ISSN 0956-9979 NOVEMBER 1995

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Winword again.** Following hot on the heels of our report on the first WordMacro virus comes an analysis of a second such virus, Nuclear: turn to p.8.

• **A bluestocking conference.** The *VB* team has just returned from Boston, where one of their most successful conferences ever took place. The full report begins on p.16.

• **Detecting a new way.** *RG Software* has released a new product which claims to detect any and all boot sector viruses. See how the product fared, from p.21.

CONTENTS

EDITORIAL

I could tell you, but then I'd have to kill you

Regular readers of this column will probably have noticed that I have a certain tendency to write about *Microsoft* with what may appear to be excessive frequency. Why should this be? Perhaps I bear some historic grudge against this company? Perhaps I was in line to rule the PC roost until that nice Mr Gates came along? Perhaps I am simply jealous of a man who, even back in 1990, was worth a cool three thousand million dollars? Well, no – none of these are true. Honest.

"The concept of an NDA is anathema to this spirit"

The reason is, as the Chinese curse puts it, we live in interesting times. Not only that, these times are, like it or not, being driven by *Microsoft*. There is a lot happening. *Windows 95* is now with us, bringing with it all its opportunities, and of late we have the intriguing new field of the macro virus opening up, currently centred around *Microsoft Word*. It is this latter which at present occupies my mind, and the minds of many others.

The phenomenon of the macro virus is proving a tricky problem for anti-virus researchers. In principle, detection of such creatures is not a problem even for the conventional scanner. The DOS/*Windows* scanner is running outside the system under which the virus operates (*Microsoft Word*), so any attempts by such viruses at stealth will not work. The viruses are trivial both in terms of their functionality and in terms of their appearance within the binary document files.

So, where does the problem lie? It lies with the information. Specifically, the information required to locate the macros within the document on disk. Without this, speedy and accurate searching for these new viruses is considerably harder; with it, it is possible for the scanner to go straight for the areas of the document in which the macros reside, and find them quickly and reliably.

Obtaining documentation on this subject is not easy. Give it a try if you have a month to spare – phone up your local *Microsoft* office and ask. It's great fun, if you like hold music. To be fair though, the goodies in this area have not been entirely withheld by the folks in Redmond. The format of modern document-types, such as *Word*, are non-trivial to say the least, and what the anti-virus industry wishes to do is not something that could have been anticipated six months ago.

Even after such information is obtained, there is a second problem. This, like so many, is concealed by an acronym – NDA. Non-Disclosure Agreement. Such an agreement is a mechanism by which a company can keep its secrets, whilst still telling people whom they consider have a need to know.

Suppose you are a large software house, and you want to commission my company to write a viewer for the files generated by your new wonder-product, *WidgetDesign™*. At the same time, of course, you don't want any other companies to know what you will have to tell me, otherwise one of them may come up with *WidgetHack*, a cheaper, smaller, more efficient Widget creation tool which is file-for-file compatible with *WidgetDesign*. In this situation, you get me to sign an NDA. This states that I may not discuss the information I am obtaining, or insights gained directly from that information, with anyone outside of our two companies.

This is an interesting concept to the normally voluble members of any programming community. Hackers, and I use the word in the traditional sense without implying negativity, are a talkative lot. They like to discuss what's being done and how to do things, and the anti-virus community is no exception. The concept of an NDA is anathema to this spirit, and to the oft-quoted 'information wants to be free' ethic. Whilst this latter phrase is both over- and mis-used, it would nonetheless be nice to believe that it still has some substance.

The anti-virus community is startling, above most others, for the level of technical cooperation which goes on within it – clearly there are limits, but these are set higher than one might expect. All NDAs can do is to stick oars into this flow of communication. However, as we move into still more interesting times, the problem of NDAs and general lack of information is bound to reappear. It will be with different systems, even different companies, but inevitably it will happen again.

NEWS

Shipping Viruses

This month has seen two more incidents coming to light of computer viruses being mass-shipped on floppy disks.

The first came from *Digital Equipment Corporation*, and was given to delegates at the *DECUS* conference held in Dublin during the second week of September 1995. The disk, which contained white papers concerning *Digital's* product strategy, was discovered also to be carrying the *Microsoft Word* virus Concept [f*or an analysis, see VB, September 1995, p.8*].

Digital has since distributed to their customers both clean copies of the documents and the *Microsoft Scan* tool to remove the Concept virus. They are also offering a Software Hotline on +353 91 754029 (08:00–16:00 UK time).

In a separate incident, *PC Magazine* in the UK distributed the Sampo virus on diskettes which were sent out to advertise their 'Editor's Day' at the end of October. This incident is made all the more ironic by the fact that, in the same month, the magazine published a review of anti-virus NLMs. *PC Magazine* has since shipped an alert, along with an anti-virus utility to detect and remove the virus, to recipients of the infected diskette ▮

Big Fish, Little Fish

McAfee Associates has announced the acquisition of two companies in the UK. The integration of *Saber Software* with *McAfee* has heralded plans for the launch of a dozen new products within the next year, and will culminate in a family of enterprise-enabled systems management tools for PC LANs.

Bill Larson, President, CEO, and Chairman of *McAfee*, said: 'The combination of our companies and product lines will create a best-of-breed family of highly integrated point products and suites.'

Following the acquisition of *Saber*, *McAfee* has also announced the purchase of *IPE*, which was until now *McAfee's* exclusive agent in the UK.

Peter Watkins, VP of International Operations at *McAfee*, had this to say of the deal: 'According to a recent report from *IDC*, *McAfee* has a 76% worldwide market share for desktop anti-virus software for our *VirusScan* and *NetShield* products. Now with a secure European base, we will be looking to expand our activities in Europe and establish *McAfee* as the vendor of choice for any user investing in quality network security products.'

IPE's subsidiary, *International Data Security* (*IDS*), will remain independent, and continue to market and sell the entire *McAfee* product range ▮

Prevalence Table - September 1995

Virus	Incidents	(%) Reports
AntiEXE	35	12.4%
Form	31	11.0%
Parity_Boot	26	9.2%
Ripper	19	6.7%
NYB	15	5.3%
Empire.Monkey.B	14	5.0%
Sampo	14	5.0%
AntiCMOS	12	4.3%
Concept	12	4.3%
Junkie	12	4.3%
EXEBug	10	3.5%
Telefonica	7	2.5%
Stoned.Angelina	6	2.2%
Cascade.1701	5	1.8%
Jumper.B	5	1.8%
Natas	5	1.8%
Manzon.1414	4	1.4%
She_Has	4	1.4%
Stoned.NoInt	4	1.4%
Barrotes	3	1.1%
Helloween	3	1.1%
Stoned.Manitoba	3	1.1%
Stoned.Michelangelo	3	1.1%
Stoned.Standard	3	1.1%
Byway	2	0.7%
V-Sign	2	0.7%
Other *	23	8.2%
Total	282	100%

* The Prevalence Table includes one report of each of the following viruses: Boot.437, BootEXE.451, Bye, Empire.Monkey.A, HideNowt.1741, Istanbul, Italian, Jackal, Jimi, Joshi, Leandro, Lixi, Print_Screen_Boot.A, Quicky.1376, Quox, SMEG:Pathogen, Stoned.Kiev, Stoned.NOP, Stop.1045, Tai-pan, Tequila, Urkel, UVscan.

Stop Press

Just as *Virus Bulletin* goes to press, there is more news breaking concerning *Microsoft Word* viruses. The latest such creation was posted to the Usenet newsgroup alt.comp.virus during October 1995, and has been named Colors by researchers. It is non-destructive, the only trigger being to randomise the *Windows* colours. The remaining techniques used by the virus appear to be fairly standard, and it is encrypted (as is Nuclear) using the internal *Word* macro encryption technique ▮

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 October 1995. 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

Army_Boots

CR: An appending, 411-byte virus, which modifies the contents of AUTOEXEC.BAT. It contains the plaintext strings: 'C:\AUTOEXEC.BAT' and '@ECHO din mamma har paa sig arme stoevlar!'.

```
Army_Boots      B80D F0CD 2181 F90D F074 558C D848 8ED8 33FF 8EC7 803D 5A75
```

CK.777

CN: A prepending, 777-byte, direct infector, infecting three files at a time. It contains the encrypted text: 'The China Syndrome Version 1.00a Written by Crypt Keeper Well, I guess you found the sectors...You got a warning...This program was written in the city of Cincinnati. Non-destructive version -A- l8rd00d'.

```
CK.777          E8AA FFBB 0010 0E07 B44A CD21 0E07 BB00 10E8 D9FF A31C 00BB
```

Crazy_Frog

CER: An appending, encrypted, 1417-byte virus with the text: 'cRaZy fROG, (c)95 by iRASCiBLE'.

```
Crazy_Frog      8B96 6E05 2E8B 8670 052E 3114 2E31 4402 83C6 04E2 F4C3 E440
```

DigPar

CR: A polymorphic virus, about 1000 bytes long, which contains the text: 'The Digitised Parasite: Australian Parasite [AIH]' and 'Weiners XOR machine 1.0 (c) Australian Parasite [AIH] June 1994'. The pattern below detects the virus in memory only.

```
DigPar          B43F B903 00BA B503 CD21 89D6 81C2 9856 3914 746E B802 4233
```

Ebola

ER: A polymorphic, 3000-byte virus which often causes system crashes. It contains the text: 'Ebola virus 1.2! Extremly stealthmutating system! Technical infos: No way to detectFucked heuristicsGreets go to allvirus detelopinggroups in Brno ! Czech republic94'. It is not likely that we will see this virus spread widely. The template below detects it in memory.

```
Ebola           9C3D 004B 746A 80FC 4074 8D3D E4F7 7447 3D2F C974 4A80 FC4E
```

ExeHeader.265

ER: A stealth, 265-byte virus which inserts its code into EXE headers. The virus hooks Int 13h and infects files when they are read. It contains the text: '[Dying_Oath] by Retro'.

```
ExeHeader.265   8B07 354D 5A74 1126 803F EB75 4426 817F 5CB4 0D74 2EE9 3900
```

H8

CR: A prepending, 1773-byte virus with stealth capabilities. It contains the plaintext strings: '[H8YourNMES]' and 'xtf-ndivskavcommand'.

```
H8              B4FF CD21 C706 0601 EB01 0BC0 7507 EB01 80B4 FECD 21E8 4003
```

Horsa

CN: An appending, 1185-byte direct infector which uses direct disk access (Int 25h/Int 26h).

```
Horsa           AA1E E800 0058 2D12 0033 D2B9 1000 F7F1 0BD2 7403 E98B 038C
```

Kela

CER: An appending, stealth, 2018-byte virus. All infected files have their time stamps set to 62 seconds.

```
Kela            B8FF FFCD 210E 1F8E C0BF 0001 8BF5 B9E8 03F3 A61F 0775 03E9
```

Lady Death

CER: A polymorphic, appending virus, approximately 2744 bytes long, containing the text: 'Lady Death: Dark Fiber [NuKE]' and 'Stainless Steel Armadillo'. The virus corrupts EXE and some COM files. The template below detects it in memory.

```
Lady Death      39F0 5E75 263D DF2E 7504 B864 9FCF 569C 50BE 4A0A FC2E AC2A
```

Leda

CR: An appending, 820-byte virus with the following encrypted text (displayed from 6–11 November): 'Masz wirusa LEDA (BDv3.0), (c) BD 27.V.1994', 'PS Dzieki dla autora wirusa FLOOR 1153'.

```
Leda            B8BD 57CD 2181 FB14 BD74 22B8 2135 CD21 895C 678C 4469 832E
```

Manzon

CER: A polymorphic, appending virus, circa 1400 bytes long, which contains the text: 'MANZON (c)'. The template given detects it in memory.

```
Manzon          3DBA DC75 0590 908B D0CF FAFC 80FC 3E74 183D 004B 7403 E95E
```

Merci

CO: An overwriting, 308-byte virus with the encrypted text: '.COM *.C* CHKLIST.MS ANTI-VIR.DAT'. When the virus infects a file it displays this message: 'Merci virus infected: <filename>'.

```
Merci           E803 00EB 3990 BE3E 018B FEB9 F600 AC32 0639 01AA E2F8 C3E8
```

Mirage.1331

CER: A 1331-byte virus with stealth capabilities. It appends itself to EXE files, but prepends itself to COM files. The virus contains the plaintext strings: 'Mirage' and '\COMMAND.COM'. The time stamp of all infected files is set to 62 seconds.

```
Mirage.1331     80FC  FA75  4B5F  5F3C  0374  15BF  0001  5751  BE33  06B9  CCF9  F3A4
```

