ISSN 0956-9979 SEPTEMBER 1996

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

Editor: **Ian Whalley**

Assistant Editor: **Megan Skinner**

Technical Editor: **Jakub Kaminski**

Consulting Editors:

Richard Ford, Command Software, USA
Edward Wilding, Network Security, UK

IN THIS ISSUE:

• **In the public interest.** As is well documented, the Internet offers many opportunities for viral spread. However, it also provides mechanisms for anti-virus spread – for a brief look at the world of downloadable anti-virus software, see p.16.

• **Scanning the options.** Using multiple anti-virus products on your system has both advantages and disadvantages, and with the added complication of macro viruses, today's scanners are more likely than ever to have different opinions about files. Shane Coursen discusses what happens when scanners collide; on p.12.

• **Viruses for the point-and-click generation.** The evolutionary story of macro viruses continues apace – only last month we saw the first *Excel* macro virus. This month, Paul Ducklin describes the first two macro virus creation kits; turn to p.15.

CONTENTS

VIRUS BULLETIN **©1996 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /96/$0.00+2.50** No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

EDITORIAL

Zen and the art of anti-virus product marketing

August was an interesting month: several events have caused me to debate again a subject which I have already spent much time considering (and doubtless will again) – marketing. I have two main themes this month; first, virus distribution as marketing…

A persistent rumour has circulated over the past few months concerning *Second Sight UK Ltd* (distributors of *InVircible*); namely, that they have been sending virus samples to prospective customers. It proved impossible to verify the reports, so, being a straightforward soul, I decided to take the obvious route of calling Richard Macmillan (head of *Second Sight*) and asking him for the facts.

“ Michelangelo … had a beneficial long-term effect on vendors ”

His initial, baffling, reply to my query was: ‘Rumours aren’t generally true, are they?’; however, he soon proved to be even more straightforward than I. Yes, he said, he did send samples of in the wild viruses to his evaluation sites, if they had a quarantine environment. ‘Viruses in the wild are freeware,’ he said, ‘and don’t belong to you or anyone else’.

It is perhaps ironic that almost immediately after this conversation, I was shown a package sent by *Second Sight* to a company evaluating the *InVircible* software. Included was indeed a labelled diskette containing virus samples, together with a letter from Macmillan mentioning their presence.

Certainly, I for one would feel more than a little concerned if, as an MIS employee of a large organization, I were to receive from a prospective supplier of anti-virus software this disk, containing as it did Ginger.2774, NightFall.4518, Junkie, Manzon, Empire.Monkey (in the form of a dropper), Tremor, and Hare.7610. Macmillan states that users of *InVircible* have ‘no fear whatsoever of viruses’ – perhaps that’s just as well...

Hare brings me to my second topic; the time-honoured media favourite where viruses are concerned: blind panic. Readers will recall that in last month’s edition we presented analyses of two significant viruses: Hare and Laroux. Both of these have become very well known over the past month, and have caused not insignificant amounts of fear amongst the Internet-using community.

Laroux was discovered at the end of July. Anti-virus companies immediately produced the usual press releases, which, as usual, varied in tone from the calm and collected to the panicked and scare-mongering. By far the most singular of the bunch came from *McAfee*: it proclaimed that its researchers had discovered (untrue) the first *Excel* virus, and that its staff would be working ‘through the night to develop a Laroux detector’ (this may or may not be true; however, it certainly shouldn’t have taken all night).

Strangely, it was shortly after this that Hare (for whatever reason) rose into the public’s forever wavering attention – curious, as it first appeared at the end of May. As we were pursued relentlessly for quotes by the press, it became ever more difficult to avoid falling into the traps laid by artful journalists, who would like nothing better than to be able to print a ‘the sky is falling, and the earth has but 28 minutes until destruction’ story. Alas, they have done this anyway, regardless of strong urges to the contrary.

Fortunately, it is now regarded as less than constructive to kick up a huge fuss whenever new viruses appear. Michelangelo, whilst a washout in terms of living up to its panic, had a beneficial long-term effect on vendors; by and large they now shy away from such scare-mongering. Certain marketing people at *McAfee* have apparently not yet learnt this lesson.

Despite the fact that most of those who asked for quotes about Hare approached the topic rationally, we have found ourselves taken out of context time and time again. Why? Doubtless vendors will be lambasted for self-promotion when the great Hare scare comes to very little. No longer is it necessary for them overtly to cry havoc; even without the dogs of war, the mass media does its best to sell products for them. Were I to be even more cynical, I could easily believe that that was the aim all along.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

NEWS

Pricey Ludwig

Following other recent news stories in *Virus Bulletin* [*see 'Outlaws Revisited', May 1996, p.3 and 'UTR, the Folding Newsletter', December 1995, p.4*], Mark Ludwig's *American Eagle Publications* has announced the production of another book on computer viruses. Five hundred copies of *Computer Virus Supertechnology '96* will be printed, and will retail at US$395.00.

The book is said to detail 'two major trend-setting issues in computer virus technology'. The first of these is 'viruses in the 32-bit environment, particularly for *Windows 95* and *Windows NT*' (as usual, Ludwig presents virus source code), and the second is, bizarrely, the Internet Worm, for which full source code (which is in any case available at many Internet sites) is also included.

It is debatable whether or not the source code for the Internet Worm of November 1988 is particularly relevant for the modern Internet – it exploited several bugs in versions of software (notably fingerd and sendmail) which were in use at the time. These bugs are now the widest-known security holes in history, and as such should be found on virtually no systems in today's world ▮

Sophos Wins Quest for Growth Award

Anti-virus vendor, and sister company of *Virus Bulletin*, *Sophos Plc* has been awarded top place in the regional finals of the investment capital group *3i's* biannual 'Quest for Growth Award'.

The award is designed to identify and commend medium-sized independent companies which have demonstrated a solid growth record and show potential for continued success. The national final will take place on 8 October, when *Sophos* will compete with nine other regional winners for the national title.

The past few years have seen the company going from strength to strength, almost doubling its turnover and profits annually. New products have brought increased revenue; the latest of these is a new release of the anti-virus software *SWEEP for Windows NT*.

Sophos has also added a new course to its training and education portfolio. The one-day workshop, entitled Practical *NetWare* Security, will tackle the problems inherent in implementing *NetWare* security.

Continuing its expansion, the company's new premises are now almost complete: the purpose-built £1.5 million headquarters will house both *Sophos* and *Virus Bulletin* from September 1996. Visit *Sophos'* website for information on the company and its products; http://www.sophos.com/ ▮

Prevalence Table – July 1996

Virus	Type	Incidents	Reports
Concept	Macro	44	14.2%
Form.A	Boot	30	9.7%
Parity_Boot.B	Boot	26	8.4%
AntiEXE	Boot	24	7.7%
Junkie	Multi	17	5.5%
NYB	Boot	17	5.5%
AntiCMOS.A	Boot	16	5.2%
Empire.Monkey.B	Boot	11	3.5%
EXEBug	Boot	11	3.5%
Imposter	Macro	8	2.6%
Sampo	Boot	8	2.6%
Quandary	Boot	7	2.3%
Natas.4744	Multi	6	1.9%
Ripper	Boot	5	1.6%
V-Sign	Boot	5	1.6%
WelcomB	Boot	5	1.6%
Hare.7610	Multi	4	1.3%
Jumper.B	Boot	4	1.3%
Manzon	File	3	1.0%
She_Has	Boot	3	1.0%
Stealth_Boot.B	Boot	3	1.0%
Stoned.Angelina	Boot	3	1.0%
Telefonica	Multi	3	1.0%
Wazzu	Macro	3	1.0%
Bath	Boot	2	0.6%
Boot.437	Boot	2	0.6%
Burglar.1150	File	2	0.6%
Cascade.1701.A	File	2	0.6%
Defo	Boot	2	0.6%
Fat_Avenger	Boot	2	0.6%
Stoned.Manitoba	Boot	2	0.6%
Other [1]		30	9.7%
Total		310	100%

[1] The Prevalence Table includes one report of each of the following viruses: AreThree, Bandersnatch, Bye, Byway.A, Carnerali.1972, Colors, Colors.C, Da'Boys, Empire.Monkey.A, Feint, HDKiller, Hidenowt.1747, Ill.573, Kaczor.444, Kiev.483, Nomenklatura, One_Half.3544, Rhubarb, Satria, Stat, Stoned.LZR, Stoned.Spirit, Stoned.Stonehenge, TaiPan.438, Tentacle, Tequila, Trojector.1463, Unknown.1293, Werewolf.1500.B, and Yesmile.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

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

Type Codes

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

Br.1180

CER: An appending, 1180-byte virus which marks all infected programs with the characters ‘BR’, located in EXE files at offset 12h (the checksum field) and in COM files offset 03h. The virus contains the plain-text string: ‘Fr.COM.EXE’.

```
Br.1180            80FC FF75 03B4 FECF 80FC 4B74 03E9 1602 5053 5152 5657 061E
```

Cascade.1701.X

CR: Yet another variant, 1701 bytes long, of the Cascade virus.

```
Cascade.1701.X     012E F687 2A01 0174 0F8D BF4D 01BC 8206 313D 3125 474C 75F8
```

Chkbox.936

ER: An appending, 936-byte virus containing the plain-text string ‘sctbclf-fpc:\dos\smartdrv.exe’ and the encrypted texts ‘iamtheboss’, ‘checkboxports’, and ‘givegodmode’. The time-stamp on infected files is set to 10 seconds.

```
Chkbox.936         B803 63CD 213D FFFF 747F 9090 8CD8 488E D88B 1E03 0083 EB3E
```

Claire.821

CR: An encrypted, appending, 821-byte virus containing the text: ‘Claire J, my love for you is absolute. (C) 1993 Scorpio’. The payload includes corrupting the contents of a buffer during writing to a file (the virus writes the string ‘Claire’).

```
Claire.821         04A1 0101 0AC4 8DB6 0701 8DBE F003 3004 463B F776 F9E9 EBFE
```

Claudia.8772

ER: A stealth, encrypted, appending, 8772-byte virus containing the text: ‘Das ist ClaudiaSchiffer virus by OS’ and ‘Sometimes when U stroling down the Avenue the way U walk it makes men Thinkof HAVING U when U walking Down the street everybody stops & turns 2 stare at U! (IrM) Hope 2 meet U...somewhere in time’. If the current year is not 1996, the virus displays the above message and shows a picture of Claudia Schiffer. This virus is dropped by another virus; the WordMacro.Satanic virus.

```
Claudia.8772       BB?? 00B0 ??2E 3007 43F6 D881 FB?? 2276 F4??
```

Compost

PR: COM files created by the virus are in reality EXE PKLited programs. They are not hidden and are of different lengths. This is due to the random number of bytes attached at the end, outside the compressed code. The virus infects one file at a time, in a subdirectory chosen as a new current directory.

```
Compost            1000 FE03 E9A7 FE89 EC5D C300 2005 4558 4550 5351 5256 571E
```

CSF.558

ER: An appending, 558-byte virus marking all files (including COM) executed while the virus is active in memory with a time-stamp of 62 seconds. It also reinfects previously-infected programs.

```
CSF.558            B8BA FECD 213D EFAB 74D7 BB2E 02B1 04D3 EB83 C303 061E 0E1F
```

Drepo.2461

CER: A stealth, slightly polymorphic, appending (EXE), 2461-byte virus containing the text: ‘Pod na jedno DREPO!’ and ‘Shareware version. Do not forget to register!’. It infects COMMAND.COM and EXE files. While infecting COMMAND.COM, the virus overwrites its last 2464 bytes (usually filled with zeros).

```
Drepo.2461         03F5 2E8A 869B 098B CD?? 81C1 0800 2E30 0481 F??? ??4E 3BF1
```

EasternDigital.1700

CER: An appending, 1700-byte virus based on EasternDigital.1600. It contains the encrypted text ‘COMMAND.COMCOMEXEDOCHLPZIPCBACKUP.COM’, ‘*** you have been destroyed by Lee Pich alien v 1.00 ***’ and ‘*** warning !!!! Lee Pich is right here !!!! ***’. The virus can be detected with the template previously published in *VB* [*see August 1992, p.8*].

