First Steps to Success

In 1982, after gaining his degree, Kuo went to work for *Proximity Technologies Inc* (now part of *Franklin Computers*) in Fort Lauderdale, Florida, where he worked on spelling technology – his 'claim to fame' from those days was to write Word Challenge, a computer version of the game Boggle from *Parker Brothers*.

He stayed there two years, and moved in 1984 to *IBM* at Boca Raton (also in Florida) – the plant which created the *IBM PC*: 'It was an exciting period,' recalled Kuo, 'when *IBM* was defining the standards. I worked in AdTech (Research) and then on the PS/2 model 30 (keyboard BIOS, pointing device BIOS, speech BIOS). This is where I gained most of my background knowledge of PCs.'

It was not long, however, before the 'retired' atmosphere of southern Florida began to tell on Kuo: 'I grew homesick for the bustle of Los Angeles,' he explained, 'so I sought a trans-fer to the *IBM* LA Scientific Center. I worked on grammar parsing on mainframes, and on robotic vision with RS/6000s.

'But my heart was with the PC, and I helped out on *IBM*'s internal bulletin board answering questions about *IBM PCs*. This BBS was being run by a group out of *IBM Research*; Steve White's group, with Dave Chess and Bill Arnold as principal participants. Eventually, I collaborated with White and Chess on a paper entitled 'Coping with Viruses'.

'Around the time I was planning to leave *IBM*, Steve was setting up the High Integrity Computing Laboratory, but things did not fall into place, and I went to *Locus Computing*, where I worked on *IBM*'s AIX OS for three years.'

From Symantec to McAfee

His contributions to the paper written with White and Chess were written from a theoretical point of view: his first practical work on viruses did not happen until he left *Locus*. A friend working at *Symantec* told Kuo that the company was looking for someone to work with viruses – 'My route landed me there and I haven't been well since! I don't remember which was the first virus I disassembled, but the first one I saw in action was called Redx (aka Ambulance).'

Kuo spent three years with *Symantec*, but disillusion set in, and in January 1995 he and the company parted ways. He now works for *McAfee*, in northern California – not an ideal situation, in his view, as his wife and two children still live in the south of the state, but he put it thus: 'I left *Symantec* on Joe Wells' heels, because I had no faith in the management team.

'At *McAfee* I run the AV Research group, which does detection and removal. I manage the group which programs the engine for "difficult to detect" viruses and other engine changes. I've recently been promoted – that means I have all the work I did, plus now I have to do more managerial stuff.

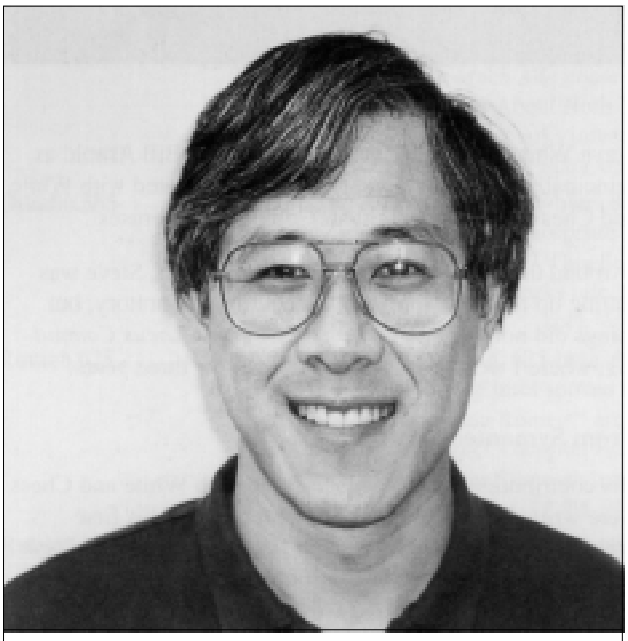
'I probably will stay in this line of work for some time. I've already "broadened" my horizons and that's what benefits me in this position. However, I may be reaching the "those who can't, teach" stage of my career. (And those who can't do that, manage. Hmmmm…)'

Developing Trends

Kuo sees the new word macro viruses as uninteresting in terms of threat, albeit still significant. They have, he feels, been dealt with to the effect that they are no longer an issue.

'The problem for the future,' he explained, 'will be the number of "firsts" that will occur with each new macro environment that might be attacked, but we will have learned from this last half-year to enable us to react even faster; as for instance with *Excel*.'

'One of the least excitable people in the world' is how Kuo describes himself. This leads him to believe that, with good backups and reasonable security, viruses can be viewed as little more than just another problem of which every user must be



Jimmy Kuo: after only three years in the field of virus research, he is already in charge of the AV research group at *McAfee*.

aware: 'One must always take reasonable precautions, though,' he warned. 'A disgruntled employee can wreak a lot of damage, whichever route he takes to attack a system.'

On the Product Front

One line of anti-virus defence which is finding its way into an increasing number of products is heuristics. With the steady growth in virus numbers, Kuo is one of many who see this as part of the solution.

'I am working on heuristics now,' he said. 'I can't talk too much about it, as it's work in progress – but will virus-specific detection remain integral? Yes. People want to know exactly what got them. There is an element of human curiosity involved, but there is also a need to identify potential problems down the road which may be hidden by the virus that attacked.

'I believe a virus scanner must be structured to be able to undergo constant updating, since it must be able to react to circumstances outside our control. *McAfee* was set up to be able to react to changes in the anti-virus environment.'

Regarding Companies

Although Kuo admits that there are in general no new virus-specific products being developed at the moment, nor are there new companies in the field, he pointed out the increase in the development of integrity- and heuristics-based products.

The reason this is happening, he believes, is that it is essentially impossible today to acquire the complete virus library which would be necessary to build a complete scanner: 'It's too late to start,' he said simply. 'At the same time, present scanner products all have or will have heuristics and integrity checking to stave off these new entrants.'

Regrettably, he does not believe that anti-virus software developers will ever cooperate to the extent of exchanging technology: 'We cooperate only in providing each other with viruses in a common interest to protect the computer user population. This is an attribute unique to the industry which I expect to continue.'

The Legal Issues

There is a great deal of controversy over the best way to deal with virus writers; the two most favoured approaches are punishment by law and re-education. Kuo's own firm belief is that legal recourse simply makes society feel good after the event.

'Other means must be employed if we wish to address this as a preventative issue,' he stated. 'Some people write viruses for the challenge; some write them for the experience. Coercion, peer pressure (of the good sort!), or other productive projects should be used to replace the virus writing activity.'

He does, however, possess a strict moral code, and has stated publicly that he will not hire people who acquire their knowledge of viruses through writing them. His hope is that the wide adoption of such a stance will make people consider their position carefully, knowing that if they write viruses, they will be shut out of certain industry posts.

Kuo has never written any self-replicating code, considering that there were already viruses enough to study when he joined the industry three years ago. 'I once pondered the "C program which executes to print its own source code" problem. That might,' he mused, 'qualify as self-replicating, but not as a true virus.'

The Greener Side

Kuo has a great many interests outside computing: 'Too many!' he laughed. 'Many people have known me to dabble in financial markets, and I also enjoy competitive sport. I play in a tennis tournament every month – unfortunately, I also play tennis just once a month!'

He is also proficient in ping pong ('Comes with the blood,' he chuckled), backgammon, bridge, chess, volleyball, and foosball. Another pastime is juggling – he can now keep three clubs or four balls in the air.

His family, too, takes up much of his time: the elder of his two children already enjoys playing on the computer, but is only allowed to do so when her brother is napping – 'Otherwise they battle for the computer!' Although Kuo is keen to encourage this budding enthusiasm, his wife (a chemical engineer) takes the 'slowly, slowly' stance.

It will be interesting to see where the next few years lead Jimmy Kuo: having been in his current field for only three years, he feels it is too early to make long-term decisions about his future. A return to engineering? A Grand Master at chess? A renowned virus researcher? Only time will tell…

VIRUS ANALYSIS 1

Manzon: Threatening Behaviour

Kevin Powis

Manzon is an 'in the wild' polymorphic memory-resident file infector which targets COM and EXE files. Its name is taken from a text string inside the virus body, which is visible after decryption: 'MANZON (c) Sgg1F5PZ'.

When an infected file is executed, the virus takes control before the host code executes, decrypting the main body of its own code. Two sub-routines decrypt the remainder; then the virus is ready to execute. The two sub-routines usually work together to provide double encryption; however, the first routine sometimes simply sets up registers for the second.

On return from the decryption routine, control passes to the next line of code, which now contains a valid instruction. This is a call to another sub-routine which handles the 'Are you there?' call and, if necessary, the installation of the memory-resident part of the virus and the eventual execution of the host.

Are You There?

The virus first calls Int 21h with AX=DCBAh. If it receives DX=DCBAh when it returns, the virus is deemed resident, and Manzon allows the host to run. Otherwise, it installs itself into memory, using standard DOS Int 21h calls.