Monica.885

CR: An appending, encrypted, 885-byte virus which contains a dangerous payload. The virus sets and activates the CMOS password with the option to verify it at both CMOS setup and PC bootup. The new password is set to 'MONICA'.

```
Monica.885      B929  0381  EE38  03E8  0100  155F  2E8A  052E  3004  46E2  FA58  5F59
```

Multiplex.815

CN: An appending, 815-byte, direct infector containing the plaintext strings: 'MULTiPLEX (c) 1994 Metal Militia\ Immortal Riot, Sweden', 'Somewhere, somehow, always :)*.com', 'IRUSES', 'ImRio'.

```
Multiplex.815   E800  0058  2D0A  01E8  9502  E814  03E8  2402  B447  B200  568D  9CED
```

NRLG.755

CR: An appending, stealth, encrypted, 755-byte virus; the shortest member of the NRLG family. It contains the text: '[MuTaTiOn INTERRUPT] 1994 - Thanks to N.R.L.G. - 800 LIMO 1-800-972-7117'.

```
NRLG.755        F303  8DBE  3301  BA01  00F6  15FF  05F6  1547  47EB  0590  B44C  CD21
```

NRLG.824

CR: An appending, encrypted, 824-byte virus with stealth capabilities. It contains the text: '[MuTaTiOn INTERRUPT] 1994 - Thanks to N.R.L.G. -AZRAEL800 JEWELRY 1-800-346-7231'.

```
NRLG.824        BA01  0080  35E5  FF05  8135  E41B  FF05  F715  802D  4F80  35AC  812D
```

NRLG.853

CR: An appending, stealth, encrypted, 853-byte virus containing the text: '[MuTaTiOn INTERRUPT] 1994 - Thanks to N.R.L.G. - 800 SEAFOOD 1-800-472-0542'.

```
NRLG.853        5504  8DBE  3301  BA01  00F6  15FF  05F6  1547  47EB  0590  B44C  CD21
```

NRLG.865

CR: An appending, stealth, encrypted, 865-byte virus with the text: '[MuTaTiOn INTERRUPT] 1994 - Thanks to N.R.L.G. - 800 ROOMS 1-800-442-6633'.

```
NRLG.865        6104  8DBE  6001  BA01  0081  354C  C581  2D95  CB80  2DA6  812D  98DB
```

NRLG.872

CR: An appending, encrypted, 872-byte virus which occasionally crashes the system. It contains the text: 'Nemesis 1995 Gooberish'.

```
NRLG.872        6804  8DBE  4701  BA01  00F7  15F7  1581  3575  BE80  35D8  802D  E880
```

NRLG.901

CR: An appending encrypted, 901-byte virus with stealth capabilities, which contains the text: '[NuKE] N.R.L.G AZRAEL' and 'Created by MuTaTiOn INTERRUPT! This Could Have Formatted Your Hard Disk! See +++rus Goobers! 1994'.

```
NRLG.901        8504  8DBE  5F01  BA01  0081  2D6D  1281  35FB  4CF7  1580  3501  8135
```

NRLG.985

CR: An appending, stealth, encrypted, 985-byte virus, which contains the text: '[MuTaTiOn INTERRUPT] 1994 - Thanks to N.R.L.G. - 800 DRUGS 1-800-872-1626'.

```
NRLG.985        D904  8DBE  3301  BA01  00F6  15FF  05F6  1547  47EB  0590  B44C  CD21
```

NRLG.1007

CR: An appending, stealth, encrypted, 1007-byte virus. It contains the text: '[MuTaTiOn INTERRUPT] 1994 - Thanks to N.R.L.G. - 800 NANNY 1-800-443-4411'.

```
NRLG.1007       EF04  8DBE  3301  BA01  00F6  15FF  05F6  1547  47EB  0590  B44C  CD21
```

NRLG.1009

CR: An appending, stealth, encrypted, 1009-byte virus. It contains the text: '[MuTaTiOn INTERRUPT] 1994 - Thanks to N.R.L.G. - 800 FLOWER 1-800-878-1073'.

```
NRLG.1009       F104  8DBE  3301  BA01  00F6  15FF  05F6  1547  47EB  0590  B44C  CD21
```

NRLG.1038

CR: An appending, encrypted, 1038-byte virus with stealth capabilities. It contains the text: '[NuKE] N.R.L.G. AZRAELi!'.

```
NRLG.1038       0E05  8DBE  5901  BA01  0080  3578  802D  95F7  15FE  0581  053E  3DF7
```

Oxan

CR: A simple, appending, 710-byte virus. On every twelfth day of February (12 February) it displays the text: 'Happy birthday Oxan !'. On any other afternoon, during the first 20 minutes of each hour, it displays the current version of DOS using the message: 'MS-DOS Version <current DOS version>'.

```
Oxan            FB9C  3D00  4B75  03E8  0B00  9DFA  2EFF  2E11  00EB  4011  0050  5351
```

OpalSoft

CN: An appending, 683-byte, direct fast infector. It contains the plaintext string: '*.COM OpalSoft 10.3.1994 v1.1 C:\'.

```
OpalSoft        C706  3C02  3412  CD19  B980  00BB  0000  8B87  8000  2E89  8129  FE43
```

V.720

ER: An appending, 720-byte virus which marks all infected files with a time stamp of 62 seconds.

```
V.720           B8FF  FFCD  213D  0001  740B  545A  3BD4  7505  33F6  E825  0058  0510
```

XERAM

CEN: An appending, encrypted, 1663-byte, direct, fast infector containing the text: 'N-XERAM'. It deletes the files \CHKLIST.MS, \SCANVAL.VAL, and \NCDTREE\NAV_._NO. The payload, which triggers on any Friday the 13th, includes overwriting 255 sectors on a hard disk if the country code is France, US, Japan, Taiwan or Germany.

```
XERAM           B904  0333  F6A1  3E01  3104  4646  81FE  2E01  7504  81C6  7800  4975
```

INSIGHT

Once a Researcher…

There is a farm in upstate New York which is avoided 'like the plague' by strangers to the area: there are signs posted on the boundaries that warn of live viruses on the property. The farm is Virus Acres; it is owned by a man who enjoys a joke: Ross Greenberg.

Despite the fact that he has kept a low profile lately, Greenberg is a familiar name to many virus researchers, and the author of *Flushot+* and *Virex PC*. However, the former is now defunct, and the latter no longer one of the major players. So where has he been, and what has he been doing?

Being There

Chameleon is a word one could use to describe this man: change seems to be a constant in his life; from student to media person to programmer to anti-virus researcher, once more to programmer, and who knows where from here?

He comes from what he calls a typical middle class, Long Island, Jewish background. His mother was a dental assistant; his father was an engineer who instilled in Greenberg a passion for seeing how things worked: 'My father made sure that, whatever I took apart, I put together again. I never got the opportunity of throwing things out,' he recalled. 'It was "Keep with it until you put it back together" – and I did!'

This practical childhood did not prepare him very well for a disappointing sojourn at university: 'I went to *Stoneybrook*, New York's state university, to study Physics, mathematics, and philosophy. I never did get around to graduating – in 1978, my senior year, I looked around at what kind of job I could get, and saw that a physicist working at *Brookhaven National Labs* with 15 years of experience and two PhDs was worth about $17,000 a year. So, I took a job with *MetroMedia TV*, a local network, starting at that salary!'

Greenberg's responsibilities at *MetroMedia* lay in setting up PC to PC communications programs to coordinate radio and TV advertising, so the company could gauge how much money they were either making or losing: he stayed a mere eight months, going from there to private consultancy.

'Communications by that time had become a speciality of mine,' he explained. 'There were few people around who could do it. If you had a spell at a thing, you became a specialist. I could charge top dollar, which was sort of fun!'

Flushot: Pluses and Minuses

Gradually, Greenberg began to branch out into more general things, writing programs. He remembers a person who was beta testing one of his products sending him a note: '…from a fellow named Ken van Wyk. That note, which he put up on the Net, said that he was being attacked by – I think he called it a virus; a Lehigh virus.

'I thought that this was really horrible, and that it would affect the on-line community adversely, so I put out a fix; a program called *Flushot* – it was downloaded astoundingly quickly, and I started getting tech support calls. Then, as it became bigger, I put it out as shareware – I think it cost $14.00 all-in – and the next thing you know people are buying it, and making demands. That was in the mid-1980s.

'In those days,' he said, 'there were no scanners. I created a behaviour blocker based on what I was told about the virus. I think *McAfee* was the first to produce a scanner. A fight soon broke out between the anti-virus people over scanners and behaviour blockers. The scanner won, for many reasons, but I think behaviour blockers are more effective. They fight the unknowns – scanners do diddley-squat for unknowns!'

Virex PC

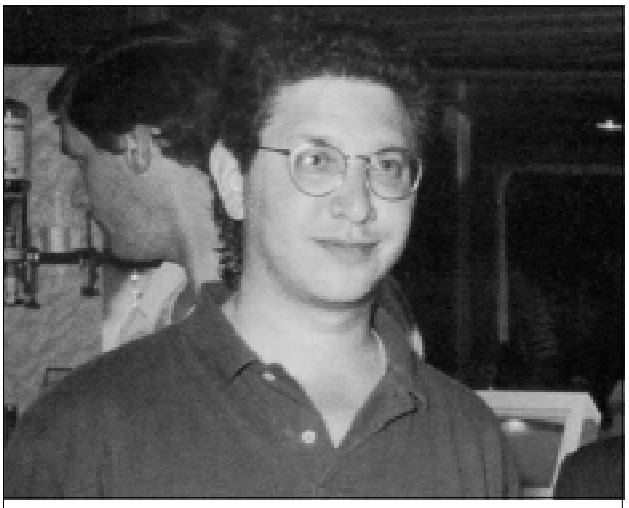
Soon after, Greenberg was contacted by a company called *HJC Software*: they had a *Macintosh* anti-virus product called *Virex* which they wanted to develop for the PC, and believed Greenberg could do it. Dealings with *HJC* were, for Greenberg, less than ideal, and the company sold out to *Microcom*: 'They marketed it into the ground,' he recalled. 'When I threatened to sue for breach of contract, they offloaded it onto *Datawatch*. I think they noticed *Virex PC* still had its head above ground, so pushed it down more.'

'Anything I had to say about the product,' he went on, 'was rejected by *Microcom* and *Datawatch*. They had a distinctive 'Not-invented-here' paranoia which prevented them ever taking suggestions from me. So, Glenn (Jordan, formerly of *Datawatch*) and I would confer and figure out how he could present them in a manner more palatable to their paranoia. He did a wonderful job for *Virex PC*.'

Leaving the Rat-race

Subsequent to this, Greenberg decided to distance himself both from *Virex PC* and the City, and moved to a farm in upstate New York. 'I haven't been doing much virus work,' he said. 'I've been developing telecommunications programs, in particular a shareware product, *RamNet UUCP*. It's a background program that talks to UUCP protocol. They came out commercially at $198.00, but I didn't like the idea of having to do the marketing and advertising, so I dropped the price to $49.00. Commercially, it's meeting my expectations – and they are that I can retire in about a year!

'Since I haven't been so active in the anti-virus world,' he went on, 'it's been interesting to see how short-term people's memories are. I've been out of the picture for three years or so,



Ross Greenberg, author of *Flushot*, *Virex PC*, and *RamNet UUCP*: a man of diverse interests.

and at *VB 95*, I noticed that some people hadn't heard of my products. All the *CARO* members know me, of course, but some of them don't know what I've been doing.'

Carrots and Other Nourishment

Greenberg is still, to an extent, an active member of *CARO*; though, as he stated, there is no membership per se: '*CARO*,' he asserted, 'is a group of people loosely affiliated who share common interests, involving computer viruses and beer drinking! I share my knowledge and expertise with fellow anti-virus people. This is what *CARO* is about. They are more active in the field than me, though – when a new virus comes in, they jump on it straight away… I do it when I get around to it. Often, when that time comes, it's been done!'

Greenberg sees no new techniques in virus writing: 'Polymorphism was one… Interrupt stripping was another… Big deal! The first fifty viruses I tore apart were fascinating, each and every one of them. Of the next couple of hundred, some were mildly interesting, most were boring. The next thousand or so were pretty tedious. The ones that came later – boy, I was glad I was out of the business. Someone had to tear them apart, and I didn't want to.'