Enjoy.1667

ER: An encrypted, appending, 1667-byte virus which contains the text: ‘Welcome to the ultimate computer virus! This program is not destructive so I hope you’ll enjoy it. But don’t think this was it now... The evolution goes on!’. The time-stamp of infected files is set to 6 seconds.

```
Enjoy.1667         2E8A 4102 8BCF 2E30 4703 43E2 F983 C608 59E2 DA5F 5E5B 5958
```

Epsilon.1498

EPR: A 1498-byte (effective length) virus containing the encrypted text: ‘COMMAND’. Unlike other companion viruses, it creates COM files that are not marked as hidden and have different lengths (the virus appends a variable number of ‘rubbish’ bytes to its code).

```
Epsilon.1498       8916 2D02 8C1E 2F02 8C1E 3302 BA5C 0089 1631 028C 1E37 02BA
```

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

IBVV.742

CR: An appending, 742-byte virus, encrypted with a constant key, containing the text: 'Ich bin VAGINA Virus !!'. On 1 April, the virus overwrites the contents of the first physical hard disk.

```
IBVV.742            8BF4 2E8A 042C 302E 8804 4944 83F9 0075 EF8B E2C0 E935 302C
```

IVP.335

CN: An appending, 335-byte, fast, direct infector containing the plain-text strings: 'Th-Th' and '*.com'. All infected files are marked with the characters 'CA' located at offset 03h.

```
IVP.335             B440 B94F 018D 9605 01CD 21FE 8654 02B8 0157 8B8E AB02 8B96
```

IVP.336

CN: An appending, 336-byte variant of IVP.335. It contains the plain-text strings: 'Th-Th' and 'd*.com', and marks infected files with the characters 'CA' at offset 03h. It infects only programs which match the mask 'd*.com'.

```
IVP.336             B440 B950 018D 9605 01CD 21FE 8655 02B8 0157 8B8E AC02 8B96
```

Kobrin.492

CR: A prepending, 492-byte, fast, direct infector containing the encrypted text: 'Andy said:Zarin is dangerous!' and 'COMSPEC=*.com'. The first message is shown on 23rd of every month when the virus tries to overwrite first 40 sectors of drives A, B, C, D, and E.

```
Kobrin.492          B440 B9EC 01BA 14FD CD21 7213 33D2 33C9 B800 42CD 21B4 40BA
```

Mad.3732

CR: A polymorphic, multi-encrypted, appending 3732-byte virus containing the text: 'TME 0.0 (c)Black Angel 14/05/96'. Its code incorporates a large number of anti-debugging and anti-tracing tricks, and other malicious subroutines (e.g. overwriting of the partition table of the first hard disk). It cannot be detected using a simple template, and requires a generic/heuristic approach.

Mango.470

CN: An appending, 470-byte direct infector based on the older Mango.468 variant. It contains the same plain-text strings: '*.COM' and 'COMMAND.COM', and the characters 'AZ' located at the end of the code. The virus may reinfect previously-infected programs.

```
Mango.468           B440 B9D4 018D 9600 01CD 21B8 0042 33C9 33D2 CD21 B440 B903
Mango.470           B440 B9D6 0190 8D96 0001 CD21 B800 4233 C933 D2CD 21B4 40B9
```

Matthew.2667

CER: A prepending (COM) and appending (EXE), 2667-byte virus containing the plain-text message: 'Therefore I tell you, do not worry about your life, what you will eat or drink; or about your body, what you will wear. Is not life more important than food, and the body more important than clothes? Look at the birds of the air; they do not sow or reap or store away in barns, and yet your heavenly Father feeds them. Are you not much more valuable than they? Matthew 6:25 (Agnes) May 92' IICS-SLU B.C.'. It also contains the encrypted text: 'R.M.O.Ordona IICS-SLU'. The virus hooks Int 21h, Int 08h, Int 13h, and overwrites the boot sector of floppy disks. If the system is booted from such a diskette the above passage appears on a screen followed by the message: 'System Disk Needed...'.

```
Matthew.2667        33C0 8EC0 2680 3EBC 0100 0775 03E9 5001 E90D 018C C0FA 8ED0
```

Matthew.3044

CER: A prepending (COM) and appending (EXE), 3044-byte variant of the .2664 virus. The displayed quotation is shorter and, if shown, is enclosed in a frame: 'Look at the birds of the air, they do not sow or reap or store away in barns, yet your Heavenly Father feeds them. Are you not much more valuable than they? Matthew 6:26 1st Bat.94 (Sw & Sh)'. The virus hooks Int 21h and Int 13h, and overwrites the boot sector of floppy disks. If the system is booted from such a diskette the above passage appears on a screen, followed by the message: 'System Disk Needed...'.

```
Matthew.3044        33C0 8EC0 2680 3EEC 0387 7410 9090 2680 3EEC 0378 7406 9090
```

PowerOff.798

CN: An appending, 798-byte, direct infector containing the text: '*.COM', '*.EXE', and 'Power off immediatley... oh it is too late ! Your system is dead. -VAMPIR- See you later !'. The time-stamp on infected files is set to 62 seconds. The payload, which triggers on the 31st day of a month, on Wednesdays and Fridays, and every day after 11pm, contains procedures for overwriting a random number of sectors or corrupting EXE files.

```
PowerOff.798        B440 8BFA 2BD1 B91E 03CD 2190 7304 90EB 1E90 3D1E 0375 18B8
```

PSMPC.466

CER: An appending, 466-byte virus. When an infected file is run, the virus installs itself in memory and hooks Int 21h, even if it is already active. As a result, an infected system runs slower and the amount of available memory decreases with every execution of any infected program.

```
PSMPC.466           B440 BA00 00B9 D201 CD21 B800 4233 C933 D2CD 2159 B440 BAD6
```

Q.485

CR: An appending, 485-byte virus which marks all infected files with the character 'q' located at the end of code. The virus resides in the Interrupt Vector Table. It hooks Int 21h, intercepts Int 09h and Int 1Ch, and generates sound through the system speaker.

```
Q.485               3D00 4B74 03E9 9E00 B8C4 0DCD 602E 891E 8801 2E8C 068A 0152
```

Rogue.1206

CER: A stealth, appending, 1206-byte virus containing the encrypted text: 'DBF, 'CHKLIST' and '???TEDESVAMORTEQUIERODOROINIMAGINABLE 20-03-89 8:57PM'. The time-stamp of infected files is set to 62 seconds.

```
Rogue.1206          EA00 00FF FF06 1F2E 833D 0174 03E8 A200 B8BD 032B F82E 8B05
```

Romania.856

CR: An appending, 856-byte virus containing the plain-text string: 'COMMANDROMANIA' and '*.com'. Infected files have another string, 'ROMANIA', located at offset 03h.

```
Romania.856         891E 2704 8C06 2904 B813 25BA C803 CD21 0E1F 8E06 2C00 33FF
```

VIRUS BULLETIN **©1996 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /96/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

INSIGHT

Marking Time

With a name so reminiscent of an ancient Classical hero, what is a man to do but make his own name echo in his chosen field? Despite his youth, Marko Helenius is doing just that. Born in a remote corner of northern Europe in 1969, Helenius has made his hometown of Tampere, Finland also the base for his research into computer viruses.

As with so many of his peers, computers have been a part of his life since childhood. Although he does not come from a family background in computing, his first computer was also his first major purchase: at the age of 14, he bought an *SVI-328* (at that time, the main rival of the *Commodore 64*), because he was 'excited at the possibilities it could provide', and viewed it as the perfect medium for playing games, a passion which he still indulges.

Within weeks of beginning, however, he was writing embryonic programs for the machines. The *SVI-328* lent itself to programming, and he was soon a 'convert to the cause' – 'These early programs were mostly games for my own use,' he explained.

He still has the games, and sometimes plays them or shows them to friends: '…but not very often – maybe twice a year.' By his own admission: 'Computing became an obsession for me then, and it still is today.'

Studying the Life Forms

1996 sees Helenius, still in Tampere, pursuing research at the university there, where he had begun a degree in computer science in 1990. He completed his Master's degree, also at Tampere, just two years ago.

It was a chance exposure to a *Macintosh* virus which led to his interest in computer viruses: 'It was in the spring of 1989,' he recalled, 'and a friend and I were doing a programming exercise on the *Mac*. A couple of weeks after we had left it for evaluation, the examiner informed us that the program was infected with nVir.A, and that it had spread to other disks which were also being evaluated at the time.

'The virus had probably come to us from a computer to which one of the University's first laser printers was attached, and it spread quickly around the University until automatic protection was installed.'

He concluded his description of the incident somewhat ruefully: 'In other words, my first exposure to a virus was unconsciously spreading one myself.'

Needless to say, Helenius immediately began to implement anti-virus precautions: the fortunate outcome of this incident was that he quickly established himself as being knowledgeable about viruses. Subsequent to the incident, he took a course in computer viruses, prior to writing his Master of Science thesis. The area captivated him, and resulted in a decision to use the arena as research for his licentiate. His Master's thesis eventually took as its subject computer viruses and virus prevention.