The next call is made to function 4Ah, which is used to request DOS to extend the current code segment to FFFFh paragraphs. This is bound to fail, but its purpose is to return, in the BX register, the maximum size in paragraphs (16-byte chunks) of the code segment. Manzon then subtracts 1744 bytes from this and shrinks the segment accordingly by repeating the call. The virus has now created a 1744-byte hole in memory.

It next calls function 48h, requesting a new segment of 1728 bytes. DOS obliges, returning to the virus in the AX register the address of its new home. The difference between the bytes freed and those requested is 16. This is not wasted: the programmer seems aware that he must allow DOS to keep 16 bytes in the newly-created segment to hold the MCB (memory control block) header.

Before making use of the new segment, Manzon modifies the MCB, placing in it a reserved marker which makes the block appear to be owned by DOS. This is a convenient lie which will enable the virus to avoid leaving any tell-tale signs, should the user use a memory mapper to investigate the contents of memory. Manzon then copies over 1712 bytes from its body to the new segment.

Hooking

Manzon's next job is to hook the chosen interrupt vectors. It targets only Int 21h: this handler will give the virus control when file activity takes place.

First, DOS function 35h is used to get the address of the current Int 21h handler. If Manzon recognises the data at this address, it pulls out that of the next handler in the chain, using this rather than the value it has, effectively removing this 'known' process from the interrupt chain. I do not know what process Manzon is looking for here: it may be that it is a resident anti-virus product which the virus is disabling.

The virus uses the obtained address to construct a JMP instruction it will use to pass control to the next Int 21h handler down the line in the future. It then uses DOS function 25h to make the Int 21h vector point to its own handler. Once complete, all future DOS file activity must pass through the virus for inspection.

Manzon in Memory

Now the virus is resident, it must allow the host to run. The actions to enable this are different for COM and EXE files. The virus determines file type by examining a word in the virus body: if the value is 100h, which signifies a COM file, the start of the file's image in memory is repaired using the values saved during infection. All registers are set to zero and the virus constructs a return to offset 100h in the current code segment – sufficient to call the host as normal.

For EXE files, the stored header values are used to calculate the program entry point. The registers are set to zero and the virus makes a RET instruction to the start of the EXE code.

The Interrupt Handler

All other functionality of the virus, including its infection routines, is provided by its interrupt handler – this is invoked automatically every time software generates an Int 21h. Each time this happens, the code in the handler checks to see whether the interrupt request is an 'Are you there?' call. If the AX register contains DCBAh on entry, the virus simply loads DX with the same value and returns this to the caller.

If it is not an 'Are you there?' call, Manzon checks the AH register to see if it is a Close File (3Eh) request. If so, the request is interrupted and processing passes further down the handler to determine if infection is required.

In the event that it is not a request to close a file, Manzon next checks to see whether it is a File Execute (4Bh) call. If so, Manzon interrupts the call, uses standard Int 21h calls to open the file, and closes it immediately before allowing the original EXEC call.

When Manzon's Int 21h handler uses an Int 21h instruction itself, it forces the PC to make a recursive call to Manzon's interrupt handler. This time, as the virus has opened and closed a file, it will satisfy its own requirement for the 3Eh close function mentioned above.

When a Close File function is intercepted, Manzon checks to ensure that this handle is not one of the DOS standard file handles; e.g. CON, LPT, PRN. This it does by ensuring the handle value is greater than or equal to five. If the handle value does not pass this test, it is ignored, and control passes down the chain to the next Int 21h handler.

File Infection

By this point, Manzon is seriously interested in this file as a candidate for infection, but the fact that the virus author has chosen to infect on File Close complicates matters. Once a file is successfully opened, DOS gives it a handle (a unique number) and refers to it only by this handle.

However, Manzon needs to know the file name to ensure that it is a program file, not a data file. The answer is to delve within DOS' internal structures, in particular the SFTs or System File Tables, which contain, amongst other things, the file names associated with all open handles.

"Manzon is an example of a file virus which has little trouble surviving and multiplying in the wild"

Manzon navigates the tables and obtains a pointer to the file name. It picks up the first two letters of the file name and encodes them; then goes through a series of comparisons against like-encoded values to ensure that the file does not start with any letters likely to indicate it is an anti-virus package (SC, F-, TB, TO, FV, FI, VI and K-). The last three letters of other file names are tested to see if they contain the letters 'COM' or 'EXE' – if not, they are also ignored.

Although Manzon now has a likely file to infect, another problem presents itself. What if the file was not opened for writing? Manzon's creator again resorts to the SFT, which contains a byte to control the file access mode. Manzon toggles on the Write bit in this field, making DOS think the file was always opened for Read/Write access.

Manzon passes control to one of two routines, depending on whether the file names have the extensions EXE or COM. Modifying the SFT entry will also bypass behaviour blocker software which is monitoring for file writes to program files.

When a COM file is being infected, Manzon obtains its size and reads its first three bytes into a buffer. The virus then makes the sweeping assumption that all COM files start with a JMP instruction and it therefore has the code which makes up this instruction in memory.

It uses this assumed JMP instruction to calculate how far from the end of the file the jump is aimed. If this value is less than 1398 or greater than 1670, the file is still of interest; otherwise it is ignored. This is the virus' self-recognition test: after infection, a COM file receives no future attention at this point.

If the potential host file is more than 62000 bytes long, it is also reprieved. Otherwise, the small three-byte portion of the host in memory is replaced with a JMP instruction to the end of the file, and a freshly-generated polymorphic image of the virus is written to the end of the host file. Manzon now simply writes the patched jump instruction, which will ensure control passes to the virus when the host is next executed, out to the start of the next file.

Infection of EXE files is similar. First, nineteen bytes of the header are read into memory to facilitate infection: this is double-checked to ensure it starts with a valid EXE signature of 4D5Ah or 5A4Dh. The self-recognition test based on the start point from the end of the file is applied to avoid multiple infection. The polymorphic routine is then called to re-encrypt the virus and write it to the file. Finally, the EXE header is amended and rewritten to complete infection.

Summary

Manzon is an example of a file virus which has little trouble surviving and multiplying in the wild. It demonstrates some very advanced programming techniques. With its polymorphic and encrypted code it should be considered a very real threat. It is fortunate that it does not carry a payload or trigger routine.

Manzon

Aliases:	None known.
Type:	Resident EXE and COM file infector.
Infection:	COM files less than 62000 bytes long; any size EXE files.
Self-recognition:	Program start point as defined in EXE header or initial COM JMP in the range less than 1398 or greater than 1670.

Hex Pattern:	No simple hex pattern is possible for files. The following pattern will locate the virus in memory:
	`3DBA DC75 0590 908B D0CF FAFC`
	`80FC 3E74 183D 114B 7403 E95E`

Intercepts:	Int 21h DOS handler.
Trigger:	None.
Payload:	None.
Removal:	Delete infected files; recover from backup.

VIRUS ANALYSIS 2

Green_Stripe: The First Ami Pro Macro Virus

Dr David Aubrey-Jones

The day of the macro virus is set to continue. Following several *Word* macro viruses, another application macro language has been targeted: I am sure it will not be the last.

As macro languages have become more powerful, they have become full programming languages and mini-operating systems in their own right. As such, they support all features necessary to support a virus. They also represent a particular danger over and above traditional viruses: they are easier to write, and are certainly far quicker to modify.

It takes no time to take a macro virus (if it is not encrypted!) and change it in a number of significant ways, so creating a new virus. While macro viruses can be detected using traditional signature scanner techniques, one must question scanners as a long-term solution, as it is so easy to modify macro viruses so that they are no longer detected.

In the Press

Green_Stripe itself is nothing about which to get alarmed. However, the code was published in the USA in an underground hacking magazine produced by *American Eagle Publications Inc* – I fear that this is likely to encourage more people to write macro viruses.

This virus differs in many ways from the *Word* macro viruses, due mainly to differences in *Ami Pro* macro language. Rather than incorporate its macro into the same file as the document, it keeps the two separate, and in the case of Green_Stripe, the macro file (extension .SMM) is always the same length (6256 bytes long), giving it in this sense some similarities with a DOS companion virus.

This creates a problem: to spread successfully, the virus must ensure that the macro file is copied along with the document file. If a file copy is performed specifying the filename using DOS Copy or File Manager, it will not happen and the virus will not spread. This is also likely to be the case if *Ami Pro* document files are sent via email.

In an attempt to overcome this limitation, the virus subverts the 'Save As' menu function, so that when an infected document is saved, a copy of the virus macro is also saved with it, without the user's knowledge.

When one first opens an infected document, the virus macro executes automatically, because it is linked to a document as an automatically executing start-up macro. The user will then witness something strange happening, which should alert him to the presence of the virus.

The virus then searches through the document directory, opening and attempting to infect all files: these will flash up rapidly on the screen, and should alert a user to the virus' presence. In addition, the process is likely to generate error boxes, which pop up if a document is already open.

The file search is performed using the macro commands FindFirst$ and FindNext$. The virus hunts for all documents with the extension '.SAM' (*Ami Pro* default extension). If one is found, a check is then made for an '.SMM' file with the same name, so that the virus can determine whether or not the file is already infected. As an '.SMM' file is normally an *Ami Pro* macro file, Green_Stripe assumes that if one already exists, it must be infected with the virus.