Not a single virus, in his opinion, stands out as an exemplary piece of coding, though some he recalls for other reasons: 'DBase was interesting… that was the first virus to screw around with data. Datacrime I remember because I was interviewed by five TV stations, and only one – *CNN* – had the guts to play what I'd said; that it was a non-problem. I didn't get any airtime with the major networks,' he related, 'because I wouldn't say the sky was falling. Unfortunately, media hype has made some vendors extraordinarily rich.'

The Legality of it All

Here in the UK, a young man will soon appear in court for sentencing after having been charged with eleven offences under the Computer Misuse Act. He is charged with writing viruses, and with inciting others to do the same. Could a similar thing happen in the USA? 'There was one person, PhiberOptik, who was sent to jail,' mused Greenberg. 'When he came out, he was a folk hero; everybody celebrated him because he didn't do anything "all that bad". So I don't know if prison is the right idea.

'Maybe a better punishment for that kind of person would be to forbid him ever using a computer again, or for a fixed period of time, and not to allow him to hold a job using computers… I'm not sure how it could be enforced, but being taken away from something he's addicted to would have more effect on the individual than being put in prison.'

Legal redress, he feels, has its place, but only if it is done in a very public way will it have any kind of prohibitive effect on virus writers: 'It's sad,' he said. 'There's this thing called the On-line World, which I loved, and the virus writers were destroying it. It used to be if someone gave you a cool program you didn't have to worry about it … now you do.'

The Next Act

Greenberg thinks that the next new wave of viruses to hit will be OS-orientated; *Windows 95* and *OS/2* viruses which will take advantage of the holes in those operating systems. Indeed, he thinks the only surprising thing about the infamous Concept virus is that it took so long to be released.

The future for detection software, he believes, will not lie much longer with scanners: 'The final solution,' he stated, 'will be a hook in the operating system. Scanners will be very useful for uniquely identifying the virus, but I think they'll be used in conjunction with heuristics. There will also be integrity checking; things like that.'

He feels strongly about the fact that many smaller companies are being swallowed up by the giant conglomerates: 'Competition is good. Seeing new and interesting technology disappear stinks. Companies are bought out,' he explained, 'then the new owners don't want to develop the ideas further, and they are lost forever. Unfortunately, with the best product in the world, if it's not marketed well, it'll be lost. Only the bigger companies have the money to keep their products exposed out there every day. That's where shareware, used properly, can be the great equalizer.'

Although Greenberg is no longer disassembling viruses daily, he still takes an active interest in the anti-virus world, and is considering returning to the fray; however, he is somewhat put off by the antics of certain vendors, whom he sees as less than ethical in their tactics and methods.

In the meantime, life goes on at Virus Acres: Greenberg's seven-year-old daughter has just acquired a brother ('mother fine, child fine, father entirely exhausted!' read the announcement). Whatever route Greenberg eventually decides to take, his expertise and enthusiasm will certainly help to make his task easier, and should he return to full-time virus research, his knowledge and ideas will be heartily welcomed.

VIRUS ANALYSIS 1

A Nuclear Concept: Another Hit for MS Word

Vadim Bogdanov, Andrew Krukov

The era of macro viruses which infect *Word* documents looks set to continue – the second was found on an FTP site in mid-September. Users who downloaded the file WW6ALERT.ZIP became infected with the new virus on reading a *Word* document held within that archive. The ZIP file contains three files: FILE_ID.DIZ, README.TXT, and WW6INFO.DOC. The file FILE_ID.DIZ contains the text:

```
Microsoft Word For Windows Document Virus
Information. URGENT! If you use Microsoft Word for
Windows 6.0 read this to find out about the Winword
virus! Could you be infected with this virus and not
know it? Downloaded from Microsoft CIS.
```

The README.TXT file, also a DOS text file, presents the package as distributed by *Microsoft*. Information on their customer support is included, as is a legal disclaimer from *Microsoft*. The file has the header:

```
MICROSOFT CORPORATION
Customer Information Services.
TO:  Microsoft Word for Windows v6.0 users.
SUBJECT:   Winword Prank virus.
REF: AN98474.
```

The infected WW6INFO.DOC file is in *MS Word* format. Inside, as well as a description of the first WordMacro virus, Concept, are the macros which make up Nuclear. Reading the text about the first WordMacro virus causes the user to catch the second one!

Virus Analysis

The virus macros are encrypted within the infected document. It is not possible to extract the macros for analysis within *Word*, nor to look at them with a DOS viewer. The macros are concealed behind the internal *Word* encryption method. This is crazy. *Microsoft* does not distribute the internal *Word* Document file format, but allows users to encrypt *WordBasic* macros. A clean field for virus writing; a problem for researchers. Fortunately, *Word* does not use complex encryption methods; therefore, after a little work, the macros were decrypted and saved on disk for analysis.

One interesting thing was found while looking through the virus code – the infected document contains the string:

```
C:\40HEX\WW6INFO.DOC
```

Does that mean this virus will be published in the next issue of *40Hex*? This would be bad: more virus writers will turn their attention to *Word*, causing more problems for *Word* users and anti-virus researchers alike.

The Infected Word Processor

While opening an infected file, *Word* executes the file's AutoOpen macro (if this has not been disabled by the standard *Word* macro 'DisableAutoMacros'). Nuclear contains, as did the first-known WordMacro virus, an AutoOpen macro which installs the virus macros into the global macro area (usually NORMAL.DOT).

The AutoOpen macro first checks whether or not the system is already infected by searching for an AutoExec macro in the global macro area. If one is not present, the virus copies its nine macros into this area. These are:

```
AutoExec, AutoOpen, FileSaveAs, FilePrint,
FilePrintDefault, InsertPayload, Payload, DropSuriv,
FileExit
```

After installation, the virus returns control to *Word*. Once installed, the virus macros substitute as follows for standard *Word* functions:

```
FileExit:     exit Word using the 'File/Exit' menu
              command
FilePrint:    print file using the 'File/Print' menu
              command
FilePrintDefault: print file using the Print button on
              the toolbar
FileSaveAs: save file using the 'File/Save' As menu
              command
```

The virus calls one of the trigger routines (the InsertPayload macro) from FilePrint and FilePrintDefault. FileExit simply disables the 'Save filename.DOT?' dialogue box on exiting from *Word* [*Using this option was an initial primitive defence against the Concept virus. Ed*]. FileSaveAs copies the macros into the document being saved, and converts the file to template format (documents being unable to hold macros), as did the first *Word* virus.

The AutoExec macro is executed when *Word* is started. The virus checks the global macros for the presence of the AutoExec macro, and tries to install itself if that is not present. Is there a reason for this? By definition the system is already infected, as the AutoExec macro is currently executing! The AutoExec macro then calls the DropSuriv and Payload macros, which form the trigger routines. The Payload macro is also called when AutoOpen is executed.

First Trigger Routine

The first effect of the virus which the user can recognize is caused by the macro InsertPayload, which is called when documents are printed. The virus checks the system timer, and if the seconds counter is greater than 55 (one time in fifteen), the virus appends the following lines of text to the document being printed:

```
And finally I would like to say:
STOP ALL FRENCH NUCLEAR TESTING IN THE PACIFIC!
```

Because most documents occupy more than one screen, the message will in many cases not be visible during printing – it is placed offscreen. Nice effect, especially while sending files via fax-server (in which case documents are *printed* into the fax-server queue). Nevertheless, this text is the only part of the virus with which this article's authors agree 100%.

More Payloads

The next trigger routine is performed by calling the Payload macro: both the AutoExec and the AutoOpen macros do this during execution. The Payload macro checks the system date, and on 5 April performs some destructive activities. It is clear that the virus is attempting to destroy IO.SYS, MSDOS.SYS and COMMAND.COM. However, it falls victim to both sloppy coding and a bug in *WordBasic*.

The *WordBasic* bug manifests itself during attempts to clear DOS attributes on files – it seems unable to clear the System attribute [*This bug is one of the few that can legitimately be called a feature. Ed.*], causing the subsequent attempt to open the file (in order to truncate it) to fail. The macro exits with an error code, because the attempt to open MSDOS.SYS uses the wrong parameters. At this point, a dialog box appears on the screen, presumably to alert the user.

All the macro has managed to do before it crashes is to unset the Hidden and ReadOnly attributes on C:\IO.SYS. Of course, if the attributes on C:\IO.SYS are not standard for DOS, the macro may succeed in its attempt to truncate it, but it will still crash out at MSDOS.SYS.

Parasitic Dropper

The most interesting way to use *WordBasic* commands is placed in the third trigger routine; the DropSuriv macro. That macro is called from the macro AutoExec – that is to say, whilst *Word* is starting.

DropSuriv checks the system time: if it is between 17:00 and 18:00, the virus tries to infect the system with a parasitic COM, EXE and NewEXE file infector, called PH33R, using the standard DOS utility DEBUG.

First, it checks that the file C:\DOS\DEBUG.EXE exists. If so, the virus creates a temporary file, C:\DOS\PH33R.SCR, and saves the hexadecimal dump of the dropped PH33R virus there. Next, it appends the DEBUG commands Go and Quit to the end of that file, and creates a temporary file, C:\DOS\EXEC_PH.BAT, which contains the following text:

```
@echo off
debug < ph33r.scr > nul
```

It then changes the current directory to C:\DOS and executes the file EXEC_PH.BAT. As a result, DEBUG receives the file PH33R.SCR as input, converts its hexadecimal dump into binary code, executes it, and returns.

Infection is performed in a background DOS box, so the user does not even know that a program has been executed in the background, and his computer is infected with two viruses.

This is 'super-multi-partism' – infection of DOC files, of DOS COM and EXE files, and of NewEXE *Windows* files. Only two months ago, such an infection method could barely have been imagined; infecting a computer with a parasitic virus using only *Word* macros.

The newly-dropped PH33R virus stays memory-resident, hooks Int 21h, then writes itself at the end of COM, EXE, and NewEXE (*Windows*) files. PH33R uses the standard set to install itself into memory during execution from infected DOS files. In the case of NewEXE files, the PH33R virus uses DPMI calls to do the same things.

There are two Int 21h handlers in the virus code. The first is active if the virus has installed itself in DOS memory; the second, under *Windows*. To intercept file execution, both handlers check for file Exec (AX=4B00h), Open (AH=6Ch), and Get Attributes (AH=43h). In addition, the DOS handler intercepts the Extended Open/Create (AH=6Ch) function, and the *Windows* handler intercepts Open (AH=3Dh).

During infection, the virus checks the names of files, and only targets those files with the extensions CO?, EX?, DL?, excluding those with names matching the patterns *86.*, *AV.*, *DV.*, *AN.*, and *OT.*.

PH33R has no trigger routine, but it does contain the following internal text strings:

```
=Ph33r=
Qark/VLAD
```

Fortunately, Nuclear cannot drop PH33R, due to faults in the coding. An 'If' statement at the beginning of DropSuriv is missing the matching 'EndIf' required by *WordBasic*, so the macro cannot execute, and fails with an error box.

Even if this bug were not present, the idea would seem not to work, as PH33R would simply go resident in a DOS box, which would promptly close, taking the virus with it. PH33R intercepts file open – the author of Nuclear may be trying to infect COMMAND.COM whilst its code is reloaded. However, the virus' DOS Int 21h handler hooks Extended Open/Create (AH=6CH), whereas COMMAND.COM uses the more traditional Open (AH=3Dh). There may be other DOS versions or shell utilities that can be hit in this way, but *MS-DOS* stayed clean in all experiments.

Detection and Disinfection

The methods of detection and disinfection for the first *Word* virus also work for Nuclear [*see VB September 1995 p.9*]. The list of macros in the Tools/Macro menu must be checked for the nine virus macro names listed above.

The presence of these macros means that the computer is already infected. You may scan your disk (DOC and DOT files) for the presence of the following ASCII strings:

```
AutoExec AutoOpen FileSaveAs FilePrint
FilePrintDefault InsertPayload Payload DropSuriv
FileExit
```

```
PAYLOAD AUTOEXEC AUTOOPEN FILEEXIT DROPSURIV
FILEPRINT FILESAVEAS INSERTPAYLOAD FILEPRINTDEFAULT
```

There are several document vaccines which can detect and disinfect the Concept virus. These vaccines operate in the same way as the virus; installing themselves into *Word* global macro areas, then checking incoming documents for the virus' presence, disinfecting where necessary.