Coming Up

The buzz-word of 1995 in the anti-virus industry was the macro virus: Concept unleashed this, in a blaze of publicity. Helenius is firmly convinced that these infectors are here to stay, mostly because they are easy to alter: 'I think, however, it would be an exaggeration to describe them as a lethal threat.

'The possibility does exist,' he explained, 'for destructive macro viruses: the macro language in many cases would allow such opportunities. Nevertheless, these viruses are simply another new genre. They can spread very effectively, but with the correct countermeasures, a macro virus can be prevented just like any other virus.'

"in my opinion it is unethical to write viruses, even for research purposes"

Polymorphic viruses, on the other hand, he views as a problem, particularly for anti-virus producers: 'There are a lot of them in the wild, and at least 20% of the viruses on Joe Wells' *WildList* are polymorphic,' Helenius commented.

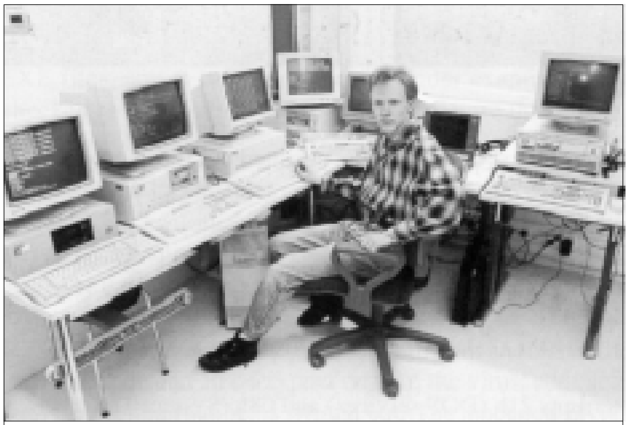
'The main problem with polymorphic viruses,' he continued, 'is that it takes a great deal of effort to disassemble them. If they have not been well enough disassembled, they cannot be detected and removed reliably. These viruses can also precipitate false alarms from scanners.'

Heuristic techniques, according to Helenius, are used to a certain extent by most, if not all, anti-virus packages: 'Whenever a scanner detects an unknown virus, it is using heuristics. All scanners I have analysed claim to detect at least some unknown viruses.

'I think all virus-non-specific techniques (heuristics, behaviour blocking, checksumming) have their advantages and disadvantages, when compared with the more traditional scanning techniques. Whether or not they are useful in any given situation depends on how well the relevant technique is implemented.'

It would be difficult, in Helenius' view, for anyone to set up an anti-virus company from scratch in the industry as it stands: 'It would not be impossible, but it would take a huge

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.



Testing times: Marko Helenius, surrounded by the toys of his trade.

amount of effort, and anyone who wanted to do this would need already to have established good relations within the anti-virus world.'

To Write or not to Write

Helenius does not have much time to spare for virus writers; in his opinion the law of every country should contain a section pertaining to the distribution of malicious code – he cited the Swiss, who specify that any such laws should not make anti-virus research illegal.

Although he believes that, if someone is convicted of distributing virus code intentionally, legal action should be taken, he views the issue of computer crime as being similar to other misdemeanours – difficult to eradicate.

'Something could probably be done,' he said. 'I don't believe that many virus writers are aware of the potential consequences of their actions, and I see anticipative education as one solution.

'When it comes to operating systems,' he continued, 'I think there are ways to implement much safer systems than the ones which we use today. The problem with the current operating systems is that they are not built with safety in mind; particularly for PCs.

'In a PC, every program can do almost anything the programmer desires; write over other programs (or entire disks), do things to the OS memory (if not running in a protected mode); read important information from files or the keyboard – passwords, for example. We can only trust that every program we run works as it should work.'

Helenius has met what he terms a 'potential virus writer' at an exhibition: 'This guy asked me for material which he could use to help him write viruses. I did try to persuade this person that writing a virus wasn't worth the consequences – in my opinion it is unethical to write viruses even for research purposes. I have never even written a self-replicating program.'

According to Helenius, there are currently four or five known active virus writers in Finland, who, he thinks, may even have had their names published. He is unaware of the direction of their current activities – they are not, he hopes, writing new viruses.

Research and Analysis

Helenius is currently senior researcher at the *University of Tampere's* Virus Research Unit: 'In fact, right now I'm the *only* researcher,' he smiled. The work involves tracing virus code within files, and searching for virus-like behaviour in suspect files. His other tasks include analysing various anti-virus products.

A specialist in PC viruses (despite his early experiences with the *Mac*), he has also developed methods for automatic virus replication (both boot sector and file viruses) and automatic analysis of memory-resident scanners, for which he uses a system he calls Automatic and Controlled Virus Code Execution.

He first presented the system at *Eicar 95*, the annual conference of the *European Institute for Computer Anti-virus Research*, which took place in Switzerland in November 1995. (A paper discussing his research is available on the Research Unit's WWW site; http://www.uta.fi/laitokset/virus/.)

He sees himself remaining in anti-virus research for the foreseeable future, although he views the coming years as still 'quite open' for him: 'It all depends very much on what opportunities the future will give,' as he explained. His next main goals are to complete his licentiate, and then to obtain his doctorate.

Delving Deeper

Although passionate about computers, one could not describe Helenius as one-sided: he is engaged to Minna, a student at the *University of Industrial Arts* in Helsinki. Minna's career plans are very different from those of her fiancé – she will be a qualified art teacher when her studies are complete. However, she does share one part of his PC-mania – both are avid computer-game players, with a special interest in challenging role-playing games involving complicated problem-solving (their current front-runners are Eye of the Beholder I, II, and III; Lands of Lore; and Ultima Wonderworld I and II).

When not tearing apart anti-virus packages or disassembling viruses, Helenius enjoys watching movies and seeing friends, but he also confesses to doing some programming in his spare time. 'My interest in computer viruses is an extension of my interest in computers: I find all the tasks related to virus research challenging. As long as this remains the case, I see myself staying in the field.'

With so many minor achievements already to his credit, it will be interesting to see what the future holds for Helenius. One thing is certain: on the foundation he has created, the coming years will be more than productive.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

VIRUS ANALYSIS 1

Ground Control to Major Tom

Kevin Powis
Precise Publishing Ltd

Major.1644 is a memory-resident EXE file infector. When an infected file is executed on a PC, the virus decryptor receives control, and enters a convoluted loop which decrypts the rest of the virus.

Viruses use encryption to prevent scanners finding a fixed sequence of bytes to use for searching. This procedure does work as intended with Major.1644, but its decryption routine is quite lengthy and fixed, and can therefore be used as a search-string.

The Major Moves In

Once the virus code is decrypted, further examination screams out 'Virus': the first instruction after decryption is an 'Are you there?' call via Int 21h, AX=ABCDh.

The expected result of 1234h in the AX register will be returned only if a copy of the virus is already loaded. If so, the virus need do nothing further. The starting point of the host program is retrieved from the virus body, and control passes to that address, allowing the host to run as normal.

If the return value is other than 1234h, the virus must place itself in memory. Major.1644 does this by taking one of the segment pointers as initialized by DOS when loading the host program, and decrementing it. This allows the virus to examine the MCB associated with this segment.

The MCBs, or memory control blocks, are linked by a set of chained pointers in memory. The first byte of any MCB indicates whether or not this particular MCB is the last in the chain. The virus walks the chain until it reaches the last MCB: if it reaches the video memory segment before reaching the end of the MCB chain, it aborts the attempt to go resident. Once the end of the MCB chain has been found, the MCB is directly modified.

Major.1644 reduces the amount of memory which is associated with the last MCB by just over 29KB. When I first worked that figure out I had to check it – I cannot see any reason for reducing memory by this vast amount unless the programmer did not realise that the values stored in the MCBs are in paragraphs (multiples of 16 bytes).

Once the MCB is modified, the virus has made a fairly large hole in memory in which to reside. It then copies 1643 bytes (*not* 1644) of its own image to the new location in memory. The next task is to prepare to become resident and to hook the virus code into the system interrupts, which will allow the virus to gain control at appropriate times.

The virus first uses the undocumented but well-loved function to obtain what TSR programmers call the INDOS flag. This is a pointer which TSR programs can use to see at a later time whether or not it is safe to interrupt DOS. DOS maintains this flag and sets it to 1 when it is in an unstable state and does not want to be interrupted. Major.1644 calls Int 21h, AH=34h, which returns a pointer to the flag.

Interrupts 21h (DOS services) and 08h (System Timer) are then hooked by the virus. The original contents of both of these vectors are used to build far calls to the original routines for later use. The vectors are then amended to point to the virus handlers, at offsets 24Ah and 407h in the virus body respectively.

From that point on, the virus interrupt handlers are active, so all that is left for this transient copy of the virus to do is to allow the host to run, which it does. All other processing for the virus is controlled by the two interrupt handlers mentioned above.

> *"the virus does not employ any stealth technology, and all increases in file size … will be evident in the directory listing"*

Interrupt 21h (DOS Services)

When this handler is invoked, the virus checks the AX register. If it is set to ABCDh, Major recognises the call and simply sets AX to 1234h to indicate that it is in memory, and then returns. The only other function of interest to the virus is 4B00h (Load and Execute Program).

If the function is not of interest, the virus passes control via the far call it constructed at installation to the original DOS handler. There are therefore no stealth capabilities, and all increases in file size and changes in file content will be visible in the directory listing.

In the case of the 4B00h function, the handler will set about infecting the target file before allowing it to run as normal. Major.1644 next obtains and saves the file's attributes, then clears them, which allows read-only files to be infected. Following this, a critical error handler is installed, in an attempt to prevent those nasty DOS error messages which occur when DOS detects a hardware error (for example, write-protected disks).

Due to programmer error, this does not work as expected. Therefore, the tell-tale sign of a 'Write Protect' error occurs when using write-protected floppies on a PC which is infected with Major.1644.

Before this, however, the virus reads in the first 18h bytes from the target file. The virus is only interested in EXE files, and at this stage the file under examination is checked to ensure that the first two bytes are the EXE signature (4D5Ah).

If this test is passed, the virus checks to see if the file is already infected, examining an unused entry in the EXE header called the CHECKSUM field. If the file is infected, this field will contain the hexadecimal value DEADh.

If the file is uninfected, the file pointer is then moved to the end of the file and an encrypted copy of the virus, complete with decryption routine, is written out. The amended EXE header, containing the DEADh signature, is written back. The file's date, time and attributes are then restored. Finally, the critical error handler is unhooked and the new host – which is now infected – is allowed to run.

Interrupt 08h (Timer Tick)

The Int 08h handler will receive control eighteen times per second: this is a hardware interrupt which is used by the timer chip routines.