Infect File Routine

Provided no .SMM macro file is found, the virus calls an Infect_File routine. After extracting the file name, it proceeds to assign it to the document as an automatically executing start-up macro. The document is then saved to disk and closed.

The virus now takes the name of the current infected virus macro, works out the name for the document it is infecting with the extension .SMM, and uses the DosCopyFile function to save the macro with the new document name.

Finally, it sets the DOS hidden attribute on the new macro file, a common trick for companion viruses, making it invisible using a normal directory search unless the /A switch is used.

After having attempted to infect all documents in the document directory, it hooks two *Ami Pro* menu functions using the command ChangeMenuAction, 'Save' and 'Save As'. The virus uses this to subvert these *Ami Pro* menu functions using its own macro routines. Whenever the user now attempts to save a file, the virus does its dirty work.

New 'Save As' Dialog Box

When the menu command 'Save As' is accessed, a user should notice that the dialog box generated to input the new name is more primitive than usual: the virus does not attempt to replicate the normal box in detail. However, it is still functional. It will also have the incriminating title 'Macro Get String'.

Once the new name has been entered, the document will be saved under this name as usual. Green_Stripe then ensures that the new document name has the file extension .SAM, and if so it proceeds to open the virus macro file and save it under the new document name entered with the extension .SMM. One task remains: to assign this macro as a Start-up macro to the new document.

Payload

The other menu function which is subverted is called whenever a file is saved. It contains the virus payload, which is extremely simple: it is performed in only three macro command lines.

The virus attempts to move to the first page of the document, and to replace every occurrence of 'its' with 'it's' before saving the file as requested. In tests, problems were experienced in getting this part of the virus to function. The difficulty seemed to lie in a failure to hook File Save rather than an error in the payload SaveFile function.

This must be one of the more unusual payloads to be seen in a virus, and is unlikely to cause many sleepless nights [*depending on your line of work. Ed.*]. However, it would be a simple task to modify it, and make it far more destructive. Imagine the possible consequences if it replaced every occurrence of the number one with a seven, or three with an eight, etc. If this were performed by a virus macro which is attached to a spreadsheet file, the results could be horrific.

Conclusion

Green_Stripe is not a great threat, and is unlikely to spread in the wild. Once one knows what to look for, it is also easy to recognise. In addition, it is easy to perform a clean-up, this being just a matter of deleting the .SMM virus macro files which will be in the same directory as the document files.

Its real significance is as an example or demonstration virus. This is a new platform for viruses, and, as the virus was published in a hackers magazine, I fear others will be encouraged to experiment along similar lines using such macro languages. This makes it a dangerous development.

Green_Stripe

Aliases: None known.

Type: *Lotus Ami Pro* macro file infector.

Self-recognition:

Tests for the presence of an .SMM file with the same filename as the .SAM document file.

Hex Pattern in .SMM Macro Files:

```
7573 2030 2030 2034 2032 0D0A
496E 6665 6374 5F46 696C 6520
```

In addition, the text 'FUNCTION Green_Stripe_Virus' is visible.

Trigger: Saving a document file.

Payload: Attempts to replace all occurrences of 'its' with 'it's' in a document.

Removal: Delete the virus macro files (.SMM) from the document directory.

COMPARATIVE REVIEW

Disk Authorization

Ross Greenberg

Take any random diskette from a workmate's desk drawer. Stick it in the A: drive and reboot. Then run each program on the disk, and read all the *WinWord* documents.

Whoops. You forgot to scan that floppy. You just infected the fileserver on your two-million-node network. You're sure the boss will understand… and your resumé is in good shape, in case he doesn't.

If you feel that the majority of viral infections in your organisation come from sticking a floppy in your A: drive and rebooting, or running infected software on that floppy in your A: drive, then disk authorization software is something you should be using.

Disk Authorization: What is it?

The concept behind disk authorization software is simple: only disks, and the software and data on them, which are approved by an automated system or by a person somewhere should be allowed to enter your system. Likewise, disks leaving your system should only contain authorized software and data and only another authorized system should be able to read them.

In a given organization, there must be some sort of gatekeeper – whether human or machine – by which diskettes are authorized for import into and export out of that corporation. This gateway machine should be the only interface between the nasty world out there and your nice controlled corporate environment.

Enforcing the scheme generally involves making diskettes unusable by machines without the requisite software having been installed or by users without the proper security. This can be done using a variety of methods, but two stand out: to encrypt the data on the floppy, and to make the disk non-readable and non-writeable except by machines authorized to do so.

As a means of enforcing disk authorization further, vendors are also applying the same ideas to the hard disk: by encrypting the Master Boot Record (MBR) or the root directory on the C: drive, they can ensure that the floppy disk authorization software is running and that the user did not boot from an alien disk.

In general, disk authorization and authentication software can be considered as a subset of access control software. There is much on which to control access: some packages allow the system administrator to control access to the printer, or comm ports. Some products allow control of read/write access to the

floppy disk. One package has a zillion options, including full scanning from several anti-virus packages before marking a disk 'Access Allowed'. All this is not required when only disk authorization is desired, however.

The price range of the packages reviewed here is striking; from about US$25 to over US$250 – the quantity pricing has quite a spread, too. Yet they all do much the same thing: control user access to foreign data on diskettes. This article describes the plus and minus points of each package, and should help make the decision a bit easier.

ESaSS: TBFENCE v3.00

TBFENCE, by the folks who bring you *ThunderBYTE* (*ESaSS*), is one of those 'less is more' products. The only thing *TBFENCE* does is disk authorization. But it does a superlative job of it.

Going with their 'less is more' philosophy, the product comes on a 3.5-inch floppy, encased in a CD jewel box, with a fifteen-page manual. Installation was very easy: stick the diskette in the drive, and type INSTALL C:\TBFENCE.

Files will be copied to hard disk and you will then be tossed into the *TBFENCE* program. Selecting INSTALL TBFENCE, you will be asked which machine you are installing; i.e. what type of *TBFENCE* machine you are creating. *TBFENCE* has five types of machine usage:

- Gateway: this configuration can read from and write to both normal and encrypted diskettes, and change their status from one to the other. Useful as the bi-directional interface to the real world.
- Importer: can read from and write to encrypted disks, but can only read (import) 'normal' disks – data internal to an organization stays there.
- Crypto: able to read from and write to only encrypted disks. Likely to be the most useful configuration. Outside data and programs stay out, internal data and programs stay in.
- Reader: can read both normal and encrypted disks, but cannot write to either. Makes a good game machine!
- Secure: cannot write to anything, and can only read authorized disks.

Make the first machine on which you install *TBFENCE* a Gateway machine: I tried it as a Secure machine and discovered I had no authorized disks – including the installation diskette (on a write-protected floppy).

I was only able to de-install the software from that first machine by installing *TBFENCE* on another computer, making that installation a Gateway machine, making a copy of the installation diskette and authorizing it. Then, finally, I could run de-install on the secure machine!

Each installation of *TBFENCE* requires a password, which must be the same for all the machines in the logical corporate group; i.e. the group of persons who would exchange diskettes with each other.

There are a few other options, such as the ability to format a standard DOS diskette normally on a *TBFENCE'd* machine, but that's about it.

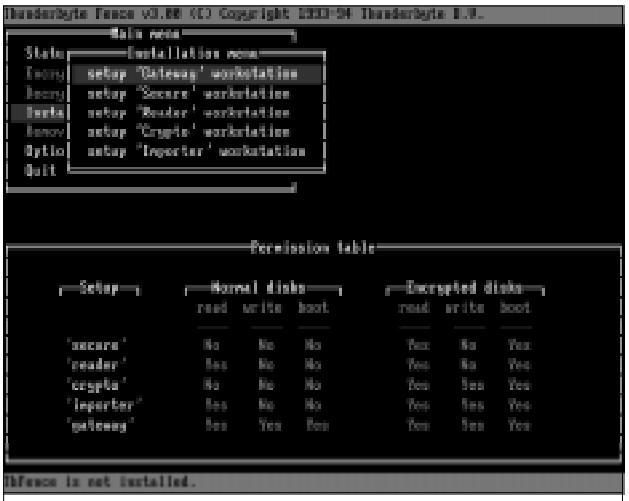
Authorizing and de-authorizing diskettes are simple menu choices from *TBFENCE*'s main menu. It would be a nice enhancement if, for any access to an unauthorized disk on a Gateway machine, an option box were to pop up asking if that diskette should be authorized. Since Gateway machines should only be on desks of supervisory types, there should not be a security problem if the code were to be enhanced in this manner.

Additionally, as was the case with almost all of the products reviewed, there is no real way to authorize a diskette for access on an 'outside-the-security-blanket' basis – that is, assuming a user is sending in a partial database to be examined internally.

The procedure by which this would be carried out would be for a Gateway machine to import the diskette, which encrypts it. The floppy disk is then sent off to a computer which is 'Crypto'-configured. Work is done on that machine and data is written to the encrypted floppy.