Automated detection, disinfection, and trigger prevention using such document vaccines is not always as easy as for Concept. The new virus does not terminate installation if there is a FileSaveAs macro already installed, but simply overwrites it.

If the macro AutoExec is present, the virus does not infect the system, but calls the Payload macro in an attempt to trash the system.

If the name AutoOpen is defined in both the global macros and the document macros, *Word* bypasses the global AutoOpen macros, executing the AutoOpen macro from the document in preference. So, the anti-virus document vaccine is disabled by *Word* when the infected document is loaded in this way.

The only reasonable way to protect against this virus is to disable the execution of the AutoOpen macro by using the 'DisableAutoMacros' *Word* system macro.

Nuclear

Aliases: WinWord.Alert, WordMacro.Alert, WW6Alert, WinWord.Nuclear, WordMacro.Nuclear

Type: *Microsoft Word* file infector.

Self-recognition in Word Processor:

Checks for the presence of an AutoExec macro.

Self-recognition in Word Documents:

Does not check files, but infects each file when File/SaveAs is selected.

Patterns in DOC Files:

The following strings are visible, both as printed and in upper case:

```
AutoExec  AutoOpen  FileSaveAs
FilePrint  FilePrintDefault
InsertPayload  Payload
DropSuriv  FileExit
```

Trigger: Drops PH33R executable file infector (COM, EXE and NewEXE virus), appends message while printing documents, corrupts DOS system files.

Removal: See analysis.

VIRUS ANALYSIS 2

Tai-Pan

Kevin Powis

Tai-Pan (aka Whisper) is an 'in-the-wild' memory-resident file virus capable of infecting EXE-format files. While it has neither a trigger mechanism nor a payload, it should be considered a threat, due both to it being in the wild, and the ease with which it replicates.

The virus appends itself to host files, increasing their length by 438 bytes. It issues an 'Are you there?' call to see if the virus is already in memory. If not, it goes resident, and returns control to the host. The interrupt handler includes the 'Are you there?' call receiver and the file infector. Tai-Pan also has a storage area for a portion of the host's EXE header.

Are You There?

When an infected file is executed, the virus receives control. Like most such viruses, Tai-Pan starts by checking that a copy of itself is not already resident and active in memory, making an 'Are you there?' call, via Int 21h, AX=7BCEh. The way this virus handles its 'Are you there' call is subtly different from the standard technique, and merits some explanation.

DOS interrupt requests are made by the calling process placing a function value in the AH register and a sub-function in the AL register. What Tai-Pan is doing, therefore is requesting DOS function 7Bh, sub-function CEh. If the DOS interrupt handler receives the request – which will be the case if the virus is not already resident – it immediately compares the function value against the highest valid DOS function. Under DOS 6, this is 6Ch.

When DOS realises that the function is out of range, it simply sets the AL register to zero and returns to the caller. As far as I am aware, this is a completely undocumented feature of DOS; however, the virus relies on it, and it works exactly as the author expects under all versions of DOS.

The resetting of the AL register signals to Tai-Pan that a copy of the virus is not present in memory. If a virus copy were resident, the interrupt request would not have reached DOS and would instead have been answered by the resident virus, which would indicate its presence by preserving the value 7BCEh in AX. This is just the beginning of a very close relationship between the two 'modes' of this virus.

When a resident virus copy signals to a 'transient' copy that it is active (as detailed above), it returns a pointer to itself in memory in the ES register. The transient copy uses this to locate the resident copy, and to copy into it hidden details from the host's EXE header. It then simulates a far return to a routine at offset 76h in the resident virus.

This routine (used again later) takes responsibility for using the details passed across earlier to calculate the correct code segment and offset to enable the host (whose image is still in memory) to run as normal. To achieve this, the resident routine issues a JMP to the original host code.

Going TSR

When the first infected file is run in each session, the virus will detect that there is not an existing resident copy of itself in memory and therefore installs itself accordingly. It makes no use of any traditional DOS TSR facilities, and demonstrates a subtlety which indicates that the author has an intimate knowledge of memory allocation.

Tai-Pan begins installation into memory by attempting to allocate 31 paragraphs (496 bytes) of memory. This may seem a strange amount at first: the author is obviously aware that DOS increases memory allocation requests by one paragraph to allow for the 16-byte Memory Control Block (MCB) which precedes all segments.

We are now seeking a nice round 512 bytes of memory. If this fails due to lack of memory, Tai-Pan simply creates its own 'free' segment by directly modifying the MCB of the current code segment, shrinking it by 512 bytes. Then it retries the original request, which now works because DOS indeed now has 512 bytes free!

In either case, Tai-Pan has secured a portion of memory in which to reside. Next, it modifies the owner field of the MCB associated with this memory to make it look as if it belongs to DOS. Once this is complete, the virus body is copied to its new home and control passes to the resident virus copy.

The first task of the resident portion of the virus is to take control of Int 21h, pointing it to the virus interrupt handler at offset D0h in the virus body. The program logic then drops into the routine at offset 76h mentioned above. This routine handles the execution of the original host, which runs as normal: the virus is now firmly lodged in memory and has control of all DOS activity.

The Interrupt Handler

As well as responding to 'Are you there?' calls, the virus interrupt handler also controls the infection process. The handler receives control every time any DOS interrupt activity takes place. Tai-Pan ignores everything except the program Exec function (4Bh).

When this is detected, the file about to be executed is opened, and the first 24 bytes read into memory. The contents are checked for a valid EXE header by comparing the first two bytes to be equal to 4D5Ah. If they do not indicate an EXE header, the program goes past unhindered. If the file has an EXE format, Tai-Pan then moves the file pointer to the end of the file. There are three reason for this:

- it allows file size to be checked – if this exceeds 64833 bytes, the file is reprieved and allowed to run as normal
- it allows Tai-Pan – by examining the EXE header file size – to calculate whether the first instruction in the program is 438 bytes from the end of the file. If so, the program is deemed infected and therefore ignored.
- it positions the file pointer, ready for the virus code to be appended to the host

Infection

If a file is to be infected, Tai-Pan uses standard DOS calls to obtain the file's time and date and stores them. It writes its own 428-byte image from memory to the end of the file, followed by 10 key bytes from the host EXE header. This makes the total length increase 438 bytes.

The virus amends the EXE header at the start of the file to ensure that it is given control when the program is executed. Finally, Tai-Pan restores the file's time and date. Infection is now complete and the original Exec call continues as if nothing has happened.

Conclusion

Tai-Pan is a tidy, well-written virus which packs quite a lot of functionality into just 438 bytes. It infects files as they are executed. The absence of any stealth capabilities, however, should make it easy to detect.

Tai-Pan

Aliases:	Whisper.
Type:	Resident EXE file infector.
Infection:	EXE-type files greater than 64833 bytes long.
Self-recognition in Files:	Program start point as defined in EXE header equal to 438 bytes from file end.
Hex Pattern (locates virus in files and memory):	`5E83 EE03 B8CE 7BCD 213D CE7B 7517 0E1F 81C6 AC01 BFAC 01B9` Alternatively, the string '[Whisper presenterar Tai-Pan]' appears unencrypted in all infected files.
Intercepts:	Interrupt 21h.
Trigger:	None.
Payload:	None.
Removal:	Automated removal is possible by picking up EXE header entries from the last 10 bytes in the file and restoring them to the correct position in the EXE header. The file must then be shrunk by 438 bytes. Alternatively, delete infected files and recover from a backup.

VIRUS ANALYSIS 3

Dementia – The File Thief

Eugene Kaspersky
KAMI Associates

Scanning inside file archives and compressed files is still not a standard feature in virus scanners; however, the number of viruses using compression grows from year to year. Viruses in the Cruncher family use *Diet*-like compression (one of which has been discovered 'in the wild'), and the ARJ virus places files into *ARJ* archives [*see VB December 1993 p.13*].

To this type of virus one can add the name Dementia. The name of this infector is taken from an internal text string, and from the video effect which displays a message about the Dementia VxBBS.

Dementia expands, to include ZIP, the list of archive formats which are understood by certain viruses. When it accesses ZIP files, Dementia places an infected dropper into the archive. The archive, when unpacked, is found to contain an extra file, called CALLFAST.COM: this contains the virus video effect routine, infected with the virus code.

Another feature of this virus is that it can 'steal' files from an infected computer. The virus intercepts ZIP archives on file open, and checks their contents for a special request file. If present, the virus adds the files listed in the request to that ZIP file – in effect, 'stealing' them from the infected PC!

Using this feature, it is possible to break some types of computer protection: it seems that this virus was constructed especially for this purpose. To break protection systems, key files (or passwords) must be obtained. To get these, it is necessary simply to infect the computer, put a special ZIP file on that computer, then take that ZIP file back. In the case of a BBS it is easy: upload the ZIP, and immediately download it again. A problem for BBS operators…

Installation and Infection

Dementia's installation and infection code contains nothing new. On execution, the infected file passes control to the virus decryption routine, which is non-polymorphic. Then the virus checks the system with an 'Are you there?' call (Int 21h, AH=3Eh, BX=1492h; the memory-resident virus code returns BX=1776h). If not already resident, it installs itself by patching the Memory Control Blocks, copying its code to the top of system memory, and hooking Int 21h.

The virus intercepts three DOS functions: CloseHandle (AH=3Eh), OpenHandle (AH=3Dh), and Execute (AH=4Bh). The virus uses the CloseHandle function, as mentioned above, as its 'Are you there?' call. On calls to Execute, the virus performs the infection routine. On calls to OpenHandle, the virus checks the filename extension. If the file being opened is a COM or an EXE file, the infection routine is called. If, however, it is a ZIP file, the virus calls the ZIP infection and 'file stealing' routine.

During infection, the virus opens the file and checks its date and time stamp – it does not infect if the seconds field has a value of 01 (i.e. 2 seconds). Then it reads the file header, checks for the EXE file stamp (MZ at the start of the file), and executes COM or EXE infection as appropriate.

Both branches of the infection routine contain standard parasitic virus algorithms. The virus encrypts and writes its own 4207 bytes to the end of the file, and then overwrites the file header with a JMP VIRUS instruction (COM files), or with modified header information (EXE files). It then sets the ID value in the file time stamp, closes the file, and returns control to the original Int 21h handler.

During infection, Dementia hooks Int 24h to prevent DOS error messages while attempting to write to write-protected disks, but it does not get and clear file attributes. As a result, the virus cannot infect files with a read-only attribute.

ZIP Archive Processing

When a file with the extension ZIP is opened, the virus calls a special routine which scans inside the ZIP file, and either infects it or, if the request file is present (see File Stealing below), adds to the ZIP file those files listed therein.

The virus does not call any ZIP-clones during manipulation of ZIP archives. It processes internal ZIP file records using information about their format (as documented by *PkWare*). The virus reads the file record by record, checks the type of these records, calculates the length of the records, the offset of the next record, etc. When adding data to ZIP archives, the virus uses a temporary file to store the ZIP file's 'system area'. This holds information on ZIP file content, enabling (amongst other things) easy extraction of single files.

The virus converts the data in the files to be added to the archive to ZIP record format, appends that record to the ZIP file, calculates the corresponding checksums, and updates the list of files in the archive before appending it to the archive. The ZIP archive then contains one additional record, which any PKUNZIP-compatible utility can unpack. The virus does not attempt to compress the data, but saves it 'as is'. This corresponds to the PKZIP archiving method called 'storing', used for data which either cannot be compressed, or which the user has requested should not be compressed.

ZIP Infection and File Stealing

When the virus detects that a ZIP file is being opened, it creates a file called !#TEMP#! to store the temporary data, gets and saves the time and date stamp of the ZIP file, and hooks Int 24h

(as it does during infection of COM and EXE files). Then the virus searches the ZIP file records for the files CALLFAST.COM, REQUEST.IVA and RECEIPT.IVA. Their presence (or absence) indicates the ZIP file's virus status. If REQUEST.IVA is present, it contains a request for the files to be 'stolen'. If both REQUEST.IVA and RECEIPT.IVA are present, the archive already contains the requested files. If REQUEST.IVA is not present, the virus checks the ZIP file for CALLFAST.COM. If this is not present, the virus infects the ZIP file.

The virus uses the 'worm' method to hit ZIP files. During infection, the virus creates a file called CALLFAST.COM, writes the video effect routine into it, and infects it. Then it adds the file to the ZIP archive being infected: thus the ZIP archive contains a file infected with Dementia.