The virus handler begins, as do all good TSRs, by using a far call to service the original timer routine. Then Major.1644 retrieves a word from low memory – this again is part of the timer functionality. This word is incremented approximately once an hour by the real time clock.

The virus checks the current value: if it is not 2, the virus does nothing, and the handler processing is complete. This word will actually have a value of 2 so infrequently that I believe it is intended to allow the programmer to switch this routine on and off manually.

If the value 2 is found, the virus then checks its own internal flags, which would indicate if either of the virus handlers were already active. If any of these or the system INDOS flag are set, the virus cannot continue, and the handler will end its processing. Assuming all these flags are clear, the virus immediately sets its own internal flag to indicate that the Int 08h handler is now active and therefore prevents recursive entry to this part of the handler.

Before continuing, a new stack is allocated, and the critical error handler is installed as described above under the Int 21h handler.

The virus now enters program logic that can have no use or purpose for any PC other than the virus author's own. The virus looks for the presence of two files in the current drive, BBSV6\BBSAUDIT.DAT and BBSV6\BBSUSR.DAT. These file names correspond to the following text, visible in the virus after decryption:

```
The Major BBS Virus created by Major tom
```

If the relevant files do not exist, the interrupt handler's work is complete, and control of the PC is returned to the user. If the files are found, they appear to have a strict format, as the virus reads in a record count first and then processes the records in the file one at a time. For each record, Major.1644 attempts to match against the user names which are hard-coded in the virus. These names are Puppet, Image, Gnat, Minion, Cindy, and F'nor.

When the virus finds a record for a user with a username matching one of these strings, it makes a small modification to that record – it writes a '1' over the first byte.

Unfortunately, it has not been possible to figure out what effect this has – without access to the file format specifications of these particular files, which are part of any *Major BBS* version 6 setup, it's impossible.

[*In spite of this, some educated guesses may be made. It is, for example, reasonable to assume that the virus is attempting to increase the access privileges of matching users in some way. Perhaps a '1' in that particular location in the record indicates that this user is a SysOp?*

The theory would then be that the author, or any of his friends, could create a user on a BBS, and then upload an infected file to it. If the real SysOp tests the uploaded file on the same machine, it will (sometimes) change the access rights of the author's account to grant him full access to the whole board. Ed.]

Summary

This virus contains little which would set it aside from most others. It should be easily detected, even despite its inbuilt encryption, and infects easily and rapidly; therefore, in tandem with the fact that Major.1644 is loose in the wild, it must be considered a threat.

Major.1644

Aliases:	None known.
Type:	Resident EXE file infector.
Infection:	EXE type files .
Self-recognition:	
	Value DEADh in the EXE header checksum field.

Hex Pattern:	This pattern will locate the virus in files and in memory.
	`1E33 DB0E 8BC3 03C1 1F8B D18B`
	`F38A 8730 008B D080`

Intercepts:	Interrupt 21h DOS handler; Interrupt 08h Timer Tick.
Trigger:	None.
Payload:	None.
Removal:	Delete infected files and recover from a backup.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

VIRUS ANALYSIS 2

Welcome Back!

Richard Ford
Command Software

Even with the explosive growth of the Concept virus, a cursory glance down the Wells *WildList* reveals that boot sector viruses still make up the vast majority of those in the wild. Why this type has become so successful is a mystery, especially given the simple preventative measure of disabling floppy boot. Regardless, boot sector viruses continue to spread, and a large number of the diskettes sent to me from concerned users turn out to be relatively simple boot sector viruses; WelcomB is a typical example of this genre.

Moving In

WelcomB's life-cycle is depressingly familiar: an infected diskette is inadvertently left in the A: drive, and the machine rebooted. When the BIOS bootstrap loader loads the first sector on the diskette, control passes to the virus code.

Installation is somewhat more convoluted than for most viruses of this ilk. Rather than decreasing the total amount of memory under 640K, and copying its code into this 'hole' between high and low memory, the virus loads its second sector into memory, at 0000:0600h.

This second block of code first obtains the Int 13h vector, then decreases the top of memory by 2K. Next, the two virus sectors are copied to upper memory, and control passed to the high copy by pushing the offset and segment onto the stack and issuing a RETF. The destination address is PUSHed directly (PUSH immediate – opcode 068h), not via a register. This instruction is only supported by 80188 processors and above; thus the virus will not replicate on XTs.

The virus then recreates the original boot sector – this is probably its most interesting feature. Most boot sector viruses keep a copy of the original boot sector, which can be loaded from disk and executed. WelcomB stores half the original boot sector in its own first sector, and the other half in the second. This makes it impossible to disinfect the virus by copying an entire sector from elsewhere on the disk; rather, the original boot sector must be rebuilt by combining the two sectors containing the virus code.

Next, the virus checks an internal flag, which shows if the copy being executed was stored on the fixed disk. If not, the virus loads the first physical sector of the first fixed disk into memory, and checks whether it is already infected by comparing the first word with the first word of its memory image.

If the fixed disk is deemed uninfected, WelcomB infects the in-memory image of the MBR (preserving the partition table stored within) and writes it back to the disk. The virus' second sector is then stored to the second physical sector. Finally, the virus loads the original boot sector and passes control to it.

Resident Operation

The memory-resident operation of the virus is extremely simple, as it contains no attempt at even basic stealth operations. Whenever an Int 13h is issued, the virus checks to see if it is a call to access a fixed disk – if it is, the call is allowed to proceed.

All accesses to any diskettes cause the virus to check them for infection. The boot sector is loaded into memory, and the first instruction compared to that of the virus. If it matches, the virus assumes that the disk is already infected, and allows the call to pass to the BIOS. If there is no match, the virus checks the media descriptor byte. If this is FDh (indicating a 360K, 5.25-inch disk), the virus uses head 1, cylinder 0, sector 3 in which to store its second sector; in all other instances, head 1, cylinder 0, sector E is used.

Conclusion

WelcomB is an unremarkable virus, though surprisingly effective. Perhaps its most notable feature is that it is so easy to prevent. Most BIOSs allow the user to choose the boot sequence in advance. For most machines, especially *Windows 95*/*Windows NT* workstations, there are very few occasions when one needs to boot from drive A.

The best protection against this virus does not come from anti-virus software, nor does it cost money… change one CMOS setting, and voilà!

WelcomB

Aliases: Bupt, Beijing, BuptBoot, Bupt1946.

Type: MBR and boot sector of diskettes.

Self-recognition in Boot Sectors:

Checks the first word in the sector for the value EB39h.

Hex Pattern:

```
B801 02B9 ??00 BA?? ??BB 0006
CD13 C606 1D7C 00EA 4D06 0000
```

Intercepts: Int 13h for infection.

Trigger: None.

Removal: The usual FDISK /MBR after a clean boot. It is difficult to repair by copying the original MBR or boot sector back.

VIRUS ANALYSIS 3

Touching the Tentacle

Dmitry Gryaznov
S&S International PLC

At the end of March 1996, *Windows* users worldwide began to notice certain problems with some *Windows* applications. The applications concerned grew in size, and took somewhat longer to start. Some of the applications stopped working properly or did not run at all. The most noticeable victim was WINHELP.EXE, one of the most-often-used *Windows* applications, which stopped working completely.

Several customers reported these problems to the technical support department of *S&S*. They were asked to send in the affected files, which were passed to me for analysis: examining the applications under a file editor revealed the string 'TENTACLE.$$$'. Sure enough, it was a new virus.

Tentacle Takes Hold

This new virus, which has been named Tentacle, is a non-resident direct action infector. It infects *Windows 3.x* applications which are of New Executable (NE) format; that is, most of the programs designed to run under *Windows 3.x*.

When an infected program is run, the virus takes control. It then searches the current directory for any files with an .EXE extension. If an uninfected NE file is found, Tentacle infects it. Then it proceeds to C:\WINDOWS the *Windows* home directory, and infects up to three more files.

Infected files increase by circa 1966 bytes. To distinguish between infected and non-infected files, Tentacle uses the MaxMem field (the two-byte word at offset 0Ch) in the EXE header of DOS stub[1] of a *Windows* application. Tentacle sets this field, normally FFFFh, to FFFEh. Replication now complete, the virus passes control to the host program.

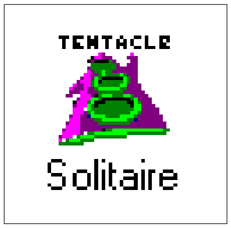
The structure of NE files is significantly more complex than that of DOS, COM, or EXE files. It is thus not always possible to infect such a file simply by appending virus code to the end of the file and then patching its beginning to ensure that control passes to the virus when a program is executed. Most DOS viruses infect their victims like this.

To eliminate this problem, Tentacle creates a temporary file, C:\TENTACLE.$$$, in the root directory. There it builds an infected file from its victim and the virus code, making the necessary changes to the NE file layout. The original is then replaced with the newly-created infected file.

Trigger

If a file is infected between midnight and quarter past midnight by the system clock, the virus triggers. All files infected within this fifteen-minute interval have their main icon changed to one resembling an octopus' tentacle (see image below). To eliminate any possible doubts as to what exactly is pictured on the icon, the virus author took care to include the text 'TENTACLE' in the image.

This change of icon, however, is not immediately obvious, not to mention the fact that not that many computers are in use at midnight. The tentacle icon does not show up until the infected program is minimized whilst running, or until such time as it is removed from a program group and placed in the same or in another program group. The icon can also be revealed using File/Properties/Change Icon from the Program Manager menu.



Bugs

As if the author of Tentacle wanted to prove again that there is no such thing as a harmless computer virus, he incorporated several bugs in the virus code. Because of those bugs, not all infected *Windows* applications can run properly. As mentioned above, WINHELP.EXE becomes unusable when infected with Tentacle. This fact alone makes this virus very noticeable, and it is exactly the reason that many users noticed its presence on their computers.

Far-reaching and Fast

How is it possible that such a non-prominent and fairly obvious virus succeeded in infecting thousands of computers worldwide in a matter of a few days?

The speed with which Tentacle has managed to spread is even more astonishing when one realizes that many computers, in different countries and on different continents, were infected within only a few hours. People do not travel with computers that quickly; nor could regular mail deliver infected diskettes with such speed from one infected location to another.

This puzzle is easily solved: Tentacle has flown so far courtesy of the Internet. A program infected with this virus was first uploaded to a Usenet newsgroup in March of this year. To make it more interesting, the newsgroup was alt.cracks, where people discuss the ways to crack and bypass different types of software protection, in particular the restrictions in evaluation and shareware copies of commercial software.