That floppy is brought back to the Gateway machine, which exports (decrypts) the diskette – this is a lot of work for a small office. Once again, the system administrator must choose between security and ease of access, and know that these are always mutually exclusive, not just for *TBFENCE*, but for any disk authorization software.

The concept is simple. It works. And the price is right, too: *TBFENCE* is available as shareware, and can therefore be tried out before it is purchased. The pricing begins with a five-user



TBFENCE uses the standard *ESaSS* menuing system for installation: here it displays the various machine types which may be created.

licence, which costs about US$200: this drops down to about US$10 per unit when quantities of 1000 or more are reached.

Developer/Vendor: *ESaSS BV*, Saltshof 10-18, NL-6604 Wijchen, Netherlands

Tel: +31 24 642 2282

Fax: +31 24 642 0899

Email: int.sales@thunderbyte.com

Portcullis: Data Access Defender v4.03

The *Data Access Defender* (*DAD*) module of *Guardian Angel*, by *Portcullis Computer Security Limited*, also has minimal packaging. Perhaps the *Guardian Angel* package itself has more extensive documentation, but this 3.5-inch floppy-in-a-CD-ROM-jewel-case came with only a few itsy bitsy pages of documentation, just enough to describe install and de-install.

Spelunking around shows that this is access control software which has a small piece which does a really spiffy job of diskette authorization.

No further documentation was included in the package which was submitted for review. On-line help was also absent in this version.

Installation was a snap: stick the disk in, type INSTALL, enter the company name and what colour information screens should use and that's it. The software itself is copy-protected: only five installations are allowed per diskette. When the software is de-installed, the count returns. Having copy-protected software in this day and age seems a little misplaced.

DAD has two main programs, GAUSER and GAPW. The first allows the user to be set: *DAD* comes with four users preset with different privilege levels, capabilities and restrictions. The supervisor comes configured as 'GUARDIAN', the privileged user is called 'EXTENDED', and all other users, including the default user on reboot, are minimally privileged.

Another user, called 'FIXUSERS', showed up on the user list, but did not appear to have any special privileges. The 'Add a User' option did not function, beeping that even the GUARDIAN user did not have enough privileges to add a new user.

Each user can be set for a number of different privilege levels, including such items as inactivity time-out/logout, 'virus' protection allowing writes to COM, EXE and SYS files to be disallowed, and whether or not a user should be allowed to use the Control key. If the user is not allowed to hit Control-C, it could make exiting from a startup batch file more complicated.

Each user's access to the printer and comm ports can be controlled – at least at a high level. Writing to the devices with direct port I/O still works, of course.

Administrators can also control users' access rights to the floppy drive (read or write), as well as read/write/rename/delete capabilities on a file-by-file basis (albeit a limited number of files) which can be fine-tuned; locking AUTOEXEC.BAT and CONFIG.SYS from any modification comes to mind. *DAD* also includes audit trails, so that a history of what took place on the system is available for the system administrator.

But what about the main point of the program, at least as far as this review is concerned? Disk authorization. The speed at which diskettes are authorized and de-authorized is not as rapid as it could be, based on the performance given by the other packages.

Whereas *TBFENCE*, above, was pretty quick to complete each task, this code is substantially slower. You can live with it – it takes about 45 seconds for a 1.2MB floppy on a DX4/100 – but it might get annoying when the product is first installed and a bunch of disks all have to be certified. Those seconds add up!

The default user cannot access uncertified disks, and certified disks cannot be accessed by non-*DAD* machines. So, what then is the Gatekeeper on a *DAD*-equipped machine? The more privileged user, naturally. But remember: setting up a Gatekeeper machine with a privileged user logged in for certifying those disks uses up one of your five allowed installations.

If you're looking only for disk authorization software, *DAD* is a bit of an overkill. If you need disk authorization *and* access control, then it is pretty good. The price is certainly good, even for smaller quantities – a single user licence is about US$45. There is not much room to manoeuvre when you start off with a low price per machine, however. It makes one wonder why this software is copy-protected.

Data Access Defender, one module of *Portcullis' Guardian Angel*, offers disk authorization and then some.

Developer/Vendor: *Portcullis Computer Security Ltd*, The Grange Barn, Pikes End, Pinner, Middlesex HA5 2EX, England

Tel: +44 181 868 0098

Fax: +44 181 868 0017

Email: portcul@dircon.co.uk

Reflex Magnetics: DiskNet v4

DiskNet, by *Reflex Magnetics Ltd*, takes a slightly different approach to disk authorization. Whereas the other products reviewed make an authorized diskette unreadable by a 'vanilla' machine, *DiskNet's* default operation merely marks a disk as being authorized and leaves it readable. However, if an attempt is made to read an unauthorized diskette, a warning pop-up box advising the user of this appears and returns with an Int 24h failure.

So, then, what makes a disk authorized with *DiskNet*? By the look of things, a checksum of some sort is done on the disk, or at least on certain key areas of it, the result of which is stored on track 0, sector 0 (replacing the *MS-DOS* label which usually resides there).

Writes by a non-*DiskNet* machine will not update that checksum, so when a *DiskNet* machine next accesses the disk, the checksum will not match the contents of the disk, and it must be re-authorized before access is granted.

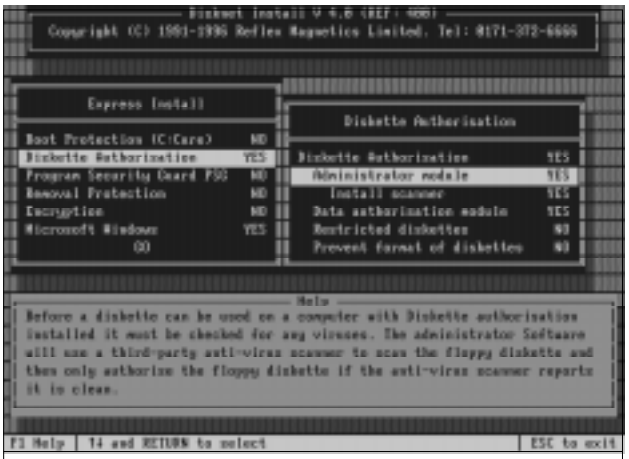
One thing which makes *DiskNet* unique is that it comes bundled with an anti-virus program – in this case *ThunderBYTE* – and additional scanners can be included in the authorization process.

All anti-virus vendors give lip service to the phrase 'it's always safer with more than *just* our scanner' (then try to convince the buyer why theirs is the best): *DiskNet* is no exception, and allows you to check a diskette automatically with a few different scanners before allowing access.

In tests run for this review, some viruses missed by the latest copy of *ThunderBYTE* were picked up by the latest copy of *McAfee's SCAN*; ones missed by *SCAN* were picked up by *F-PROT*, and so forth. As each vendor has its own schedule for updates, having the ability to use several scanners will allow for updates to be included more frequently than provided by *all* the vendors.

Another unique feature of *DiskNet* is that if you are installing only the Disk Authorization module, there is no need to mess around with the MBR or the boot record at all. Disk authorization is accomplished via a small (approximately 8K) TSR which the installation program inserts into the file AUTOEXEC.BAT.

If you do not have an AUTOEXEC.BAT, the installation program will claim this is an error and not create one; also, if in your AUTOEXEC you have an empty PATH statement, the



The installation screen of *Reflex Magnetics' DiskNet*: this product allows multiple scanners to be used to check each incoming diskette.

install program will merrily append ';C:\DISKNET', which presumes there was something on the PATH statement to begin with. The programs used in the authentication module are copied to your root directory on the C: drive – you have no say about this – then marked as hidden files.

As an option, *DiskNet* also contains the more common authorization scheme of making disks exported from a protected system unreadable on other systems. Such floppies are called 'restricted' diskettes. A restricted disk's odd formats are utterly transparent if the TSR is running; if not, its contents are inaccessible. Changing from a restricted to a 'normal' diskette requires a password.

The package also contains some sophisticated access control. From password protection on boot, to protecting EXE and COM files from write or delete access, to 'locking' CONFIG.SYS and AUTOEXEC.BAT files, *DiskNet* tries hard to provide a 'one-stop shopping' approach to access and diskette authorization.

However, unlike *DAD*, which provides for specific access control on a user-by-user basis, *DiskNet* only has two users to speak of; the regular user and the administrator, and they share the same profile. So, for instance, if the ability to format diskettes is turned off for the regular user, the administrator will also have a problem, although he can always change the profile. Separate profiles for each user and the ability to log a user on/off the system would be a useful feature in the next version.

An additional feature which was quite nice was the ability to have a small (4–32MB) encrypted hard disk area. In essence, a separate hard disk springs into existence which in reality is a large file in the C:\ directory. A password is required to access anything on that 'disk', and it may be backed up just like any other file.

The theory is that only a small amount of information need be encrypted, hence the small size of the encrypted disk. Although the encryption algorithm was not examined, the product claims

to use a 64-bit key – as such, US users might well be in the odd position of being able to import *DiskNet* but not being able to export it without violating the US's rather silly arms laws.