For the virus to be able to spread to another computer, the ZIP file must be unpacked, and CALLFAST.COM executed. Sadly, many users download files from BBSs and execute them without scanning, despite the thousands of incidents of computers being infected during execution of such files.

If the virus detects the file REQUEST.IVA, but no corresponding RECEIPT.IVA, it reads REQUEST.IVA from the archive and checks its format. If REQUEST.IVA contains four ID bytes (92h, 14h, 76h, and 17h) at the beginning, the virus looks for the file names following the header. These names are encrypted, and the virus decrypts them before searching the disk for corresponding files. There may be several file names, or masks. Use of wildcards is allowed.

The virus then scans the subdirectory tree for each requested file mask, starting from the root directory. The virus saves all files found in RECEIPT.IVA, using a format I have never seen before. RECEIPT.IVA contains records of special format, each of which starts with a header containing the file name and length and some other information, followed by the file body.

When the search is complete, the virus encrypts the file RECEIPT.IVA (simple XORing with FFh), storing it in the ZIP file containing REQUEST.IVA. Thus it is possible to take any file from an infected PC without direct access. Uploading, and then downloading, the special ZIP file will be sufficient to break the security.

Figure 1: The message displayed by the Dementia virus, from which it takes its name.

Trigger Routine

The trigger routine receives control during execution of CALLFAST.COM. This file is the infected dropper placed in ZIP archives. When it is executed, the dropper displays the message shown in Figure 1.

The virus contains internal text strings, only some of which are used while ZIP files are processed. The others are unused:

```
!#TEMP#! REQUEST.IVA RECEIPT.IVA CALLFAST.COM *.*

Dementia]

Copyright 1993 Necrosoft enterprises - All rights
reserved

I am the man that walks alone
And when I'm walking a dark road
At night or strolling through the park
When the light begins to change
I sometimes feel a little strange
A little anxious when it's dark
```

Dementia

Aliases: Dementia2.

Type: Memory-resident, parasitic, COM and EXE file infector. Adds worm (dropper) to ZIP files. Encrypted.

Self-recognition in COM/EXE Files:

Checks seconds field in file time stamp. Infected files are marked with 01h in seconds field.

Self-recognition in ZIP Files:

Checks ZIP archive for the presence of the file CALLFAST.COM.

Self-recognition in Memory:

'Are you there?' calls, Int 21h, AH=3Eh, BX=1492h. The TSR code returns 1776h in the BX register.

Hex Pattern in Files:

```
E800  005E  81C6  6A10  8BFE  FDB9
2908  BA??  ??0E  0E1F  07AD  33C2
```

Hex Pattern in Memory:

```
9CFC  80FC  3E75  0B81  FB92  1475
05BB  7617  9DCF  80FC  3D75  663C
```

Intercepts: Int 21h for COM, EXE, ZIP file infection, and the 'stealing' of files by special request (see analysis for further details).

Trigger: 'Steals' files; displays message.

Removal: Under clean system conditions, identify and replace infected files. Check ZIP files for CALLFAST.COM, REQUEST.IVA, and RECEIPT.IVA.

FEATURE

Revisiting the DOS Scanner Testing Protocol

It is undoubtedly true that one of the trickiest things *Virus Bulletin* has to do is to compare products. This is in some ways akin to comparing, say, a deck chair, an upright dining chair, and an easy chair – in different respects, each is the best. They all have their role.

So it is with anti-virus products – a product which is best in one set of circumstances may fall far short in another. In recent months, *VB* has decided to extend, adding to and modifying, the existing testing protocol [*see VB, February 1995, pp.12-13*]. As always, it is hoped that this protocol will invite suggestions for improvement and refinement.

Scanning: The Be-all and End-all?

Modern anti-virus products offer much more than scanning, and are capable of more than scanning a cleanly-booted machine, pointing out the viruses on the hard disk, and then exiting. Where increased functionality is offered, it should be tested. With this in mind, we must consider what to test.

The issue of testing TSRs is ever-present, and will be the focus of a separate review in the near future. Forming a testing protocol for resident products is a tricky exercise; in many ways more difficult than for scanners.

Disinfection has as yet been tested by *VB* only once [*see VB, September 1994, p.11*]. Whilst it is always the case that replacing infected objects with a clean backup is the best solution, it is not always possible. Disinfection seems to be growing in importance, with more products offering it and more people using it; thus, a small disinfection test will be included in future comparatives. Products which do not offer this will not be explicitly penalised, but it seems reasonable to expect that if a product has such a feature, it should work.

Another problematic task for the scanner is being run on a machine which has not been clean-booted. Stealth viruses are common in the wild these days: under many conditions, if these viruses are not detected in memory, the scanner itself will cause the virus to spread throughout the machine. If the virus can be detected (or better, *disabled*) in memory, the user is to an extent protected. Future *VB* DOS scanner comparatives will include a small number of such tests.

The tasks (to be called 'disinfection' and 'virus active' tests) are challenging for reviewer and scanner. Testing requires considerable time, and a great deal of care, for each product, and cannot be done with the whole test-set – we would be testing from now until the millenium, by which time it could quite legitimately be argued that the scanners were out of date.

Thus, both tests will be done with a carefully selected, *small* set of viruses. For disinfection, tests will be selected by the way viruses infect the system. For the 'active in memory' test, the test-set will only include viruses known to be in the wild at the time of the deadline for submission of products for the review. Where possible, the different techniques used by viruses in memory will be covered; however, there are relatively few of these.

Thorn in my Side

The thorns discussed by my predecessor in February are just as sharp today – so-called review modes, and other ways in which a scanner modifies its sensitivity (and thus speed) in mid-scan without informing the user when it detects a review situation. Indeed, one product detunes its sensitivity in such situations to run faster across the test-set: it is a potent argument that no user is going to have hundreds of different viruses on a computer.

Heuristics lend a hand here: if a product can be said to detect viruses heuristically, why then should it not act heuristically in response to its environment? Quoting scan times on clean sets of files, we can avoid problems introduced by many different infections tipping the product off that it is being reviewed. This does not help the detection issue, however.

As in February, the only real response to a product which *increases* its sensitivity in a review situation is to spot-check single files. Though it is easy, using a simple combination of DOS commands, to run the scanner afresh for each file in the collection, the load-time of most products in this scenario would greatly exceed the actual time spent scanning, rendering the technique impractical for a reviewer.

Scoring

The weighting used to calculate the score of a scanner on the polymorphic test-set will continue in future reviews. The importance of a scanner being able to find all individual occurrences of a virus has in no way diminished.

Scoring in the remaining categories of file virus (In the Wild and Standard) will be slightly modified. Previously, scoring was a simple fraction (files tagged as infected divided by files in test-set). A formula has been suggested which makes the system immune to accidental overweighting by one particular virus, and is described in the panel opposite.

Scoring for the new tests is altogether more problematic. It is impossible to reduce to a simple numeric score either the product's ability to disinfect, or its reaction to finding a virus active in memory. The results of the tests will be described and tabulated: one advantage of having a small range of tests is that it is possible to describe the results in more detail.

This article and the table below describe additions and modifications to the testing protocol which will involve extra care and work on the reviewer's part. Information generated, however, will be worth the extra work.

Any comments readers might have on this protocol are more than welcome – please contact the editor, either by fax (+44 1235 531889) or by email (ian@virusbtn.com). The next comparative review will appear in *VB* in January 1996.

VB Comparative Evaluation of DOS Scanners

Product Category: Non-resident virus-scanning and virus-removal software running under DOS.

Objective: To provide the essential criteria by which to judge the relative speed and accuracy of virus scanning and removal programs on infected and uninfected files.

Components Tested: Only non-resident scanning and removal tools are tested. Separate TSRs and checksummers provided within the same package will not be tested. All tests except the active virus test will be carried out in a clean *MS-DOS* environment, and the active virus tests will be performed with only one virus active at once. Disinfection tests will be performed on objects infected singly – multiple infections will not be used.

Hardware: The hardware used for the test is supplied by *Virus Bulletin*, and specified in detail when the review is published. All speed trials will be carried out on the same machine using the same configuration, although different machines may be used for the active virus and boot sector scanning tests. Information concerning the speed of the hard disk drive used will be provided, as calculated by *Norton Utilities v7*.

Virus Test-sets: The viruses used for testing will be provided by *Virus Bulletin*. They will consist of genuine single infections of computer viruses stored in a replicable state. No first generation samples, droppers, or Trojan horses will be used, unless they themselves are created by a virus which is included in the test-set. File viruses will (where possible) be attached to one of a number of standard goats. Boot sector viruses are supplied individually on genuinely infected diskettes (see note under 'Testing methodology'). Details of the test-sets used will be given in the review.

Tests: The tests which the products will undergo are:

- Scan time for an clean diskette
- Scan time for an infected diskette
- Scan time for a clean test-set, stored on hard disk (also doubles as false positive test)
- Detection rates and percentages for the three file virus test-sets and the boot sector virus test-set
- Response to execution with viruses active in memory (one at a time)
- Disinfection (where available) of file and boot sector viruses

Testing methodology: All scan tests will be performed with the scanner in its default mode of operation, apart from certain options which must be used. These are: run non-stop, write a report file, do not issue audible alerts, and report names of files considered clean as well as those considered infected. Where the default mode of the scanner is not expressly defined, the mode in which it is used will be specified.

Boot sector tests will, where possible, be run using SIMBOOT (a program written by Dmitry Gryaznov) which allows scan tests against boot sector viruses to be performed against images of diskettes rather than against the diskettes themselves. This greatly increases the speed of such tests. If the product cannot be made to work within the SIMBOOT environment, this is considered to be a problem with SIMBOOT rather than the product, so the product is not penalised. If a product fails to detect a boot sector virus within SIMBOOT, it is tested against the genuine infection on diskette.

Calculation of overall scanning results: [*See VB, February 1995, p.12 for details of how the percentage detection rate on the polymorphic test-set will be calculated. Copies available on request from VB offices.*] The detection rate of a product against the In the Wild and the Standard test-sets will be calculated using the following formula:

$$100 * \left(\frac{V_1}{N_1} + \frac{V_2}{N_2} + \ldots \frac{V_i}{N_i} \ \ldots \ + \frac{V_n}{N_n} \right) / n$$

Where: V_i = Number of samples of virus 'i' in test-set identified as infected

N_i = Number of samples of virus 'i' in test-set

n = Number of viruses (as opposed to *samples*) in test-set

CONFERENCE REPORT

VB 95: Reaching the World

'I like to be in America; OK by me in America…' The words to the famous Leonard Bernstein song seemed very apt as the plane touched down at Boston's Logan International Airport; the beginning of *VB 95*. Over the next two days, delegates and speakers from around the world would register at what has become the world's most respected conference on computer viruses, held annually by *VB*.

This year saw an increase in the number of talks presented, with one corporate and two technical streams. Previous years have seen delegates concentrating on one or other of the streams: 1995 brought changes, with many attending a mixture of talks in all streams. Discussion centred on cooperation; on sharing of knowledge and information.

Kick-Off

The first session of the conference was, as ever, the introductory talk on computer viruses by Dr Jan Hruska of *Sophos*. The seminar was well-attended, and gave delegates the opportunity to familiarize themselves with the current state of play in the anti-virus world.

Thursday morning's opening address saw many slightly bleary-eyed following the previous evening's cocktail reception sponsored by *McAfee Associates* (offers for next year's events, producers?!), but a blast of a certain Rolling Stones song, much touted with the advent of *Windows 95*, shook everyone out of their somnolescent states.

VB editor Ian Whalley discussed the pros and cons of *Microsoft's* latest operating system, and the ease with which *Windows 95*-specific viruses may be written. He also addressed problems associated with new types of viruses, particularly with regard to Concept (a topic which was to surface again and again over the next two days).

Professor Harold Highland, one of computing's 'elder statesmen' (who prefers to refer to himself as one of its dinosaurs), gave an informative and interesting keynote talk on his experiences in computer security – in fact, a short history of computer viruses. The first macro virus, to his knowledge, was written in 1989 – by Highland himself! After experiments, he realized its ability to spread was vast, so he stored it in a secure place, hoping such a virus would never be seen in the wild – this year saw that hope dashed.

The First Goals

After a welcome coffee break, the conference separated into one corporate and two technical streams, and the real work of the day began. Sarah Gordon (*Command Software*) opened the corporate stream with an extension on her last year's talk on the psychology of the virus writer; addressing the more general issues of why viruses are written and how to encourage end-users to implement anti-virus strategies.