DOGZCODE.EXE, the program in question, supposedly provided anybody running it with a secret code to enable all the features of a fairly popular (and quite cute, I must admit!) program 'DOGZ, Your Computer Pet™' without paying money for a licence. Thus, using such a program constituted a copyright violation.

This may explain why so many people with Tentacle-infected computers would neither report their problems, nor admit to downloading and running DOGZCODE. We received only a few proper reports of the virus, but dozens of anonymous ones.

The Internet is becoming more and more popular nowadays. Unfortunately, it is also becoming more and more popular with virus writers and distributors. Through the Internet, especially by means of its worldwide discussion forums and Usenet newsgroups, viruses are able to travel much faster than used to be the case in the ‘good ole days’.

It is not only the Tentacle virus which has spread in this manner: several *Word* macro viruses have also been distributed all over the globe in a matter of hours. The most recent example is the Hare virus: a number of programs infected with variants of Hare were posted to popular newsgroups, including alt.software.shareware, alt.cracks, and alt.sex.

Conclusion

So, be careful what you download! Do not run anything without first checking it for viruses, and ensure that your anti-virus software takes care of your downloads.

This does not necessarily mean the software should be Web- or email-specific, or even Internet-aware. A decent DOS TSR scanner, or better still, a *Windows* VxD scanner, should be able to intercept known viruses in any downloaded program before it gets executed.

[1] A DOS stub is a small DOS program which is prepended to all New Executable files. The program normally just displays something like ‘This program requires Microsoft Windows’ and terminates when a *Windows* application is run under DOS rather than under *Windows*.

Tentacle

Aliases: None known.

Type: Non-resident, direct action infector.

Infection: *Windows 3.x* applications in New Executable format.

Self-recognition in Files:
FFFEh in MaxMem field (the two-byte word at offset 0Ch) in EXE header of DOS stub of *Windows* applications.

Hex Pattern:

```
1E60 0E1F 81EC B700 8BEC B41A
8D56 001E 161F CD21 1FBA 2300
```

Trigger: Time between 00:00 and 00:15.

Payload: Icon of infected file changes to an octopus’ tentacle.

Removal: Delete infected file; replace with known clean copy.

FEATURE 1

When Scanners Collide

Shane Coursen

From false IDs to failed repairs, interpreting and interpolating information gained from using multiple anti-virus programs can sometimes be abstruse. Being alerted to the presence of a possible computer virus shouldn’t make you run for cover… or technical support.

The ability to distinguish between a computer virus and something other than that is especially valuable in the business world of today, and the use of two or more scanners just might give you that ability. It could mean the difference between a productive day and one filled with confusion.

Using Just One Anti-virus Product?

While using one scanner can provide a strong defence against viruses, common sense tells us that using more than one provides an even stronger data defence. In fact, anti-virus forums on *CIS* and *AOL*, and Internet newsgroups such as comp.virus, tell a story of MIS and other people doing exactly that. As we will see, however, when employing multiple scanners, each with their own feature sets – both documented and undocumented – mixed or conflicting reports are not rare.

This might lead one to believe that using more than one scanner is a bad idea. Not at all! Thankfully, the advantages outweigh the consequences of insufficient protection. There are obvious benefits of relying on more than one manufacturer’s product. The most apparent is that by doing so, you provide a finer net – a comprehensive binary mesh, if you will – that most computer viruses are unable to permeate.

If your anti-virus program indicates a computer virus, what should the next step be? If the infection is real, repair it. This is a simple undertaking and nearly all virus incidents conclude in exactly the same manner. If you detect a virus, but cannot repair it with one scanner, you should suspect a false positive. To verify the existence of this possible virus, you might try another scanner. If further scanning yields similar findings, it is more than likely that you have a virus. Next, you must find a way to repair or replace the infected areas.

There are instances, however, where the resolution is not as easy as using just two products. What would you think if, for example, you were witness to the following events: Product X detects a virus and is unable to repair infected files. Product Y detects the same virus, and is also unable to repair files. Product Z, which is known to detect and remove the virus reported by products X and Y, sees nothing at all.

To those who find themselves in this predicament, there is strong evidence of a computer virus, but still there is no way of being absolutely certain, and there is no way to remove it.

From the viewpoint of the unfortunate soul in the previous example, life is truly unkind. Three anti-virus products, all capable, and still no resolution! Granted, this example is a bit extreme. It is rarely necessary to use more than three products just to isolate a computer virus; the point is, however, that their availability proves an effective tool.

Repair versus Replacement

So far, we can see that there are advantages to using several products as tools to isolate a computer virus. One benefit is confirmation: using a second scanner gives an end-user the ability to confirm (or refute) the existence of a computer virus. The second case is just an example of futility. Although we never get a final or decisive answer, with each successive scanner, the answer comes a little closer.

When it comes to eliminating a computer virus, it is a contest between confidence and convenience. Confidence is the undeniable annihilation of a computer virus by means of removing or replacing infected files, or rewriting the boot and/or any other affected areas. Convenience is the ability to manipulate existing data, and restore it to its original form.

> *“with macro viruses, the preference of repair over replacement becomes less of a nicety and more of a necessity”*

While one cannot dispute the guaranteed integrity that replacement affords, one also cannot deny its costly inconveniences. Time to restore, time to set up to restore, finding the latest backup set (and wishing there were ‘just one more’ of a later date) – all this takes time. Compare that to the efficiency of an anti-virus product with the ability to repair the infected areas. With the click of a mouse, or the addition of a switch to the command line, the amount of time to recover from a computer virus is reduced to just seconds.

In a brief Q&A session at the 1995 *Virus Bulletin* conference, the subject of repair vs. replace was brought up. Most favoured replacement. Why such a bad rap for repair? After repair, life simply moves forward; no worries, no concerns – unless, of course, the repair didn’t go as planned. Documented occurrences of failed repairs are a heavy stigma in the history of repair; resulting in a lack of confidence in the repair capability of scanners. Due to some unforeseen event, repair fails, leaving the operator as fragmented as the data.

Catastrophes due to a failed repair rarely[1] occur: scanners take precautions, and can distinguish an ‘acceptable’ repair versus a failed repair before anything is committed to disk. Ironically, it is the subtleties (of a failed repair), not the catastrophes, that tend to make ‘repair’ a bad word.

One disadvantage is that, in using more than one product, the doors of scrutiny are opened. If any defects in the repaired file exist, they are immediately visible with a simple file comparison. Similar to writing any computer program, what one person authors, another will surely claim to be able to do better!

Problems with Repairing (Standard File Viruses)

To illustrate, a quick view of the classic repair problems may help. To restate: what one product (technically correctly) repairs and considers clean may not conform with another product’s standards of cleanliness.

Classic example 1: A standard file virus infects a COM file, modifying the first three bytes, appending 512-530 bytes. Scanner X is set to detecting and repairing. A post-repair file comparison reveals that, while the first three bytes are restored to their original state, and 512 bytes are clipped from the end of the infected file, there is a flaw. There is still (not always) 1-15 bytes of viral code at the end of the file; bytes scanner Y will consistently detect as a virus. Which scanner is at fault?

Classic example 2: A standard file virus infects an EXE file, modifying several header bytes, appending a variable amount of code. With results not so different from its counterpart, scanner Y leaves extra bytes at the end of the repaired file. Also, several bytes in the EXE header do not match the original uninfected file. Scanner X reviews the work: the extra bytes cause no adverse affect, and scanner X seems to approve. Later that day, however, when scanner X’s TSR encounters the file, an alert is generated. Which scanner is at fault?

In the first example, an analogous equivalent would be like saying the repaired file is all dressed up (the extra code), but has nowhere to go. (Since the repaired file compares byte-for-byte from beginning to end against a known clean copy of the same file, there is no code path to the extra code; thus, the extra code will never have an opportunity to execute.) The repaired file runs, but because of the extra code at the end, there is the chance of a ghost positive.

In the second example, there is also a logical, albeit more technical, explanation. Of note is the common location of the differing bytes. Looking at different repaired files, you might notice that bytes 2h and 3h, and sometimes 12h and 13h, differ from the original file. The differing bytes reside in the EXE header: while they may not all match the original file, they may represent more accurately the state of the repaired file[2].

In shop talk, bytes 2h and 3h combine to form a word indicating the number of bytes in the file’s last page. If a file, post repair, is larger, it is likely that the product modified the EXE header to reflect (correctly) the increase. Good, but at the same time, not good according to someone else.

Again, we see that one or both scanners, or neither, is at fault. One product manufacturer may contend it is the fault of the other, both may deny any wrong-doing, and both can easily create the illusion they did nothing wrong. Happily for COM and EXE repairs, such cases are rarely seen any more. Over the years, various standards have been set, and the unprecedented cooperative efforts between anti-virus researchers have resulted in the elimination of such conflicts.

Macro Virus Repairs

Macro virus repair problems are similar to the case of a standard file virus in that while repair is technically correct, the repaired file may not compare byte-for-byte to a known copy of the same file. For reasons of brevity, a detailed look at the structure of a *Word* document will not be covered.

Since the accidental release of Concept, there have been 60-70 similar variations of the same 'point'. Of these, Boom, Imposter, and Wazzu have all made their point public. Now, one year after *Virus Bulletin* first reported Concept – and right on schedule with most researchers' expectations – there is a new concern: a fully functional *Excel* macro virus.

For many years, macro viruses were anticipated, but was the anti-virus industry fully prepared for the complexities which lay underneath proper detection and removal? Most anti-virus companies immediately looked to *Microsoft* for help. After all, *Microsoft Word's* DOC/DOT was the target, and this type of file is much more than just simple lines of text.

> *"documented occurrences of failed repairs are a heavy stigma in the history of repair"*

Taking time to meet with anti-virus companies, *Microsoft's* response was swift and decisive; it was the implementation that might take some time. In the interim, certain guidelines could be followed to slow its propagation. With users' data needing protection, anti-virus companies were left responsible for providing a more immediate answer.

Some, mainly virus researchers, made the decision to do what they do best, and reverse-engineer. Over time, through a combination of help from *Microsoft* and anti-virus research ingenuity, macro virus detection and several different implementations of repair were made available. But the answer to one problem immediately created another!

Methods of detection are unique to each engine and, as long as you do not attempt to implement simultaneously more than one VxD or TSR (on-access) scanner, the products will rarely cross paths directly. Repaired files, however, are heavily scrutinized in an environment where multiple scanners exist, so the likelihood of a ghost positive is at an all-time high. This has especially been true with regard to repairing macro viruses.

Many different methods are available which can be used to 'repair' an infected document. It should thus be no surprise that there are now almost as many different implementations of (macro virus) repairs as there are scanners.