An important drawback is that, assuming *DiskNet* can be found sitting on the retailer's shelf, there is no way that the buyer can be sure how old the included copy of *ThunderBYTE* – installed by default – might be.

In light of this, it would be a good idea, immediately after installing the program, to overwrite the copy of *ThunderBYTE* residing in the C:\DISKNET directory with the current TBSCAN, or to install your own favourite virus scanning package. *DiskNet* provides for just about any of the anti-virus scanning packages you may have heard of.

DiskNet is an expensive product, starting at about US$250 and working its way down to a cost per unit of US$20 for quantities greater than 1000. It is a relatively new product, however, and the price tends to fluctuate widely – and wildly – on new products, until the vendor finds its point of equilibrium. *Reflex Magnetics* would do well to look at the competition and see their prices. And please include at least some documentation on disk!

Developer/Vendor: *Reflex Magnetics Ltd*, Unit 1, 31-33 Priory Park, Kilburn, London NW6 7UP, England

Tel: +44 171 372 6666

Fax: +44 171 372 2507

Email: sales@reflex_magnetics.co.uk

Sophos: D-FENCE v3.04

D-FENCE from *Sophos* puts its disk authorization code in the MBR and, like *TBFENCE*, does not try to be any type of access control program. Installation is easy – simply stick the floppy in the drive and type DFENCE. A standard horizontal menu will be displayed: select 'Install on PC' from the Install menu.

D-FENCE is designed to be used by individuals, and by corporate users and corporate security officers, with specific installation for each type of user. After installation, only authorized diskettes may be accessed; attempts to access unauthorized diskettes will result in a pop-up box (with a user-configurable message) and a DOS critical error.

To access a diskette, it must be imported: this makes it unreadable by non-*D-FENCE*-equipped machines. Interestingly, *D-FENCE* does not change the imported disk to a format guaranteed to be unreadable but encrypts the data on the floppy, including the root directory entries, the BPB, etc.

The end result is a floppy with a strange volume name, containing zero files and zero bytes free. CHKDSK shows such a disk as probably a non-DOS disk – and rightly so!

D-FENCE offers the standard easy-to-use menu system, along with good functionality at a reasonable price.

D-FENCE is a little strange in its concept of group security, because of its concept of security as a whole. Strange is not always bad, though.

The basic theory seems to be that the system administrator makes a copy of his original *D-FENCE* diskette for each user. The administrator can then configure each user's copy individually so that certain menu items are greyed out, and are therefore visible but not accessible. In my opinion, it would have been better to remove the disabled items on the menu from view entirely, but that is merely a matter of personal preference. The availability of the following items can be controlled:

- Import a floppy disk
- Export a floppy disk
- Install on PC
- De-install from PC
- Configure (or install) on a device driver
- Remove from a device driver

So, for example, to keep a user only using disks which the system administrator had authorized and not to have the option of taking work home, the system administrator would disable the ability to import or export floppy disks.

The ability to install or remove *D-FENCE* from specific device drivers is a nice option: if there is an external device drive for removable hard disks, a system can be configured to encrypt only that medium – hence you are not restricted to being able to authorise only floppies. As removable hard drives become less costly and more popular, people bringing work home will need a copy of *D-FENCE* for their home machine, too.

Once the system administrator is satisfied with the various security settings, the security menu itself can be disabled, locking these settings down. *D-FENCE* is then installed on the

user's machine (which writes the security MBR) by an unmodified copy of *D-FENCE*, and the modified *D-FENCE* is copied to the user's machine.

This sounds confusing, but is actually easy in practice: the only difficulty is some juggling of diskettes as a new user's machine is *D-FENCE'd*. It is easier still if the end-user does not need a copy or a modified copy of the program: the installed security MBR is enough to prevent unauthorized import or export of floppy disks.

One important field on the Security menu is the ability to enter an ID and a group name. The group name is merely for looks, and is displayed each time the *D-FENCE* program is run. The current group name (retrieved from the configuration file when *D-FENCE* was last exited) is displayed, and the user is asked whether or not that name is correct. A negative answer returns the user to DOS; an affirmative answer drops the user into his copy of *D-FENCE*.

Each authorized diskette is encrypted using the ID as a portion of the key. This means that diskettes imported with one ID cannot be read from or written to by a system with a different ID logged in. If the Security menu is disabled, this effectively locks a user to a single ID and, thereby, a single set of disks: Department A will not be able to access Department B's diskettes.

Of course, this means that a single Security Officer responsible for the security of a compartmentalized corporation has his work cut out. Keeping track of what diskettes are from which department, and acting as go-between for the exchange of diskettes from one group to another (with a forced stop to non-secure 'normal' disks as part of the import/export process) can be tiresome.

The next version of the code ought to include an administrator function for porting diskettes from one ID to another. Perhaps a port in each direction as another menu choice configured for individual users: permit the import/export officer for each group to be able to export disks to another group, etc. That is a minor quibble, however, and only applies to corporations and organizations large enough to have the need for different group IDs.

The pricing of *D-FENCE* is, frankly, surprising. Compared to its competitors, and looking at its capabilities and functionality, *D-FENCE* is underpriced – but don't tell them or they'll raise the price! Starting at about US$30 for single units, there is not much room to play as quantities rise. At quantities of more than 1000, each incremental copy will cost about US$12.

Developer/Vendor: *Sophos Plc*, 21 The Quadrant, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YS, England

Tel: +44 1235 559933

Fax: +44 1235 559935

Email: jbs@sophos.com

Best Buy

Of the four products reviewed, each has its pluses and minuses. For example, the *TBFENCE* program is distributed as shareware so you can try it out before splashing out your hard-earned corporate dollars. Contrast that with the copy-protection scheme of *Data Access Defender*.

Documentation is another matter to consider: *Sophos' D-FENCE* includes their well-written and equally well-received *Data Security Reference Guide*, stickers to let the user know which diskettes have been secured, and a mouse mat, which has nothing to do with disk authorization – but I did need one.

When looking at cost, the quantity sale price is important to consider, too: *DiskNet*, as a diskette authorization package, is simply not worth over US$250 for single quantities (but then, the market for single user sales of disk authorization products is very limited); however, it gets much more reasonable when you buy in large quantities, slipping down to US$20, similar to *TBFENCE*.

"which is the best buy, the most effective product for the best price?"

For the small business user who really does want only a disk authorization package, *D-FENCE* seems a clear winner. A good price, although the installation might take a bit more work than some of the other products.

For the medium to large corporate user who is little concerned about access control, *TBFENCE* is probably the right choice. It is fast, easy to maintain and install, and the concept of the different user types mentioned above would fit easily into the way in which the real world business community uses disk authorization.

For those who want access control with disk authorization, *Data Access Defender* is a bargain (US$45, quantity one; US$15 for 1000+), and makes me want to take a look at the rest of the *Guardian Angel* package, too. The copy protection is a turn-off, though – but paranoia is to be expected in data security software, right?

If your corporate security is good, but people are not doing virus scanning, then get *DiskNet*. When this product is properly installed, you can rest assured that no viruses are sneaking into the corporation, and the disadvantages of a single gateway machine disappear. Remember, though: most of the additional virus scanners you might install for *DiskNet* are likely to be shareware, so the cost of the shareware package(s) registration should be added onto *DiskNet's* base price.

So there you have it: the answer to the question 'Which is the best buy, the most effective product for the best price?' is simple: it depends.

PRODUCT REVIEW 1

VET_NET: A Network Solution

Martyn Perry

VET_NET 1.0 is a *NetWare* server-based package from the Australian company *CYBEC*. This product is the stable-mate of *VET*, the DOS scanner which was reviewed in the December 1995 edition of *VB* [*see that edition, p.20*]. This review will evaluate the features applicable to the server environment. The package was tested under *NetWare 3.12*, but the package is also certified for versions 3.11 and 4.10

Licence Considerations

The software licence is essentially server-only, although two licences are shipped with the product. The licences are:

- *VET_NET* NLM: User Licence Agreement for one server, or a specified number of servers. This covers the number of server licences purchased. The licence is on a per server basis, irrespective of the number of users.
- The Corporate Service Licence agreement: this allows unlimited checking of PCs but does *not* allow permanent installation of *VET* on a PC. This permits a workstation to be scanned for viruses before the software is installed onto the server.

If permanent installation of *VET* on a workstation is required, a separate *VET* site licence can be obtained.

Presentation and Installation

The *VET_NET* package contains installation disks and documentation for the DOS product (*VET*), which is used for the initial checks of the workstation, and the *NetWare* product (*VET_NET*). There is also an excellent book *Viruses & Your PC*, which provides an introduction to viruses and their impact on various parts of a PC.

For *NetWare* v3.x only, CLIB 3.12g or later is required. The CLIB used in these tests is version 3.12j, obtained from LIBUP6.EXE. The documentation includes details of where to obtain the files. Without a sufficiently recent version of CLIB, *VET_NET* will not allow the installation to proceed.

The installation is performed by first checking the installation workstation, using the DOS scanner (*VET*). The workstation installation also gives options to install on individual PC or to install on the server for subsequent installation from the network.