At the same time, the technical streams had Jonathan Lettvin (*Lotus*) discussing the PC boot sequence, and *ESaSS'* Frans Veldman lecturing on one of his areas of expertise, heuristics. Paul Ducklin's (*Sophos*) presentation on learning from mistakes was salutary, and illustrated how human weakness could lead to errors being made and even remade.

After lunch, *VB's* technical editor, Jakub Kaminski, gave a talk on the Flash BIOS, and the problems which can arise when the BIOS contents are reprogrammed. He was followed, in the other technical stream, by *Symantec's* Shane Coursen, who gave a very topical lecture on the vulnerability of *Windows 95* to viruses. In his view, the discovery of the Concept virus shows that new types of virus are becoming more prevalent.

After the tea-break, while presentations were held in both technical streams, Wes Ames (*Boeing*) led a Corporate Stream discussion forum on problems encountered by IT security managers.

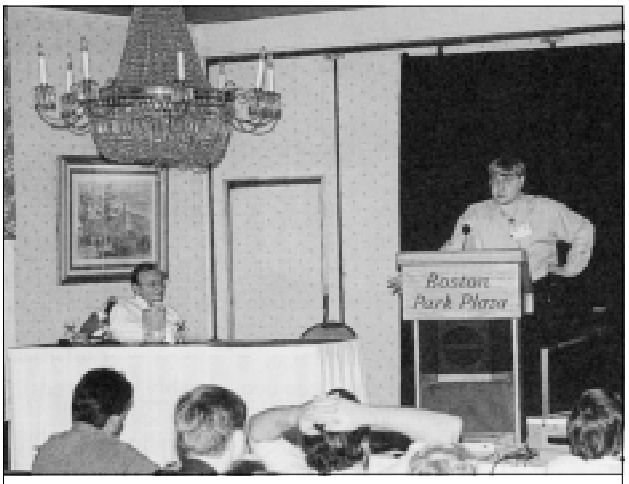
Half-Time

Happily, Friday morning's sessions began somewhat later than the previous day's – after a large gala dinner, and an enormously entertaining casino evening, some faces still only surfaced after the coffee break!

Paul Robinson, editor of one of our fellow security publications, *Secure Computing*, opened the corporate stream for the final day with a lecture on how to test and review anti-virus products. Other highlights of the morning included Righard Zwienenberg's talk on heuristic scanners (involving two eggs!), and Pavel Lamacka's discussion as to whether it is possible to have harmless/useful viruses – he believes not.

Roger Riordan of *Cybec* was scheduled to give a talk on IDE hard disk security; however, as he had been rushed to hospital for emergency surgery just days before he was due to leave for the conference, a colleague, Robert Stroud, gave the talk in his place – twice, by popular demand! Riordan, however, did rush a specially-made videotape over from Australia, so that he could at least introduce his talk.

The afternoon sessions saw some of the heavyweight anti-virus 'names' take the rostrum: Fridrik Skulason on the latest trends in polymorphism, Dmitry Gryaznov on the future of the scanner, Steve White on a global perspective for computer viruses, and Jim Bates on virus writers. Former *VB* editor Richard Ford (now at the *NCSA*) presented a stimulating talk (including last-minute alterations and additions) on this year's most-discussed topic; macro viruses.



Fridrik Skulason at *VB 95*; here discussing things other than *CARO* and beer drinking!

The conference closed with an invigorating and lively panel session, in which speakers were posed questions by their audience. On the panel were Jim Bates, Paul Ducklin, Richard Ford, Sarah Gordon, Mike Lambert, and Steve White. This session carried on almost naturally from Richard Ford's talk on macro viruses, and led to the somewhat surprising discovery that many of the delegates had been caught by Concept, and that, in the larger companies, infections went into the hundreds.

Steve White posed the theory that macro viruses could possibly supplant all other types of virus, in terms of prevalence. Another speaker, Jonathan Lettvin (this time as part of the audience), put forward the widely-shared view that macro viruses are the beginning of a large new problem.

Discussion on the efficacy of product reviews followed, with participation from vendors, researchers, and a *VB* journalist. Much of the audience felt that disinfection should be an integral part of anti-virus software, and calls were made for comparative reviews of that capability.

Taking a Gamble

Contrary to what readers may think after reading thus far, not every waking hour was spent with viruses, viruses, viruses – there was certainly time for play! Many delegates brought their partners, who enjoyed an extremely busy partner's programme on the first full day of the conference, touring Boston and its environs, and seeing local sites of historical interest.

Thursday night's Gala Dinner was, as always, enjoyed by the great majority of delegates and guests – the food was superb, as it was throughout our stay at the *Boston Park Plaza*, and copious amounts of wine were available for those of us who imbibe. In a departure from previous years, some delegates brought not only their partners, but also their children – they (and the grown-up children!) were kept entertained throughout dinner by roving magicians who performed card tricks and made balloon models.

After dinner we were regaled with a cabaret, the most memorable point of which was the artiste balancing a stepladder – indeed, at one point, there were two! – on his chin. The cabaret was interrupted from time to time, as had been much work on the previous day, by fire alarms – the broadcast message will stick in many memories for some time: 'The cause of the alarm is still being investigated. There is no need to leave your rooms…'

The real entertainment of the evening was the casino, with blackjack, roulette, a 'wheel of fortune', and crap tables. Every 'gambler' was given $50,000 (sadly, fake…) on entering the ballroom, with the promise of a bottle of champagne to the person who won the most 'money' by the end of the night. This correspondent was proud of a $285,000 final total [*I went bust. Ed.*], until realising that the winner, Maureen Morar, had made a cool $¾ million.

This year saw another departure from tradition – rather than a speaker's dinner, there was a whale watch for speakers and staff. Despite warnings of heavy seas, most speakers decided to brave it, and were rewarded with sightings of five whales. Unfortunately, sightings of lunch proved more elusive: after a fire on the bottom deck, most of the food was ruined. However, as luck would have it, while Jan Hruska and the crew were running around putting the fire out and throwing smouldering cloths overboard, Philip Statham and Chris Baxter had the foresight to apply themselves to rescuing the sandwich trays, and managed to salvage enough to keep the weary travellers going until the end of the trip. Why do fires seem to follow the *VB Conference* so faithfully… ?

Thanks and Thoughts

A great deal of hard work went into the organising: thanks are due to many people, in particular Dale Tabrum for sterling work in keeping everything under control and 'holding the fort' in England while we were all away, and to Julia Line for masterful efforts with conference papers and proceedings. Thank you to Penny Halliday and Kim Ducklin for helping out in Boston, and to conference pro John Merne for assisting our conference manager Petra Duffield. Petra, as always, was the brains behind the operation; special thanks to her. We understand she is already thinking about *VB 96* (did someone mention Tahiti, Petra?).

Thanks of course to all the speakers; many more than there was space to mention in this report (apologies to those not covered). Without their expertise and their commitment, this conference could never have taken place. Finally, a vote of thanks to all the delegates: your active participation and continuing interest are the reasons we hold this conference.

The *Proceedings of the Fifth Annual Virus Bulletin Conference*, with copies of papers given at the event, are now available from *Virus Bulletin* offices; contact Dale Tabrum or Petra Duffield for information.

Those delegates who have not submitted a completed assessment form (to be found inside the Proceedings) may still do so – these enable *VB* to continue to improve next year's conference.

PRODUCT REVIEW 1

NetShield

Jonathan Burchell

Sooner or later it had to happen: not only has *McAfee's NetWare* virus protection solution (*NetShield*) been extended to include new functionality and features, but version 2.2 also ships on CD-ROM.

McAfee has always used shareware and the public domain distribution network (BBSs, computer networks, FTP sites and specialised dealers) to provide a 'try before you buy' product: going to CD-ROM not only reduces distribution costs (CDs are cheaper than multiple floppies) but also allows distribution of several products, support files and documentation on a single medium.

NetShield provides protection for *NetWare* 3.11, 3.12, 4.X, and SFT III. Like many anti-virus products, *McAfee* requires patching of *NetWare* 3.11 and 3.12 servers: in fact, the recommended versions of A3112.NLM, AFTER311.NLM and NWSNUT.NLM are 4.10A, the highest seen so far.

One advantage of CD-ROM distribution is that the necessary *Novell* patch files can be included. If you download the product from a BBS, it's back to the modem for another download from *McAfee's* or *Novell's* BBS. Installation for both CD-ROM and BBS versions is identical (having first unzipped the downloaded files) and consists simply of running 'setup'. This can be done from *Windows* or a DOS prompt, but the install program is a *Windows* executable, so requires *Windows* on the installation workstation.

The installer allows options to be set for both workstation console program and server installation, as well as checking that required disk space exists in all target locations. Assuming the disk space is available, you will want to install on-line documentation and have a *Windows* program group and icons generated, to simplify later program start-up.

What's New

Compared to earlier versions, *NetShield* has been extended in a number of ways. The first and most obvious is the provision of a *Windows*-based server administration and configuration program. This program allows GUI-based server control from a workstation and, whilst it does not allow servers to be grouped into logical domains, it does allow access to all those servers running *NetShield* from a single location. For teletype-based die-hards [*who, me? Ed.*], the original *NetWare* console interface is available either on the file server itself or via RCONSOLE on a workstation.

Extensive new facilities have been added to alert users and operators via *NetWare* broadcasts, console messages, email and even directly via dial-up pager services. *NetShield* now offers several options for networks, requiring extra levels of security over and above those *NetWare* has built. *NetShield* allows administrators to restrict write access to specific files and directories to selected users. It is also possible to grant temporary authorisation (in terms of a number of minutes) to specific users so as to allow software upgrades and installation work to take place.

The documentation has also been considerably improved. *NetShield's* printed manual used to be a slim and always rather out-of-date document: now, however, the software is accompanied by a smart new manual, which provides a good introduction to the working and configuration of the software, including screen-shots. Even better; for CD-ROM users, the documentation is included in electronic form, in *Adobe Acrobat* format, which is a joy to look at and use. (*McAfee* includes a copy of the necessary *Acrobat* reader on CD-ROM.) *McAfee* is to be commended for distributing documentation on disk in such a usable manner.

Software Features

The *Windows* software is started from a standard *Windows* icon, at which point a list of available file servers is presented. Getting to work consists of attaching to a chosen file server – the password must be known, but it does not require you to be logged on to the server as a user.

Whilst the software does not allow identically configured servers to be administered as a single logical domain, the idea of server configuration files is supported, and these can be manually copied between servers which are to be identical in configuration, easing multi-server management.

The *Windows* software needs both a keyboard and a mouse to drive it. Some functions cannot be accessed without a mouse – this is an increasing trend in such software (including many

The *Adobe Acrobat* document reader, included on the CD-ROM, provides online documentation in easy-to-use form.

examples from *Microsoft*). Despite the fact that it contravenes the original User Interface Specification, this is a not unrealistic evolution in program operation.

In use, the software provides a real-time overview of server operation, subdivided into scanning, notification and security property pages, as well as configuration options. The software offers no warranties with respect to operation under *Windows 95*; however, it seems to work perfectly satisfactorily in this environment.

Scanning

NetShield supports three types of scanning: on-demand, for immediate scans of selected volumes/areas; on-access, for real-time scanning of files as they are read from or written to the server; and periodic scanning, for scheduled scans of the server on a daily, weekly or monthly basis.

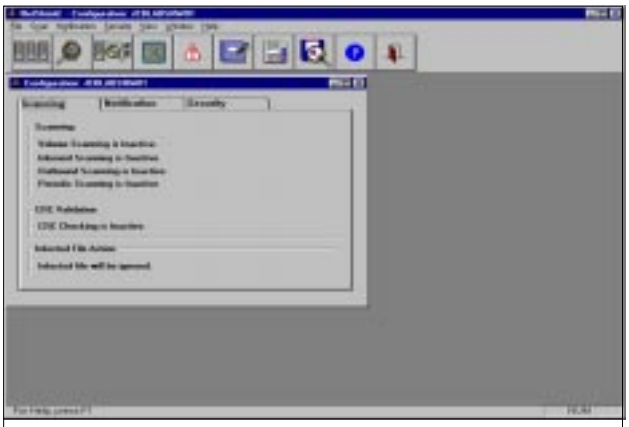
On-demand scanning consists merely of specifying the volume to be scanned. The scan cannot be limited to a given area, although specific areas may be excluded via an exclusions list, global to all types of scanning. Nor is it possible to specify what file types to scan: on-demand scanning probably scans those files listed in the default extension list.