Due in part to the sudden widespread proliferation of Concept and the lack of defined cleaning standards for macro viruses, the number of macro virus reports has gone through the roof. Concept is now the most prevalent virus worldwide. As repairs have yet to be perfected, reports come in many cases in the form of a disagreement between two products.

At the time this article was written, it was common for a product to report and repair, but not necessarily repair correctly – at least according to product Y. The root of the problem may be the method of repair used by product X. While technically correct in that the infectious mechanism is disabled, the repair does not satisfy product Y's standards of cleanliness. One, of course, could just as easily assert that the problem lies within the other product's scanning engine.

Why repair? Why not just replace? The pros and cons of each would cover an entire article and still not provide a solid conclusion – nor would everyone agree on all points.

Interestingly, with macro viruses, the preference of repair over replacement becomes less of a nicety and more of a necessity. For standard file viruses or boot infections, replacement is usually possible. Even if you haven't backed up for some time, the COM and EXE files needing replacement probably haven't changed since the last backup.

The same static attributes rarely apply to *Word* files. They change, and often! Feel free to check the rate of your fastest typist, but you would probably have to institute an 'every ten minute' backup schedule just to keep somewhat current!

Conclusions

From my experiences, I find that people tend to rely on more than just one anti-virus scanner. Learning about the advantages and disadvantages of using more than one product also helps you to recognize possible conflicts. Just remember, the very same 'extra security' also opens the door to their interaction. If interpreted incorrectly, it might lead to a very time-consuming, and confusing, situation.

The increasing complexity of computer viruses requires the almost constant development of new anti-virus technologies: because of this, most scanners require periodic updates. Updates often arrive with special virus alerts and procedures to follow on encountering a virus. Depending on severity, products may even go so far as to include manual removal procedures. If you rely on an anti-virus product to protect your data, such information makes required reading.

For those afflicted, have faith! There is good news. Repairs undergo continual refinement as the structure of *Word* and *Excel* files are better understood. Soon, the same 'basic set of rules' we enjoy for COM and EXE files will evolve from the apparent confusion surrounding macro virus repairs.

Finally: if a problem occurs, please make an attempt to contact the manufacturer. It should be a very rare case where the developer doesn't want to make its product shine above the rest.

[1] in a controlled lab, by experienced virus handlers.

[2] repair performed by means other than restoring information from an inoculation-type database

Shane Coursen is an anti-virus research specialist at *Symantec Corp*; readers may contact him at scoursen@symantec.com.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

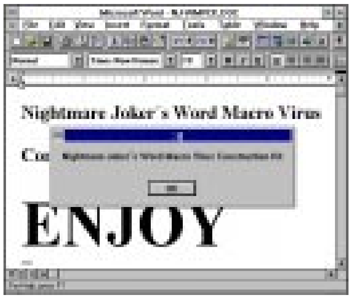
FEATURE 2

Roll-your-own Macro Virus

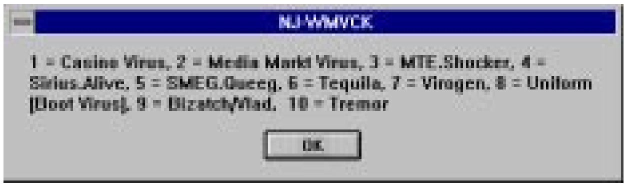
Paul Ducklin
Sophos Plc

Macro viruses are now even easier to write. Two virus writers, Nightmare Joker and Wild Worker, have produced virus construction kits for users of the German and English versions of *Word* respectively.

Nightmare Joker's Word Macro Virus Construction Kit comes as a *Word* template which directly generates first-generation infective viral templates, using a series of dialogues to guide the user through a set of options. One allows selection of German or English, but only German is currently supported.


Nightmare Joker's Word Macro Virus
Co
Nightmare Joker's Word Macro Virus Construction Kit
OK
ENJOY

NJ-WMVCK-generated viruses include two warheads, which are the only variable parts of the viruses the 'product' is able to create. The first warhead randomly appends user-specified text to documents as they are printed; the second attempts to inject and invoke a user-selected non-macro virus.



1 = Casino Virus, 2 = Media Markt Virus, 3 = MTE.Shocker, 4 = Sirius.Alive, 5 = SMEG.Queeg, 6 = Tequila, 7 = Virogen, 8 = Uniform [Boot Virus], 9 = Bizatch/Vlad, 10 = Tremor

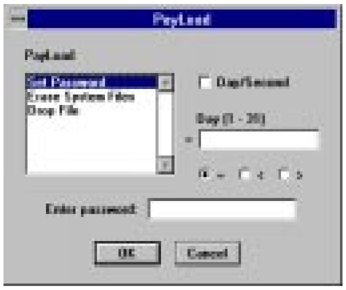
The virus injection is done by means of debug scripts stored in a set of data macros within the NJ-WMVCK template, so the kit also serves as a mini distribution database for those who care to extract the scripts. The variant of Boza included (called 'Bizatch/Vlad' by Nightmare Joker) is, however, the broken version shipped with VLAD's sixth underground 'zine.

All viruses produced with NJ-WMVCK can be detected in a similar way, as they all consist of a similar set of macros: AutoExec, AutoOpen, DateiBeenden, DateiDrucken, DateiNeu, DateiÖffnen, Info and an optional viral dropper macro, named by the user. However, first-generation infected files include unencrypted macros, which are converted to 'execute-only' during replication; thus, scanners using brute force detection will need two search-strings for such viruses.

Wild Worker's Macro Virus Development Kit works slightly differently, and produces viruses in textual form. Although this means it cannot generate infected files directly, it also means that the viruses it produces can more easily be modified before they are incorporated into a first-generation template.

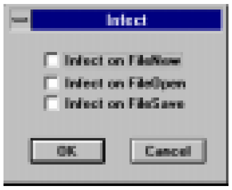
All MVDK viruses consist of an empty AutoExec macro (intended simply to overwrite any security measures, such as DisableAutoMacros, currently initiated through an existing AutoExec) and an AutoOpen macro, which handles initial infection of the global template, as well as further replication of the virus.

MVDK viruses offer a choice of three warheads. Files can be randomly saved with a user-specified password (this is a side-effect borrowed from the Atom virus, which Wild Worker mentions in his documentation), system files can be erased, or a user-specified file can be dropped and invoked in AUTOEXEC.BAT. This user-specified file is supposed to be a debug script, and a utility to convert arbitrary files into debug scripts is included.

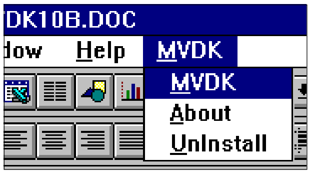


The program logic which triggers an MVDK warhead is also user-selected, and involves either the day of the month or the seconds of the time being compared (using one of the operators 'equal to', 'less than' or 'greater than') to a user-specified value.

Up to three macro hooks can be selected for the New, Open and Save options on *Word's* File menu. Since MVDK viruses also hook AutoOpen, none of these File hooks are actually required for the virus to work.



The MVDK itself is distributed as a *Word* template, which sports a macro button that can be used to install the kit onto the *Word* menu bar. In a fit of professionalism, the author has even included About and Uninstall utilities.



Some observers have noted that macro virus writers seem to be treading the same sort of evolutionary path as DOS virus writers did before them: NJ-WMVCK mirrors the early but simple Virus Construction Set (VCS) that generates DOS file viruses, while MVDK is closer in style to the more sophisticated PS-MPC (Phalcon/Skism Mass Produced Code) virus constructor.

If this is true, we can expect to see other DOS virus 'developments' mirrored in the macro virus world. Watch for enhanced stealth (the stealth functionality in current macro viruses is elementary), polymorphism and cross-application portability.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

FEATURE 3

What You Pay For...

Much is written, in these pages and in other publications, about the risks of downloading a virus from the Internet. Viruses can be picked up both intentionally and inadvertently – this is the threatening side of the free-for-all (well, almost) data transfer the Internet offers.

However, this article will look at the brighter side. Not only does the Internet give viruses the chance to penetrate computer systems, it also offers users the chance to retrieve both software and information to help combat them.

The Old Days

The concepts of ‘freeware’ and ‘shareware’ have been around since the dawn of computing. Indeed, at the very beginning most software was free, and written mainly to further a primitive art. In the PC age, most software on the majority of PCs is purchased, as are the computers themselves. However, the twin worlds of freeware and shareware have managed to survive – whilst most corporates wouldn’t touch shareware with a barge-pole, it is invaluable for genuine enthusiasts and hobbyists.

One of the few problems with software for nothing (or very little) was this: how do you get access to it? The software may be free, but the floppy disk containing it is not, nor is the postage. Certainly, BBSs went some way to alleviating these problems, but long-distance telephone charges which are incurred downloading from BBSs across the world are never welcome, and those were the days of modems with a maximum data transfer rate far less than half of today’s.

Today

The Internet, now in so many offices and homes throughout the world, provides an obvious transfer mechanism for such software. Once connected, the user can download from anywhere from London to Ulan Bator, free of charge.

When it comes to anti-virus software, the Internet has a big advantage. It can take up to several weeks to get a piece of software by post, but you can get it immediately (it takes many minutes, in tedious reality…) from the Internet. If you think your computer has a virus, the last thing you want is to wait days to receive the program to detect and cure it. Now, a user can simply point a browser at a WWW site and download a scanner, and fix the problem there and then.

What’s Out There?

As with everything on the Internet, the problem is not so much ‘is it there?’ but ‘where is it?’. Most major anti-virus companies are on-line, and the rest are working towards it.

In addition, you can sometimes use the major software resources on the Internet – *SimTel* springs to mind as the most well-established of these, but there are many others.

There are many products from smaller companies, or even individuals, which do not have their own Web or FTP site. These are consequently only available from such software resources – more on these later on.

A Word of Warning

This article was begun with clear mental definitions of the terms ‘freeware’, ‘shareware’, and ‘evaluation copy’. However, as soon as the research began, a discrepancy between the way the words are used in the industry, and the definitions perceived at the outset, surfaced.

Take ‘shareware’, for example. This is used to describe software which may be used for a certain length of time before you must either delete it, or pay the registration price to the company, which is similar to the meaning of the phrase ‘evaluation copy’. With shareware, the user may pass the software on to anybody: each user is then subject in turn to the same terms and conditions.

The terms and conditions placed by various manufacturers on their software will be illustrated throughout the article by quoting from the relevant documentation.

Free Stuff

So, what’s out there that will cost you nothing? By far the best known anti-virus product to match this criterion is *Frisk’s F-Prot* – not only is it free, but it’s also very good.