Having established a virus-free environment on the workstation, the installer can now log on to the target server with supervisor or equivalent rights. INSTALL is run from the installation disk in the workstation floppy disk drive, and produces a screen confirming the target directory on the server. Two files, VET_NET.NLM and VET.DAT, are then copied into that directory. The installation program also gives the option to modify AUTOEXEC.NCF to load the NLM at start-up.

Capabilities

The NLM uses the same engine as the DOS and *Windows* versions of the product, and is loaded from the server console prompt using LOAD VET_NET. *VET_NET* has three modes of operation: Immediate (or on-demand), On-access and Periodic.

The Immediate test scans the server on demand, using the current Immediate Test configuration. This can be started and stopped from the opening menu.

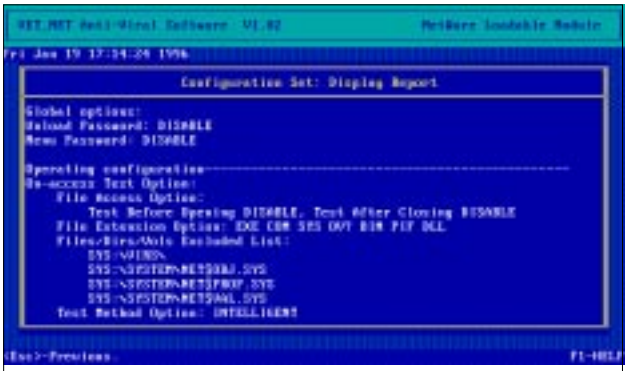
The On-access test can be configured to scan files as they are opened, thereby preventing a user from working with an infected file. In addition, it can test files as they are closed, to prevent a user saving an infected file back to the server.

The Periodic test scans the files on a timed basis: daily, weekly, or monthly, using the defined periodic configuration set. This test can only be stopped by unloading the NLM. Depending on the configuration, unloading may require a password. This option is available for enhanced security, since most scheduled tests would be running out-of-hours, and therefore unattended.

Configuration Options

For each mode of operation, various selections can be made: file extensions to be included in the scan, which volumes to be checked, what to exclude, which test method to use, and the action to take on finding a virus.

The default list for file extensions to be included in the scan is EXE, COM, SYS, OV?, BIN, PIF, DLL. Extra extensions can be added, including the wildcard characters '*' and '?'.



The report file contains details of the configuration used to produce each report file section.

Specified volumes, directories or files can be excluded from the scan. The default files to be excluded are *NetWare* bindery files NET$OBJ.SYS, NET$PROP.SYS and NET$VAL.SYS.

Vet_Net has two scanning methods: Intelligent Test and Blind Test. The Intelligent Test checks only the entry point rather than the whole file: its speed makes it the preferred method. The Blind Test performs a byte-by-byte check on a file, and also identifies inactive viruses (for example, where a virus has attached itself to a file, but has not managed to alter the file so that the virus can execute).

When a virus is found, various actions may be taken. Users in a user list may be notified, and the infected file may be renamed or moved to a quarantine directory, the default name for which is SYS:\SYSTEM\VNLM$VIR.VIR\.

The results of each type of scan can be stored separately in their own log files. SYS:\SYSTEM\VNLM$?.LOG is the default file name; where '?' represents O, P, or I (On-access, Periodic and Immediate test respectively).

The results in the log file are displayed in five sections: From (which test – On-access, Periodic or Immediate – initiated the message), Time (the time and date when the message was created), File (the name of the infected file); Event (the problem discovered in the file), and Action (the action taken by *VET_NET*).

Configuration Sets

The different configurations for the various types of scanning can be pre-defined and stored in one of sixteen configuration sets. One of these can then be retrieved from the set to be the new default configuration.

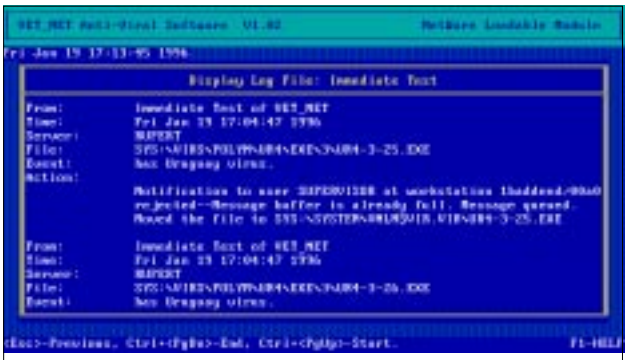
Varying the selections for volumes, directories and files which are exempt from the scan, as well as file types which need to be included, ensures that a full range of scan options and actions can be defined, documented and maintained to meet different needs.

The configuration sets are displayed as a single on-screen report. This can become a problem when paging through the report to find the required set in a group of sixteen. There is an option to hive off the information to another file for permanent record, and the default file in this case is called SYS:SYSTEM\VNLM$CON.TXT.

These sets are particularly beneficial for alternative network administrators to run pre-defined configurations without worrying about the set-up details: they make the required selection from the configuration set to make it the current configuration, then simply let it run.

Administration

The program menus are in the familiar style found in most NLMs. The program uses *Novell* selection techniques for Help, selecting files, etc. The opening menu has the options:



VET_NET's verbose log file format made for large test logs, but the format is very clear.

- Start/Stop immediate test – controls operation of the on-demand scan
- Configuration – creates and maintains configuration sets
- Display log file – displays the result of any virus detected, with action taken and user(s) notified
- Password control – prevents unauthorised unloading of the scanner. There are separate password controls to prevent changes to the various configurations in the menu system.

VET_NET concentrates solely on the server, and has no direct control of workstations attached to that server apart from sending a warning message to a specified user list. There is no inter-server communication, so for a multi-server network, each server must be set up and controlled individually.

There is no display of scan progress, with the exception of a flashing message in the top right-hand corner of the main window. It would be a useful feature for long scans on large servers to show a progress meter, particularly in the case of on-demand scanning.

Detection Rates

The scanner was run against three test sets: In the Wild, Standard and Polymorphic (see summary for all results). The sets were checked in turn with both the Intelligent Test and the Blind Test. The undetected viruses were identified by using the Move to Quarantine Directory option and listing the files left in the virus directories.

The Intelligent Test missed one extra virus from the In the Wild set, Sayha, when compared with the Blind Test.

The tests against the Standard set gave a mixed result. Both scans failed to detect three viruses, two samples of Cruncher and one of Vacsina.634. Additionally, the Blind Test failed to find two more viruses; Liberty.2857.D and one sample of Vienna. Meanwhile, the Intelligent scan failed to find another virus – AIDS, which the Blind Test had found.

The Polymorphic test yielded the same result for both modes of scanning. The main problems were MTZ.4510, which *VET_NET* failed to detect at all, and DSCE.Demo – only fourteen out of 500 samples were detected.

Real-time Scanning Performance Overhead

Although scan speed on a server is usually of less importance than on a workstation, the scanner performed well not only in the intelligent mode (e.g. 89 seconds to check 7454 files on the test server) but also in the slower byte-by-byte Blind mode. However, it is the impact on the overall server performance which is usually of more interest.

To determine the impact of the scanner on the server when it is running, a test was executed, the basis of which was to time how long it took to copy 63 files totalling 4,641,722 bytes from one server directory to another using NCOPY from *Novell*.

The use of NCOPY keeps the data transfer within the server itself, and minimises network effects. The directories used for the source and target were excluded from the virus immediate scan so as to avoid the risk of a file being scanned while waiting to be copied.

Because of the different processes which occur within the server, the timing tests were run ten times for each setting and an average taken. The test was performed under eight different conditions.

The performance difference between having the file access checks enabled and disabled are within the process variability of the server, so leaving them enabled appears to create little additional overhead. The time difference between having the NLM loaded and not loaded is probably due to the difference in server memory available for the NCOPY program to use to buffer the files being copied.

Running the immediate scanner does have quite an impact on server performance: the Blind Test makes this worse. Therefore, users would certainly be aware of a slow-down in performance. That said, because of the speed of the scanner, any slow-down experienced by users would be short-lived.

Conclusion

The documentation is clear and informative. In particular, the functionality and menu system diagrams on the fold-out back cover are useful aids in understanding the options and configurations available.

The installation process is straightforward. The administration uses *Novell*-type menus, and the software can be configured to meet different situations for on-demand, file access and scheduled scans.

It would be useful to have the option to let the scanner wait during periods of high server activity for a pre-set time, thereby reducing the impact on users.

The lack of an hourly timed scan option is surprising, since administrators may want to provide a regular scan of a specific portion of a server during the normal working day; however, the presence of On-access scanning helps offset this. In addition, being unable to have more than one timed scan active at any time means that the administrator must constantly change the configuration from a daily routine to weekly or monthly rather than having all three periods loaded at once.

In terms of the basic functionality, the scanner produced good detection rates for both intelligent and blind testing. Therefore, apart from the merely basic administration features in the scanner, the overall performance of the product meets the objective of handling viruses on a server very effectively.