Periodic scanning consists of specifying volumes to be scanned, together with frequency (daily, weekly, monthly), start time and start day of the week (or of the month). Only a single periodic scan can be active at any one time, though scan settings can be saved and restored. As with on-demand scanning, configuration is limited to specifying the volumes and the same global list of excluded directories. Also as for on-demand scanning, no facilities exist to specify what files are scanned, so we must assume that all files are scanned.

On-access, or real-time, scanning can be configured to check files which are incoming, outgoing, or going both ways. Checks can be set to all files or limited to those with extensions BIN, COM, DLL, EXE, OVL and SYS. This list can be edited, but not from the *Windows* software – to do this, you must pull up the real *NetWare* console interface. Minus ten for this; it is a pain, and it may be confusing for users to have to use several programs to administer server operation.

Once a suspicious file has been discovered by any scanning process, the 'Infected File Actions' are invoked. These can be set to ignore (only a log entry is generated), delete and move. The move option allows the exact location of the quarantine directory to be specified – by default SYS:\INFECTED is used, which should cause few problems. It would be nice if the quarantine directory was automatically added to the excluded directories scan list. It is not, so if you forget to do this, immediate and periodic scanning will produce 'false' reports once the quarantine directory is no longer empty.

Scanning also allows CRC checking of files to be established. The documentation is rather poor at explaining what this does. The manual warns that these should not be data files, binary files, log files etc, but does not explain how to establish the list



The *Windows*-based server administration program (here running under *Windows 95*), displaying the 'Scanning' property page.

of files which are CRC'd. I assume it is limited to the same list as real-time access scanning. Proper CRC operation requires great flexibility in configuration. I did not check further into this mechanism; however, it can only represent viable virus protection when adequately documented and properly designed to counter attacks.

The final configurable feature in this section is cross-server updating. If this is enabled, *NetShield*-protected servers converse amongst themselves to establish who has the latest version of the signature database. Once decided, they automatically update themselves from the server with the latest version, so all servers have the most recent version installed.

Notification and Reporting

NetShield offers several options for notification of virus detection, including network broadcasts, console messages, email notification, and pager notification. Optionally, it can keep a record of scanning events and results in a log file. Logging options are simplistic, and allow you to choose whether logging is active (and if so, whether the output should be appended to an existing file or simply overwrite the current file) and the name of the file to hold log data.

No options to print or filter this file for report purposes are provided and, as the file is not documented, writing a third party report generator based on the contents may be problematic. However, users can view, print, and export configuration and report files from the administration program, which uses *Windows Notepad* to manipulate the files.

Users may be notified via network broadcast that an infection has been detected. A drop-down list allows users (but not groups) to be chosen. Strangely, the concept of 'file owner' seems to be missing from the user list. It is not possible to customise the message the user sees.

Email may also be sent via *Novell's* global message handling service (also basic MHS) to a specified list of users. Mail has the advantage of being persistent (i.e. not dependent on the recipient being logged in at send time), but, like the broadcasts, the content of these messages cannot be customised.

NetShield supports pager notification via a *Hayes*-compatible modem connected to the server and a compatible pager service. Finally, console administrators can be alerted to infections via the standard console message system.

Security

In addition to virus scanning and detection features, *NetShield* offers a number of optional enhancements to *NetWare* security. The control here makes it possible to restrict access and to monitor access on a per user or group basis. This control is flexible; it is possible to specify files (and file types) to be excluded from monitoring. Any attempts to access a protected file are stopped and recorded to a log file.

I see this as an extremely useful feature, both in terms of enhancing security and, perhaps more practically, tracking down the elusive application which is legitimately changing its executable, or the users who insist on fiddling with settings on the server applications to suit themselves.

Another security feature is the ability to give temporary access to a restricted area, on a per-user basis, for a given period (a maximum of 180 minutes). This is great: allowing maintenance or software updates to a server often means extending supervisor privileges to non-authorised personnel – fine in the context of an update, but it is a problem remembering to withdraw privileges later. Using the temporary authorisation feature, they are automatically withdrawn at the preset time.

File Server Console

The file server console allows access to features described above. Some options (such as the extension list, and NLM priority when scanning) can only be set from the file server console. If features are so specialised or advanced that ordinary administrators should not be given access, then this should be controlled from within the administration program by providing different logins, not by requiring the advanced user to use two different interfaces to administer the system.

Conclusions

I find myself in rather a dilemma in rating the product. The improvements to the user interface and the provision of a *Windows*-based administration tool are excellent; however, it is still necessary to use *Windows* and *NetWare* console tools – for instance, the name of the file being scanned, and results, only show up on the *NetWare* console.

The functionality should be sufficient to manage limited numbers of servers, but features such as logical domains are lacking, and scheduling/scanning options seem limited. The Standard test-set score is excellent, but In the Wild results are not terrific – there is absolutely no excuse for not getting a 100% score in this test-set.

Most disappointing were the polymorphic scores: a score of less than 60% does not place *NetShield* in the top division. The past year has proved that this hallowed ground, once occupied by *S&S* and *Sophos* alone, is not unapproachable: recently, *InocuLAN's* (*Cheyenne*) and *IBM's* results rate them as strong contenders.

It is encouraging, however, to see that *NetShield's* detection rates overall have improved considerably since the last stand-alone review undergone by the product [*see VB, August 1994 p.21*]. *McAfee* has told *Virus Bulletin* that a new version with an enhanced scanning engine will be released at about the time that this review goes to press – we shall have to wait and see.

NetShield v2.2

Detection Results:

Standard Test-Set[1]	229/230	99.6%
In the Wild Test-Set[2]	114/126	90.5%
Polymorphic Test-Set[3]	2622/4796	54.7%

Technical Details

Product: *NetShield v2.2*.

Developer: *McAfee Associates*, 2710 Walsh Avenue, Santa Clara, CA 95051-0963, USA. Tel +1 408 988 3832, fax +1 408 970 9727.

Price: $450 for a 25-user, two-year site licence; other site licences available. Includes monthly updates and tech support.

Hardware used: Client machine – 33 MHz 486, 200 Mbyte IDE drive, 16 Mbytes RAM. File server – 33 MHz 486, EISA bus, 32-bit caching disk controller, *NetWare 3.11*, 16 Mbytes RAM.

Each test-set contains genuine infections (in both COM and EXE format where appropriate) of the following viruses:

[1] **Standard Test-Set:** 1049, 1260, 12 TRICKS, 1575, 1600, 2100 (2), 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 777, 800, 8888, 8 TUNES, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), AntiCAD (2), Anti-Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Captain Trips (2), Cascade (2), Casper, Dark Avenger, Darth Vader (3), Datalock (2), Datacrime (2), Datacrime II (2), December 24th, Destructor, Diamond (2), Dir, Diskjeb, DosHunter, Dot_Killer, Durban, Eddie, Eddie2, Fellowship, Fish_1100, Fish_6 (2), Flash, Flip (2), Fu_Manchu (2), Halley, Hallochen, Helloween (2), Hymn (2), Icelandic (3), Internal, Invisible_Man (2), Itavir, Jerusalem (2), Jocker, Jo-Jo, July 13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (5), LoveChild, Lozinsky, Macho (2), Maltese_Amoeba, MIX1 (2), MLTI, Monxla, Murphy (2), Necropolis, Nina, Nomenklatura (2), NukeHard, Number_of_the_Beast (5), Oropax, Parity, PcVrsDs(2), Perfume, Pitch, Piter, Polish_217, Power_Pump, Pretoria, Prudents, Rat, Shake, Slow, Spanish_Telecom (2), Spanz, Starship (2), Subliminal, Sunday (2), Suomi, Suriv_1.01, Suriv_2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Syslock, Taiwan (2), Tequila, Terror, Tiny (11), Todor, Traceback (2), Tremor, TUQ, Turbo_488, Typo, V2P6, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virdem, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Warrier, Willow, WinVir_14, Whale, Yankee (6), Zero_Bug.

[2] **In the Wild Test-Set:** As printed in *VB*, August 1995, p.19.

[3] **Polymorphic Test-Set:** 4796 genuine samples of Cruncher (25), Uruguay.4 (75), Satanbug (100), Girafe (1024), MtE (500), One_Half (1024), Pathogen (1024), Smeg_03 (1024).

PRODUCT REVIEW 2

No.More #*!$ Viruses?

Dr Keith Jackson

NoMore #!$ Viruses* (*NoMore*) is a new product. Not only is it new to the marketplace, its very concept is different from other anti-virus products. It is not based on scanning technology. Its main function is to detect boot sector viruses, but it can also detect multi-partite and system infectors. It does this by executing at power-up, and checking that the system is clean. *NoMore* was provided for review on a single 3.5-inch, low density (720 KByte) floppy disk.

Documentation

The product's documentation comprises a single, 63-page A5 booklet. Its style is quite straightforward, if (as are all too many such tomes) a tad boring for the poor reviewers who actually have to read it, as opposed to using it merely for reference purposes.

Thinking along such lines, the manual contains no index, and no detailed explanation of any error messages which may appear, so finding things is not too easy. However, it must be said that the long descriptions in the Table of Contents do help matters somewhat.

The manual itself is well-written, with explanations by somebody who obviously knows what he is talking about. The content is well-explained, even if it does not resort to great detail – do not expect to use this volume as your sole guide to fighting computer viruses.

The documentation makes it clear that *NoMore* is not intended to be a complete virus protection system; the manual states that '*NoMore* is not intended to replace scanning technology such as our *Vi-Spy Professional* and *Vi-Spy Universal NIM* … it is designed to complement them'.

Under Warranty

It is not often that I comment on the legal agreements which accompany various anti-virus products, but there are exceptions to every rule. In the terms of the licence agreement which covers *NoMore*, the user must agree to 'certify in writing' to the developers that all copies of *NoMore* have been destroyed if and/or when the licence is terminated.

This is another fine example of lawyers inhabiting a different planet from the rest of us mere mortals. How many people are going to comply with this constraint and write off to explain that they no longer wish to use a product?

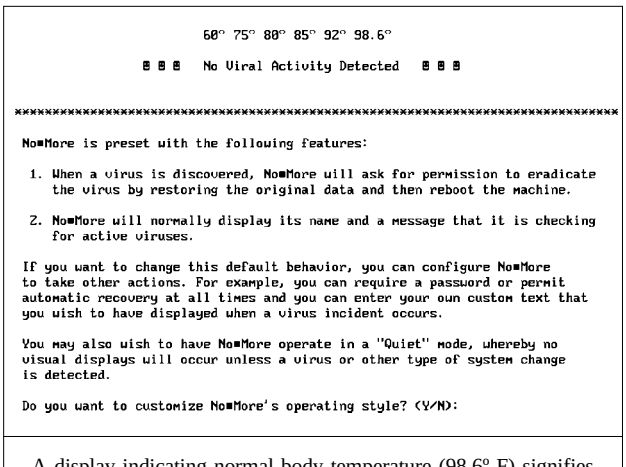
The warranty provided with *NoMore* is also a curious document. I quote: 'In the event of any Warranty claims, RG, at its sole discretion, will repair or replace the diskette'. So, you'll get a new floppy disk – if you're really unlucky, the company might even 'repair' (their words!) the original. All this gobbledegook, and other clauses which exclude all liability (unless the local legislature has had the sense to outlaw such shenanigans), has no place in a serious product. Unfortunately, many products these days have such bizarre legal agreements – perhaps it is time to bring lawyers back to the real world?

Installation

The manual which is provided with the product claims that the installation should 'typically take 3–5 minutes': this is a claim which corresponds closely to what actually happened. I had no problems installing *NoMore* – it really was very straightforward.

The installation program asks first if you are using an 'active software security package' (e.g. an access control system or a disk encryption system). Given the way in which *NoMore* operates, it cannot perform properly if such a product is present. This fact is well explained in the manual, and installation does not proceed unless the answer to this question is 'No'.

Continuing onwards, the installation program then says that 'PC Thermometer™ is analysing your system'. Note that this name is even trademarked! The manual claims that PC Thermometer will 'check your system for the presence of an active virus'. When PC Thermometer executes, it produces onscreen messages (see Figure 1) saying 60º, 75º, 80º, 85º, 92º, 98.6º (body temperature in Fahrenheit, geddit?). I have no idea what all this means, and as the manual does not give any details, I also have no idea how my computer was checked for viruses – but it certainly seemed happy enough with my system.



A display indicating normal body temperature (98.6º F) signifies that no viruses are present; and, although *NoMore* has several preset features, customization options are also available.