The terms and conditions state that the ‘English-language shareware version of the F-PROT anti-virus program is free of charge for any individual using on his/her personally owned computer, which is not used for a commercial purposes [*sic*]’. Of course, for companies there is a range of prices, based on the number of PCs to be protected. [1]

Almost Free Stuff

Many more companies advertise their products as being available as ‘shareware’, with exact conditions varying only slightly from product to product. For example, *ESaSS’ ThunderBYTE* document states: ‘We invite you to download an evaluation version of our software with no obligation, other than to remove the software at the end of your 30-day trial period if it does not meet your needs or expectations.’ [2]

Similarly, *Stiller Research’s* terms for *Integrity Master* allow the user to evaluate the software for sixty days [3], after which he is expected to pay for the software if he wishes to continue using it.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

The terms and conditions for *McAfee SCAN* are not clearly stated in the accompanying documentation; however, when run, the program displays the message: ‘This version of the software is for Evaluation Purposes Only and may be used for up to 30 days to determine if it meets your requirements… If you choose not to license the software, you need to remove it from your system.’ [4]

Evaluation copies?!

Few products appear in this, the final category used by vendors to describe the versions of their products available for download. However, the term is in current use.

Sophos’ Sweep, when downloaded from its WWW site and executed, informs the user: ‘This software is licensed for evaluation purposes only. It is not licensed for business use or for resale.’ [5] The time allowed for evaluation is not mentioned; the closest specification is the statement: ‘You can download … for a limited period of free evaluation.’ [6]

Also available for evaluation in this way is *S&S’ FindVirus*, which is configured to execute until two months after the release date. As the documentation states: ‘This version of Dr Solomon’s FindVirus is for evaluation purposes only. It is NOT free, shareware, or public domain.’ [7]

It is interesting to note the declaration that the product is not shareware, although the conditions under which it may be used are not vastly different from those which do label themselves shareware. The only material difference is that users are clearly not meant to pass on products calling themselves ‘evaluation copies’, whereas they are positively encouraged to pass on shareware.

Other Notes

So far, this sounds fantastic, doesn’t it? Product after product after product, all free, even if after a while most must be thrown away. However, there is a lot of rubbish out there, and the uninitiated find it difficult to separate the wheat from the chaff.

All sorts of interesting anti-virus products were available for download from the mysterious ether that is the Internet; unfortunately, many had one major problem in common. The text editor with which this article was written dates from early 1994, and the editor with which the author writes email dates from the mid-1970s.

Despite this, both work as well as the day they were completed. However, unlike text editors, anti-virus products have a use-by date: there’s little point in using a scanner from 1991 to protect a PC in 1996. This is the main factor allowing anti-virus vendors to sell you a subscription to their product, rather than just a box with some disks inside.

Products from as long ago as the late 1980s were available (and downloaded) from the anti-viral haunts of the Internet, but no updates to any of these pieces of software could be found. The authors gave up writing it, but somewhere out there, the software lives on. Worthy of note was a Brain detector which was discovered – Brain died out in the wild years ago, and would never survive today. The software worked (it detected and removed Brain from a 360K 5.25-inch diskette…), but is entirely useless in the world of the 1996 computer user.

However, not even the issue of outdated software is easy. Very old (1993) copies of *F-Prot* were also available for download – not only must you beware of obsolete products, but also obsolete *versions* of software. Whilst *F-Prot* in its current incarnation is a perfectly good anti-virus product, the same cannot be said of a three-year-old copy.

Conclusions

That users can download and use anti-virus products at a moment’s notice should they suspect a problem on their machine is undoubtedly a positive development. With the exception of long download times, which are the bane of the Internet (especially for home users connecting via modem), the only real problems involve ensuring that the software is both up to date and up to the job.

A final thought: the software is made available by vendors largely on the honour system – if the conditions require it, it is not only legally a good idea to pay for continued use of the software, but also a matter of courtesy.

The products which have been mentioned in this article are all available to download from the Internet, and can be found at the following sites:

ESaSS TBAV:	http://www.thunderbyte.com/ ftp://ftp.thunderbyte.com/pub/thunder
Frisk F-Prot:	ftp://garbo.uwasa.fi/pc/virus
McAfee Scan:	http://www.mcafee.com/ ftp://ftp.mcafee.com/pub/antivirus
S&S FindVirus:	http://www.drsolomon.com/ ftp://ftp.drsolomon.com/pub/findvirus
Sophos Sweep:	http://www.sophos.com/ ftp://ftp.sophos.com/pub/evaluation
Stiller Integrity Master:	http://www.stiller.com/ ftp://garbo.uwasa.fi/pc/virus

A good general site for links to anti-virus information and copies of updates to many anti-virus products is Joe Hartman’s Anti-Virus Site, HAVS, which is reachable on http://www.psnw.com/~joe/.

Bibliography

[1] ORDER.DOC, part of the *F-Prot v2.23a* distribution, file dated 5 June 1996

[2] WWW document, http://www.thunderbyte.com/eval_req.html

[3] WWW document, http://www.stiller.com/

[4] Output from *McAfee SCAN v2.5.0*

[5] Output from *Sophos Sweep v2.88*

[6] WWW document, http://www.sophos.com/Products/download.html

[7] README.1ST, part of the *FindVirus v7.62* distribution, file dated 17 July 1996

FEATURE 4

Big problems; Simple Solutions

Phil Bancroft

The company where I work makes PCs, and my task is to provide anti-virus tools for roughly 50,000 PCs with users whose experience varies from tyros to PC super-techs.

A phone-based Help Desk is available to help users deal with viruses. The personnel who man these Help Desks vary in expertise, meaning that a central group has had to make virus fighting as simple and easy as possible, and supply the Help Desks with ‘cookbooks’ for the elimination of viruses. With only a small team of real virus experts available for the toughest problems, this solution has worked quite well.

The Process Begins

No one ever intentionally infects their computer system, and no one has ever admitted to booting intentionally from a non-scanned diskette or loading unchecked executables. Early in the anti-virus effort, we determined not to try to assign blame or to punish, both of which are obviously counter-productive. Driving the problem of viruses underground would only delay detection and cause more damage; thus our approach is to cooperate to remove infections, and to try and educate the users in anti-virus technology.

All infections are supposed to be reported to me, so that I can trace trends and best plan our responses. The anti-virus software we employ is so easy to use that the added effort of reporting the infection to me is normally ignored. Possibly our user base is a little ahead of the general PC users; after all, we are a computer company!

I can act, from the reports I get, to make plans which have been successful so far (touch wood!). That the toughest and most difficult infections are forwarded to me for solution gives me some idea how the anti-virus effort is progressing.

Getting the Boot In

The normal infection used to be a boot sector type, and was contracted by the user leaving a diskette, often a data diskette (not system-formatted), in the diskette drive and either rebooting or turning off the PC and rebooting the next day with the diskette still there.

This booting was unintentional. It did not occur to the user that, when they see the ‘Hey, I’m not a system disk’ message, they have really booted the diskette. I attempt to get this information to everyone, but as there is a natural turnover of employees, the task is never done. A semi-solution has been to make diskette-testing easier by creating a *Windows* icon which runs a .BAT program to scan floppies and remove all removable viruses. Reducing the scanning function to selecting an icon with no further interaction has made it much more likely that users will scan new diskettes.

Our newer PCs offer an excellent solution to the boot virus problem, which I expect other manufacturers also offer: the system is set to boot from the hard drive first, instead of always booting from the diskette drive if a diskette is in the drive. The latter default has historical significance, but is no longer desirable for systems with hard drives.

Now that the *Word* macro virus is widespread, a group of users who never exchanged diskettes or copied executable files is at risk. This has affected particularly our management and management support people.

Anti-virus Measures

We have three levels of documents: Policies, Standards, and Guidelines. Policies define the general states required within the company; Standards define the states required for particular environments; and Implementation Guidelines specify how to get to the states required.

Thus a Policy says ‘The Company will maintain information security’, a PC Standard says ‘PCs will be protected against viruses’, and the Implementation Guidelines say ‘You must have an anti-virus TSR running on your PC at all times, and scan all new executables and diskettes before using them’.

The Implementation Guidelines are the document of main interest to the users. They are audited against the Standard, and users employ the Implementation Guidelines to make sure they have attained the required state of security.

Our computing environments vary from key-locked tower PCs in locked offices within badge-access controlled buildings, to laptops used on the seat of an international flight. Realistically, our danger areas include employee-owned PCs which may be used for company work at home, but also by family members. That expands the exposure to virus problems immensely. In response to that threat, our scanners have been licensed for use at home and at work.

Our solutions are to educate as many PC users as we can to avoid viruses, and to assist those who get viruses to eliminate them – eventually the populace will become virus conscious. *Windows 95* systems and *Windows NT* are becoming the operating systems of choice, and at the same time, more vicious viruses are being developed. This helps motivate PC users to keep their systems virus-free.

Phil Bancroft is a member of the *Virus Bulletin* Advisory Board, and an employee of *Digital Equipment Corporation*. He can be contacted via email on bancroft@minotr.ENET.dec.com.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

PRODUCT REVIEW 1

Cheyenne InocuLAN

Martyn Perry

One of *Cheyenne Software's* main lines is its anti-virus software: its products in this field have shown steady improvement in recent *VB* comparative reviews. Its *NetWare* product was last reviewed some time ago [*see VB, December 1994, p.18*] – how does it compare in its latest incarnation?

Presentation and Installation

InocuLAN is licensed on a per-server basis, and allows limited numbers of workstation attachments to the server. The product comes with two manuals, a User Guide and a Supervisor Guide, which cover DOS, *NetWare*, and the *Macintosh*. There are four main set-up diskettes, plus a licence diskette with a serial number, and an additional diskette for the *Macintosh*.

The installation program installs the *InocuLAN Windows* Manager (*Windows* administration program), the *InocuLAN* DOS Manager (DOS administration program), the *InocuLAN* Server NLM, and the Alert NLM, which provides a message-sending facility via various communication methods. *InocuLAN* and Alert use, respectively, the home directories SYS:\INOCULAN and SYS:\ALERT.

The Manager may be installed on the server or on a specific workstation, as required. If it is installed on the server, Critical Disk Area files will be backed up to the server in the CRITICAL.WS subdirectory, in subdirectories for each workstation node address. The Critical Disk Area for a workstation includes: Boot sector, Partition Table, CMOS Ram, and IO.SYS, MSDOS.SYS and COMMAND.COM. This back-up feature could be useful in case of an infection, as it allows rapid restoration of workstation key configuration data without having to hunt for the correct restore diskette.