CYBEC VET_NET for NetWare

Detection Results

Blind Scan		
In the Wild Test-set [1]	274/286	95.8%
Standard Test-set	260/265	98.1%
Polymorphic Test-set	4500/5500	81.8%

Intelligent Scan		
In the Wild Test-set [1]	273/286	95.5%
Standard Test-set	261/265	98.5%
Polymorphic Test-set	4500/5500	81.8%

Overheads

On-access scanning	Copy/s	Overhead
NLM not loaded	11.34	–
NLM loaded, on-access disabled	11.57	2.0%
NLM loaded, on-access enabled	12.42	9.5%

On-demand scanning	Copy/s	Overhead
Intelligent scan:		
on-access scan disabled	48.66	329.1%
on-access scan enabled	46.94	313.9%
Blind scan:		
on-access scan disabled	55.83	392.3%
on-access scan enabled	56.15	395.2%

Technical Details

Product: *VET_NET 1.0 Anti-Viral Software*.

Developer/Vendor: *CYBEC Pty Ltd*, Suite 3, 350 Hampton Street, Hampton, 3188 Victoria, Australia.
Tel +3 613 9521 0655, fax +3 613 9521 0727,
BBS +3 613 9521 6109, email info@cybec.com.au.

Price: Single server licence, AUS$795; 2-10 servers, $AUS395. Other quotes are available on request. All prices include quarterly updates.

Hardware Used: Server – *Compaq Prolinea 590* with 16MB of RAM, 2.1 GB Disk, and *NetWare 3.12*. Workstation – *Compaq 386/20e* with 4MB of RAM, 540 MB Disk, *DOS 6.22*, and *Windows 3.1*.

[1] Test-sets: For details of all the test-sets against which this product was run, see *Virus Bulletin*, January 1996, p.20.

PRODUCT REVIEW 2

A Web of Detection

Dr Keith Jackson

Dr Web is a relative newcomer to the scanner stable. A Russian package, the product can be run either in interactive mode (drop-down menus) or in batch mode (executing a specified scan and returning an error level code). The latter can be used to execute *Dr Web* from AUTOEXEC.BAT whenever a reboot takes place.

Installation

This product is the easiest package I have installed in a long, long time. After using anti-virus packages that draw pretty pictures, spend aeons copying files, and update my system files thoroughly, it is a pleasure to use a product which merely instructs the user to copy all the files contained on the *Dr Web* disk into any desired subdirectory. It really is that simple.

Dr Web was provided for review on a single 1.44MB (3.5-inch) floppy disk. The files on this disk occupied only 373KB – of course, the same amount of hard disk space is required. *Dr Web* is itself a DOS program, and its only concession to *Windows* is the provision of a *Windows* icon and PIF file.

Documentation and Help

The documentation provided with *Dr Web* came in the form of a 69 KB long text file, which was stored on diskette. This file explains the available options in a clear and concise manner. Another file contains all the error messages which can be produced by *Dr Web* – unfortunately, however, these error messages are not explained in the help file. This omission should be corrected.

The product currently claims knowledge of 1450 viruses, and a file provided on floppy disk gives details of the known properties of each of these viruses. This file is very easy to follow – it is somewhat terse, but nonetheless comprehensive. As it is plain text, it can be examined at will for information about any particular virus.

Dr Web includes a content-sensitive help system which can be activated by pressing the F1 key. Although this system is eminently understandable, it is one of the weaker points of the entire package, as the on-line help does not go into any great detail.

I was also underwhelmed to see the following message in *Dr Web*: 'The entire risk as to the quality and performance of the program lies with the user', a disclaimer so sweeping and generalised that, in many parts of the world, it is almost certainly illegal.

Options

Dr Web can be tailored in many ways to provide a specific scan. One selection screen offers expanding windows, mouse support, various display options, font changing, even a language change option. Currently, *Dr Web* can operate either in Russian or in English – the default for the copy provided for review was English. This proved useful; my Russian is rather rusty!

For each scan, it is possible to enable or disable the memory test, the boot sector test, the report log, various scanning selections, and heuristic analysis. The third (and final) setup screen will permit selection of the path to be scanned, the log file name, and checking for a TSR virus – this is jargon for testing whether the size of a file has altered after it has been opened.

Heuristic Scanning

Dr Web offers three levels of scanning, collectively known as 'heuristics'. This is a term which is used by many anti-virus products when they use virus-non-specific tests to decide whether or not a program is infected. Simply put, the product examines the file to see whether or not it contains code which appears to be in some way virus-like, or is a standard uninfected file.

This means that, even if the heuristic testing finds what it thinks is an infected file, it does not know which virus is causing the infection. Heuristic methods also increase the prevalence of false alarms; however, they do enable products to detect some viruses which are not specifically known to that software.

Dr Web offers these three levels of heuristics: 'minimal', 'optimal' and 'paranoid'. The higher the level of checking, the slower the product (see Operation, below). When the 'paranoid' mode is used, one of the additional checks *Dr Web* performs is an examination of a file's date and/or time stamps. This can be an indication of virus infection: for example, some viruses set

Dr Web displays the progress of a scan onscreen in the traditional manner.

certain fields of the time stamp to illegal values as a flag indicating that the file is infected.

Test-set

Once again, *VB* has provided me with a shiny new virus test-set. It has been only three months since the last upgrade, but things are obviously moving apace if we need to upgrade the test-set with such frequency.

The content of the test-set is listed in the Technical Details section [*for a complete listing of viruses in the standard, polymorphic, and in-the-wild test-sets, see VB, January 1996 p.20*]. It includes 5500 polymorphic viruses, a 'Standard' test-set, an 'In the Wild' test-set, and twenty Boot Sector viruses. Even though the total number of polymorphic virus samples has remained of the same magnitude, there are now 500 samples each of several more polymorphic viruses.

Operation

When *Dr Web* is executed, it first scans memory, then the Master Boot Record (MBR), and then whatever particular set of files, subdirectories or drives have been selected.

The first notable occurrence was that when using the default settings, the scan of the hard disk of my test computer took a long time. A *very* long time – 44 minutes and one second, to be precise. The default settings also provided an onscreen listing of all the files which were found to be compressed by programs such as PKLITE, DIET, LZEXE and EXEPACK.

Comparative results of scanning speed are the only fair way to measure how fast a scanner can operate, so I timed how long it took *Dr Web* to scan the hard disk of my test computer with various options selected, and compared these scanning times with two other well-known scanners.

The default *Dr Web* scan time stated could be reduced to 30 minutes 45 seconds by not scanning inside packed/archived files, and further reduced to 20 minutes exactly by switching off all heuristic scanning. Although I tried disabling other options, including memory scan, boot sector scanning, and even the log file, I was unable to reduce the scan of the entire hard disk below 19 minutes 25 seconds. This is, however, unsurprising, as the first two of these are one-time loads at the start of the scan, and the third is insignificant.

By way of comparison, *Dr Solomon's AVTK* could scan the same hard disk in 4 minutes 8 seconds, and *Sophos' Sweep* required 3 minutes 12 seconds to perform the same task. Whichever way this is presented, there is no doubt whatsoever that *Dr Web* is very slow at scanning. Impressively so.

Scanner Detection

Because there are so many options available, the detection capabilities of this product are difficult to express in just a few words.

Dr Web offers limited options for configuring its actions. These may be selected from the interface or from the command line.

Without heuristic detection enabled, *Dr Web* detected only 182 of the 286 In the Wild virus test samples, corresponding to a detection rate of 64%. With 'minimal' heuristics, the total number detected rose to 267, and with either 'optimal' or 'paranoid' heuristics, the score reached a total of 270, a detection rate of 94%. Detection of In the Wild viruses reaches high levels only when heuristic detection is used.

When *Dr Web* was tested against the viruses in the Standard test-set, it detected just 83 of the 265 test samples without heuristic detection enabled. This gives a detection rate of merely 31%, by no stretch of the imagination a high score.

When heuristic detection was enabled (at any of the three levels), *Dr Web* detected 238 of the 265 viruses (90%). Even more so than with the In the Wild test-set, heuristic detection is needed to provide a high level of virus detection.

Without heuristic detection, *Dr Web* detected only fourteen of the twenty boot sector test viruses: it missed Da_Boys, Peanut, Quox, Ripper, She_Has, Unashamed, and Urkel. All these boot sector virus test samples were spotted as 'suspected for infection' when optimal heuristic scanning was enabled.

When tested against the polymorphic virus samples, *Dr Web* performed very well indeed. It was 100% perfect against ten of the eleven sets, and missed only fourteen of the of 500 DSCE.Demo test samples. This adds up to 5486 of the 5500 polymorphic test samples detected correctly, an overall detection rate of 99.7%. *DialogueScience* states that this oversight has now been corrected.

This result is impressive, and makes the product in that respect one of the best I have tested. The score was achieved without resorting to any of the heuristic detection options.

When the 'paranoid' level of heuristic scanning was used, the polymorphic detection rate rose to 100%, as *Dr Web* then spotted the final fourteen polymorphic viruses as having a 'strange creation time'. That is not an explicit virus detection, but it would be enough to alert a user that something odd was afoot.