After choosing between a few customization options (for example, whether *NoMore* should ask before removing a virus, or whether a password is required), and performing a couple of reboots, installation was complete. During the installation process, *NoMore* creates a fixed-name subdirectory (C:\MBBOOT) and creates 125 KBytes (marginally more than the 120 KBytes claimed in the manual, but still a small amount) of files within it.

The next point which must be noted is that the diskette used for installation is, and must be, write-enabled, as the installation program insists on writing back to the floppy disk. The manual explains at some length that the installation process writes PC-specific information back to the floppy disk, allowing disaster recovery.

I can think of no reason why this information could not be written to a blank diskette, which could then be stored in a secure place [*RG Software believes it is easier for users not to need a blank formatted diskette before beginning the installation process. Ed.*].

NoMore modifies AUTOEXEC.BAT and CONFIG.SYS, installing its own code as the first line in each of these files. A backup is taken of each of these files before any alterations are made.

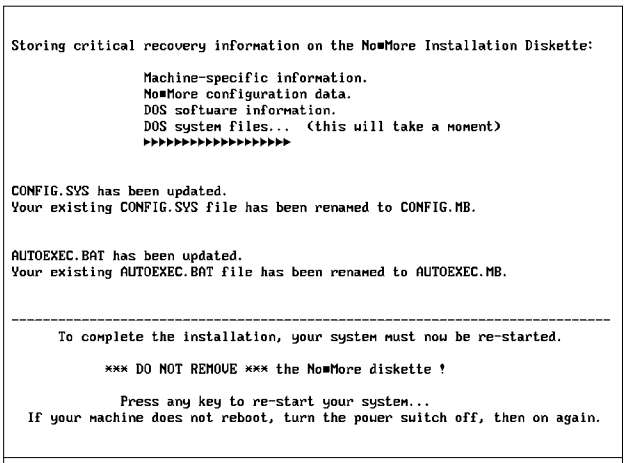
Operation

When a PC is rebooted with the product active, its position as the first line on CONFIG.SYS and AUTOEXEC.BAT ensures that it can perform its tasks before any other program has loaded. This location is enforced by *NoMore's* software. If the product's device driver has been moved away from the first line of CONFIG.SYS, *NoMore* will replace it at the start of the file. Problems would arise, however, if another product turns up which also requires to be in this special position.

By making a comparison with the 'snapshot' of the PC taken during installation, the *NoMore* device driver checks that none of the vital component parts of the PC's software have been altered. The manual also states that specific 'viral detection checks' are made; however, as (unsurprisingly) no details of the company's proprietary algorithms are provided, I cannot comment on their efficacy.

If no evidence of a virus infection is found, *NoMore* removes itself from memory, and the PC boots as normal. Nothing is left behind in memory – *NoMore* is not a TSR. All checks are performed in about five seconds (presumably less on a fast PC), making *NoMore* very transparent indeed; and thus less likely to be disabled by the user.

One customization feature offered by *NoMore* is to make it run silently, with no displays unless something is amiss – this, in tandem with the speed with which the product operates, could make it valuable for those who do not want their users even to see that their systems are being checked for viruses.



The installation process will only complete successfully if the floppy disk used for installation is write-enabled. Once this is done, recovery information is stored on the diskette.

Methodology

I tested how well *NoMore* detects virus infections by repeatedly infecting my test PC with various boot sector viruses. This PC was booted from an infected floppy disk, and then immediately rebooted from a floppy disk known to be free from viruses, which also contained a scanner (*Dr Solomon's AVTK*). Executing this scanner confirmed that the hard disk of the test PC really was infected.

If an infection was found, the PC was rebooted from the hard disk, and the action taken by *NoMore* was observed. Before the next infection was tested, the reboot from a clean floppy, followed by a scan of the hard disk, was repeated to check that *NoMore* had removed the previous infection.

Detection of Virus Infection

NoMore's virus detection capabilities were checked against eleven boot sector viruses; namely: AntiEXE, BootEXE, EXEBug, Form, Junkie, LZR, Natas, Stoned.NoInt, NYB, Quox, and Sampo.

In these eleven samples, *NoMore* always correctly detected the presence of the virus, and also gave a short description of the changes which had been made to one or more of the DOS boot record, the Master Boot Record (MBR), the partition information, and the command interpreter.

In all of these cases, *NoMore* was able to remove the virus infection successfully by restoring information about the hard disk preserved when the product was first installed. This result is impressive; a 100% hit rate.

Note that *NoMore* did not know which virus had affected the test PC; it merely knew that something had changed, and took the appropriate replacement action. Samples of whatever were thought to be virus infections are saved on the disk by *NoMore* as DOS files, and subsequent examination by a scanner (*SWEEP* from *Sophos*) confirmed that they were, in fact, virus infections.

NoMore also maintains a file called INCIDENT.LOG which contains details of all detected problems. The contents of this file can be examined using a utility which is provided with the product – this utility is placed on the hard disk at installation time.

Integrity Checking

I used the *Norton Utilities* to make various single-bit changes to the DOS boot record and the 'boot area' (I am not exactly sure what *Norton* means by this term). *NoMore* detected every modification that I made, and could also remove all alterations. I have no complaints about this.

In addition, when I made single-bit changes to the DOS command interpreter file (COMMAND.COM), *NoMore* always succeeded in detecting these changes when made to the copy of COMMAND.COM stored in the root subdirectory. This was accurate, but somewhat less than useful, as the PC was set up to use another copy of COMMAND.COM which was stored in the DOS subdirectory.

The problem is actually more complicated than it appears at first sight. I use a multi-boot system which sometimes uses a shareware command interpreter called 4DOS. This is contained within a file called (unsurprisingly!) 4DOS.COM. No matter what type of boot is performed, *NoMore* always simply checks the copy of COMMAND.COM stored in the root subdirectory. This file will be the one attacked by a direct system infector; however, the technique cannot be considered foolproof.

Damage

Given the above results, *NoMore* performs its claimed features very well. It really does detect any boot sector virus at boot time. Given that it can detect single bit alterations, it seems likely that the developer's claim of being able to 'Detect and provide no-hassle immediate repair for any boot virus, past, current and future' is very likely to be true. Personally, I would have toned down the use of the word 'any', as some smart virus writer could possibly find a way round *NoMore*, but that's a quibble rather than an objection.

Even so, there is a problem lurking behind all this seemingly limitless capability. Firstly (and this is made clear by the manual) the product does not provide a complete solution to the problem of file-infecting viruses. It does not attempt to hide this point, and the manual advises use of a scanner in conjunction with *NoMore* (*RG*'s own, of course!).

NoMore will, of course, spot multi-partite viruses, but only after they have dropped their boot sector portions. This fact is not made clear in the documentation, but is intuitively obvious from the method by which the product functions.

It could be suggested that checking only at boot time for the presence of a boot sector virus is insufficient. However, consider how a pure boot sector virus infects – it does so at boot time, if a floppy is accidentally left in the disk drive. By definition, immediately after the hard disk becomes infected, a reboot occurs, at which time *NoMore* should spot the infection.

In addition, *NoMore* cannot detect damage caused either to data or to executable files. However, neither can conventional scanners, and this is not *NoMore's* stated aim. Nonetheless, I am left with a slight sense of incompleteness.

Conclusions

Although this review refers to this product as *NoMore*, its official title is *No.More #*!$ Viruses*. The disadvantage with names such as this is that, although on first hearing they raise a smile, it is difficult to know how to refer to the product in everyday use. The developers would be advised to think about changing this name, as the joke wears thin after a while. Whoever wrote the product manual seems to agree, as the name reverts to merely *NoMore* on the fourth page.

The name, and the propensity to write back to the installation disk, are a pity, because *NoMore* does what it sets out to do very well indeed. It is true (currently!) that the majority of virus infections are caused by boot sector viruses: in all my testing, I did not find a single boot sector infection which *NoMore* failed to spot. Given the fact that it also spotted single-bit alterations (no matter where I made them), this is not surprising.

NoMore cannot prevent a hard disk from becoming infected with a boot sector virus, but it can spot that such an infection has occurred the next time the PC is rebooted. This works well against purely boot sector viruses, because (as was described above) the computer is in the process of rebooting when an infection occurs.

Overall, *NoMore* allows a 'hands-free' response to the current most common type of virus – what has become known as a 'Fully Automated Response' (FAR) – and as such will probably find a home in large organisations.

Technical Details

Product: *No.More #*!$ Viruses, v1.07.95.* (Currently v2.10.95, which runs on *Windows 95*, DOS, and *Windows*.)

Developer/Vendor: *RG Software Systems Inc*, 6900 East Camelback Road, #630, Scottsdale AZ 85251, USA, Tel +1 602 423 8000, fax +1 602 423 8389, BBS +1 602 970 6901.

Availability: Any PC running DOS version 3.0 or above. A hard disk drive with 120 KBytes of available space, and one floppy disk drive, are also required.

Price: $89.95 for a single copy. Corporate licences and disk distribution plans also available..

Hardware used: A *Toshiba* 3100SX; a 16 MHz 386 laptop computer with one 3.5-inch (1.4 MByte) floppy disk drive, a 40 MByte hard disk and 5 MBytes RAM, running under *MS-DOS* v5.00 and *Windows* v3.1.

The boot sector viruses used for testing in this review are: AntiEXE, BootEXE, EXEBug, Form, Junkie, LZR, Natas,, NYB, Quox, Sampo, and Stoned.NoInt.

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

Infosec, the UK's first dedicated information security show, will be held at the *London Olympia* (London, UK) from 30 April–2 May 1996. It is planned that **the programme will include conferences and seminars on topical security issues**. Information on attending or exhibiting is available from *Infosec* on Tel +44 181 910 7821.

Leprechaun Software Pty (Australia) has announced the launch of its ***Windows 95* anti-virus software**; *Buster for 95*. Further details are available from the company; Tel +61 7 3823 1300, fax +61 7 3823 1228.

The next round of **anti-virus workshops presented by *Sophos plc*** will be held at their training suite in Abingdon, UK, on 22/23 November 1995. Cost for the two-day seminar is £595 + VAT. Any one 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.

Online Data Recovery has announced the appointment of a new press agent: effective immediately, ***Harvard Public Relations* will be handling *Online*'s press and PR** in the UK. *Harvard* can be contacted on Tel +44 181 759 0005, fax +44 181 897 3242.

A company called *Greenscreen* has announced the launch of a new anti-virus package, ***InControl Virus*, which puts incoming disks through a designated controller** where they are scanned for viruses. Details from the company's Portsmouth UK base; Tel +44 1705 214127, fax +44 1705 214130.

Reflex Magnetics has announced the launch of a ***Press Virus Advice Line***. The facility is aimed specifically at journalists, whom *Reflex* sees as being a high-risk target for computer viruses, due to the multitude of electronic files they receive. To contact the *Advice Line*, telephone +44 171 328 1044.

The first virus to be named after a Taiwanese political party has appeared in that country: according to a report in *Computer Fraud and Security* (October 1995), New Party, while apparently a virus that destroys data on the hard disk, is in fact harmless [*! Ed.*], and can easily be erased.

On 13/14 November 1995, *S&S International* is presenting a further **Live Virus Workshop** in Buckinghamshire, UK. The two-day course costs £680 + VAT, and offers the opportunity to gain experience with viruses within a secure environment. Contact the company for details: Tel +44 1296 318700, fax +44 1296 318777.

***IBM* has now made its anti-virus products available on the Internet**. Subscribers may access the *IBM Anti-Virus* home page on http://www.brs.ibm.com/ibmav.html.

Precise Publishing Ltd has scheduled another **Live Virus Workshop** for Thursday 30 November 1995. The cost is £395, to include lunch and refreshments. Details from the company; Tel +44 1384 560527, fax +44 1384 413698, *CompuServe* 100043,2441.

***Integralis* has launched an Internet firewall** which controls access from exterior networks to internal corporate data systems. The company describes *PortMaster* as the easiest, most affordable way to defend any corporate data network, both externally and internally. Contact *Integralis* for further information; Tel +44 1734 306060, fax +44 1734 302143.

McAfee Associates has begun to distribute **free evaluation copies of its *VirusScan*** anti-virus software to users who wish to perform a system scan prior to installing *Windows 95*. This is intended to counter the fact that users with boot sector virus-infected PCs cannot install *Windows 95*. Users who wish to obtain copies should contact *McAfee* directly on Tel +1 408 988 3832 (US), or Tel +44 1344 304730 (UK).