Dr Web slowed down enormously when the polymorphic virus samples were scanned. The time taken to perform the first scan of the Magneto-Optical disk containing the complete virus test-

set listed in the Technical Details section was 8 hours 44 minutes. When heuristic detection was used, this increased to a maximum of 10 hours and 50 minutes. *Dr Web* was taking so long to scan the entire test-set that I had to run tests overnight for four consecutive nights.

Rather curiously, the highest level of heuristic detection ('paranoid') was actually faster than the other two levels of heuristic detection. It scanned the test-set in a time which was only five minutes longer than having no heuristics enabled. I know not why.

The onscreen reported times were also intriguing. Whilst a scan time of 7 hours 37 minutes and 14 seconds was correctly displayed as H7:37:14, the scan time of ten hours 49 minutes 56 seconds was shown as H0:49:56. Once again I have no idea why, but would suggest the developers test *Dr Web* against a very long scan time. A bug is lurking there.

Miscellanea

Dr Web maintains a log file which contains details of anything has been found during a scan. Along with various other options, the name of this log file can be chosen at will.

There is no memory-resident software provided with the product: it is therefore imperative either to scan manually, or to place *Dr Web* in AUTOEXEC.BAT, so that a scan is performed every time the computer is rebooted.

Dr Web provides an option to 'Cure' an infected file, but in common with my usual stance, this has not been reviewed. Infected files should be replaced with copies known to be uninfected. Doing anything less is just playing games.

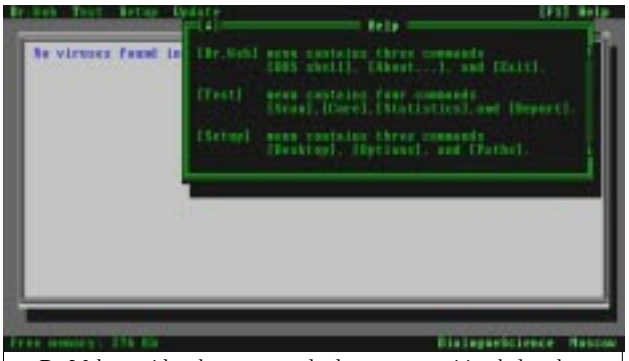
Conclusions

As long as heuristic scanning is enabled, *Dr Web* is quite competent at detecting viruses. However, it has a basic lack of knowledge of many (most?) of the viruses in the In the Wild and the Standard test-sets, and needs its heuristic options to raise the detection rate to reasonable levels.

This need to resort to heuristic detection means that *Dr Web* does not always know which specific virus has caused a particular infection. *Dialogue Science* states that its reasoning for this is that *Dr Web* is usually sold in tandem with its product *Virus Hunter*, which provides for faster detection of the more standard viruses.

When it comes to detecting polymorphic viruses, *Dr Web* is excellent. It requires no heuristic detection to detect these, and gets as close to 100% perfection as is reasonable to expect of any product. A performance as good as this with such reliance on heuristic scanning is impressive.

The main drawback with *Dr Web* is that it is very slow at scanning. The most recent *Virus Bulletin* comparative review found that, of all products tested, it was the slowest: I can only agree with this result. Contrary to expectations, it may well be that resolving the speed problem will be more difficult for the developers than the addition of specific information about more viruses to *Dr Web*. Both of these tasks, however, need to be done.



Dr Web provides the now standard context-sensitive help when the F1 key is pressed – this is helpful but incomplete.

Technical Details

Product: *Dr Web*.

Developer/Vendor: *DialogueScience Inc*, Room 103a, 40 Vavilov Street, 117967 GSP-1, Moscow, Russia. Tel +7 095 938 2970, fax +7 095 938 2855, BBS +7 095 938 2856, email: antivir@dials.msk.su, FidoNet 2:5020/69.

Availability: Not stated.

Version evaluated: 3.08.

Serial number: None visible.

Price: *Dr Web* can be purchased separately as a stand-alone program, or as an integral component, along with three other anti-virus programs, of the *DialogueScience Anti-Virus* kit (*DSAV*). *Dr Web* itself is available as a one-off purchase or as an annual subscription.

One-off purchase:	2-4 users	US$5
	5-25 users	US$10
	26-100 users	US$20
	101-500 users	US$40
Annual subscription:	2-4 users	US$40
	5-25 users	US$80
	26-100 users	US$160
	101-500 users	US$240
	101-500 users	US$320

Larger licences are also available; prices on request.

Hardware used: A *Toshiba 3100SX*; a 16 MHz 386 laptop computer with one 3.5-inch (1.44MB) floppy disk drive, a 40MB hard disk and 5MB of RAM, running under *MS-DOS v5.00* and *Windows v3.1*.

Viruses used for testing purposes:

Where more than one variant of a virus is available, the number of examples of each virus is shown in brackets after the virus name (if the total is greater than one). For a complete explanation of each virus, and the nomenclature used, please refer to the list of PC viruses published regularly in *VB*.

The boot sector test-set contains twenty boot sector viruses, one each of: AntiCMOS.A, AntiEXE, Da_Boys, Empire.Monkey.B, EXE_Bug.A, Form.A, IntAA, Jumper.B, Junkie, Natas.4744, NYB, Parity_Boot.B, Peanut, Quox, Ripper, Sampo, She_Has, Stoned.Angelina, Unashamed, Urkel.

The Polymorphic, the Standard, and the In the Wild test-sets are listed in detail in *Virus Bulletin* January 1996 p.20.

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, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email editorial@virusbtn.com
CompuServe address: 100070,1340

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA

Tel +1 203 431 8720, fax +1 203 431 8165

END NOTES AND NEWS

***Symantec Corp* has released the technology in its anti-virus software** engine to other developers by giving them access to application programming interfaces (APIs). These will allow companies to incorporate anti-virus scanning into other types of applications which run on *Windows 3.1*, *Windows 95*, and *Windows NT*.

On 15/16 April, and 13/14 May 1996, *S&S International* is presenting **Live Virus Workshops** at the *Hilton National* in Milton Keynes, Buckinghamshire, UK. The two-day courses cost £680 + VAT. Details from the company: Tel +44 1296 318700, fax +44 1296 318777.

First.Base is hosting a series of IT security and Internet workshops in Sussex, UK, throughout the next four months. Sessions will include **Internet security (incorporating defence against viruses)** and disaster contingency planning. Information can be obtained from *First.Base* on Tel +44 1903 879879, fax +44 1903 879274.

Infamous computer hacker **Kevin Mitnick has pleaded guilty** in a Los Angeles federal court to using a third party's cellular phone to make unauthorized calls. In exchange for cooperating with police on the case, twenty-two further charges against Mitnick have been dropped.

The next rounds of **anti-virus workshops presented by *Sophos Plc*** will be held on 27/28 March and 22/23 May 1996 at the training suite in Abingdon, UK. Cost for the two-day seminar is £595 + VAT. Any day (day one: Introduction to Computer Viruses; day two: Advanced Computer Viruses) can be attended at a cost of £325 + VAT. Contact Julia Line on Tel +44 1235 544028, fax +44 1235 559935, for details.

***McAfee Associates* has reported record results** for the fiscal year ending 31 December 1995; this despite writing off approximately US$12 million related to its acquisition of various other software companies. For detailed information, contact the company in the UK on Tel +44 1344 304730.

SecureNet Technologies Inc has announced the release of version 2.0 of its anti-virus security software, *V-Net Gold*. The program claims to **protect PCs from viruses received via the Internet or on floppy disks.** Each copy of the product is bundled with the well-known scanner *TBScan* from *ESaSS*. Further information is available from the company on Tel +1 206 776 2524, fax +1 206 776 2891, email info@securenet.org.

Reflex Magnetics has released a new version of its disk authorization package. *DiskNet v4.0* includes such features as hard disk encryption and enhanced virus protection. The company has also scheduled **Live Virus Experiences** for 12/13 June and 9/10 October 1996. Information on the two-day courses, and on *DiskNet v4.0*, is available from Rae Sutton: Tel +44 171 372 6666, fax +44 171 372 2507.

IVPC 96, the ***NCSA's* fifth conference on virus issues**, will be held on 1/2 April 1996 in Washington DC. Speakers will include *VB* editor Ian Whalley. Information from the *NCSA* on conference@ncsa.com.

***Windows 95* protection** is the latest feature to be added to *IBM's* anti-virus software, which provides coverage for multiple operating systems in one box. Information on the package is available from Andrea Minoff on Tel +1 914 759 4713, email minoff@vnet.ibm.com.

The *Computer Security Institute's* ***NetSec 96***, scheduled for 3–5 June 1996, will focus on security issues, problems, and solutions in networked environments. For information, contact the *CSI* by email at csi@mfi.com, or Tel +1 415 905 2626, fax +1 415 905 2218.

SecureNet 1996 will be held at the London Olympia hotel (UK) from 30 April–2 May 1996. Topics covered will include **network viruses, firewalls, risk assessment, and email security**. For more information, contact Alex Verhoeven on +44 1865 843654, fax +44 1865 854971, email a.verhoeven@elsevier.co.uk.