The final choice to be made on installation is whether or not to add AVUPDATE to the system login script. This allows for the workstations to be updated automatically, with the latest *InocuLAN* files, when they log in.

The program options IMMUNE and EXAMINE can be added to AUTOEXEC.BAT on the workstations. IMMUNE is a TSR that scans files for viruses when they are executed or accessed. There are three versions (large, medium, and small), with a number of options which determine where they load in memory, the types of check they perform, and the actions which can be taken. EXAMINE checks the Critical Disk Area of the workstation for changes.

My one concern about installation is the time it takes for the initial load screen to be displayed – nearly two minutes on the (albeit slow) test PC. This could leave users believing that installation has hung.

Multiple servers can be grouped together into one or more 'security domains', and then controlled either on a per-machine or per-domain basis. There is no default domain, but one can be created and amended using the Domain Manager.

Loading the NLM

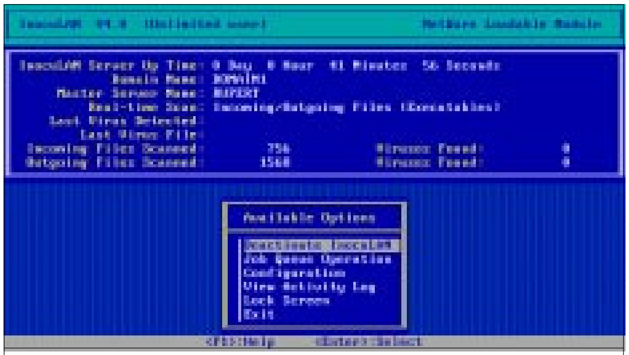
The installation process has no automatic option to include loading the *InocuLAN* NLM in AUTOEXEC.NCF files, possibly due to the number of options that can be specified at load time. These options include automatic synchronization of virus signatures across a domain, and whether to scan only DOS/*Windows* files, or *Macintosh* ones as well.

An option worth special mention is the product's ability to load *InocuLAN* in an inactive state, allowing the supervisor to set up the configuration before the scanner is run. Some products start a scan immediately, with a default set of control values which can only be changed after the initial scan has completed. *InocuLAN* can be driven from the server console to configure and start/stop an immediate scan. Other options allow the administrator to view the activity log, to monitor and control scan jobs, and lock the server screen.

Administration

The system can be administered from workstations running either DOS or *Windows*. The DOS program has the look and feel of a *Novell* menu utility. The main administration screen allows domains and their constituent servers to be viewed.

InocuLAN offers immediate, real-time (on-access), and scheduled scanning. An immediate scan checks the server on demand, using the current immediate settings. Server scanning can be started from the option on the workstation or from the server console. The on-access option allows scanning when a file is copied to or from the server, or when a file is accessed on the server. It is possible to disable this option completely. Scheduled scans provide scanning on a timed basis. Periodic scanning can also be selected, which



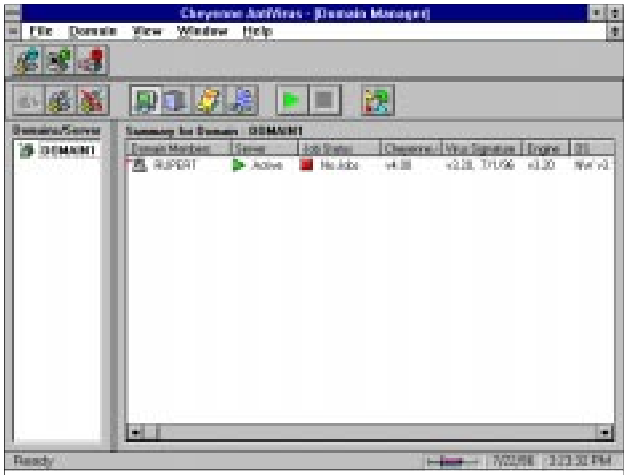
```
InocuLAN v4.0 (Unlimited users)                    NetWare Loadable Module

InocuLAN Server Up Time: 0 Day  0 Hour  41 Minutes  56 Seconds
            Domain Name: DOMAIN1
     Master Server Name: RUPERT
         Real-time Scan: Incoming/Outgoing Files (Executables)
    Last Virus Detected:
        Last Virus File:
 Incoming Files Scanned:      756          Viruses Found:        0
 Outgoing Files Scanned:     1568          Viruses Found:        0

                         Available Options
                        Deactivate InocuLAN
                        Job Queue Operation
                        Configuration
                        View Activity Log
                        Lock Screen
                        Exit
                 <F1>:Help      <Enter>:Select
```

The *InocuLAN* server console is controlled via a standard *NetWare* menu-based interface.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.



Domain Members	Server	Job Status	Cheyenne	Virus Signature	Engine	OS
RUPERT	Active	No Jobs	v4.00	v3.28, 7/1/96	v3.20	Win' v3

The *InocuLAN* domain manager allows complete control of all the servers making up the various *InocuLAN* domains.

occurs at regular intervals (e.g. hourly), from a defined time. No facility exists to start another NLM after a scheduled scan is completed.

Configuration Options

Selections can be made for each mode of operation, including which file extensions to scan. For scheduled scanning, defaults are APP, COM, DRV, EXE, OVL, OVR, PRG and SYS, with extra file extensions added as necessary. The defaults for the real-time scan on the test-set are the same as above, except for the omission of COM files. I assume this is an oversight from the display, as COM files *are* checked. Also, specific files and directories can be excluded from a scan.

Various actions can be taken on virus detection: report only, delete an infected file, rename a file with extension AVB, attempt disinfection, move the file to the quarantine directory (default is SYS:\INOCULAN\VIRUS), and purge, move or rename the file. A virus found in a compressed file cannot be disinfected – the file must first be decompressed.

InocuLAN provides three scanning speeds: a fast scan checks the start and end of each file type selected (COM and EXE files are always fully scanned. A secure scan examines the entire file, and a reviewer scan [*Umm… Ed.*] checks the whole file and also checks for virus-like code within the file.

Alert Management

InocuLAN can issue basic *NetWare* broadcasts on virus detection without outside assistance – they are sent to all users defined in an ASCII file called NT_USER.DAT located in the *InocuLAN* product directory on the server. However, to provide more elaborate and flexible alerting facilities, a separate NLM, Alert, is provided.

This allows the transmission of alerts by alphanumeric or numeric pager, *NetWare* broadcast to users or groups, fax using *Cheyenne's* own *FAXserve*, MHS mail, or Simple Network Management Protocol (SNMP) messages via systems such as *NetWare* Management System (NMS).

Reports and Activity Logs

InocuLAN records activities in an event log. The events to be logged include detection of viruses by WIMMUNE (*Windows* real-time monitor on the workstation) or the real-time monitor (viruses discovered by the workstation or domain scanners are reported in the scanning report), loading *InocuLAN*, and virus signature changes.

Limited control of the data generated is available by placing a limit on the number of messages to be stored, and filtering them such that only messages of certain types are logged. The three types of message are: critical (always selected, indicates a possible virus or network problem), warning (if *InocuLAN* skips a file), and informational (records start/stop of a job, etc).

Other reports available include a list of viruses detected (including the version number of the signature file), and results of local scanner jobs.

Updates and Detection Rates

Updating involves installing the latest signature files in the *InocuLAN* home directory (there is an option to download updates automatically from *Cheyenne's* BBS). If all servers in the domain are loaded with the automatic update option, the update will be propagated to them. Workstation updates can be achieved automatically, provided the system login script was modified at installation time to include AVUPDATE.EXE. To distribute updates between domains, a separate program, SUPDATE.EXE, can be run from a workstation.

The scanner was tested using the usual three test-sets; In the Wild, Standard and Polymorphic. The undetected viruses were identified by using the delete files option and then noting the files left behind.

The tests were conducted using the default scanner file extensions and the virus signatures shipped with the product. The results were generally good: in the In the Wild set, the product only slipped up on the samples of Ph33r, and achieved 98.4%. The Standard set produced a score of 96.1%; the Polymorphic, 88.5% – the product missed all samples of Digital.3547, Girafe:TPE, and 126 samples of Sepultura:MtE.Small.

Real-time Scanning Overhead

To determine the impact of the scanner on the server, 63 EXE files from SYS:\PUBLIC, comprising 4,641,722 bytes, were copied from one server directory to another using *Novell's* NCOPY. NCOPY keeps the data transfer within the server itself and minimizes network effects. The directories used for the source and target were excluded from the virus scan so as to avoid the risk of a file being scanned while waiting to be copied.

The time tests were run ten times for each server setting and an average was taken. The tests were executed in two groups for two conditions. The first group was run with on-access (real-time) scanning selected for incoming and outgoing files, then for incoming files only. The tests were first run with the real-time scan set to FAST, then to SECURE, to gauge the effect of the two modes on performance.

Another test was performed with the Alert NLM loaded in addition to *InocuLAN*, to measure any additional overhead generated by the extra software. All manual (immediate) scans were performed in the default SECURE mode. As the test involves copying EXE files, these would be fully scanned, even in FAST mode.

The four tests were:

- NLM not loaded – establishes the baseline time for copying the files on the server
- NLM unloaded – run after the other tests to check how well the server is returned to its former state
- NLM loaded, using the default setting of on-access scanning for incoming and outgoing files, but with the immediate scanner not running – this tests the impact of on-access protection
- NLM loaded, with on-access scanning for incoming and outgoing files and immediate scan running

The tests were repeated, checking incoming files only. Interestingly, the results consistently showed that the Incoming/Outgoing check was slightly faster than Incoming only. The figures are well within the margin for error, but one would still expect Incoming alone to be quicker. The other result indicates there is an approximately 2% additional overhead when the Alert NLM is loaded. The overhead when *InocuLAN* is unloaded is due to a number of files needed by the NLM (e.g. CLIB) remaining on the server.

Conclusion

InocuLAN is simple to install, and the upgrades are straightforward. The documentation is clear and comprehensive, and includes a good section on recovery procedures. Scan results are good overall, and scanning overhead is relatively low – this can be further reduced by adjusting the CPU usage level. There is also good integration between server and workstation scanners.

The product is aimed at large installations with multiple servers and a large number of workstations to be administered. With this in mind, additional features (not tested in this review) are now available.

For the workstation, these include: remote installation, automatic signature update and auto-disconnect of workstation on infection. Under *Windows 3.x*, *InocuLAN* uses a VxD rather than a TSR to save precious memory space on workstations. On the server side is the Server VirusWall, which can prevent a clean file on a server being overwritten by an infected file of the same name from the workstation.

Overall, the product offers a high degree of configurability with good anti-virus performance.

InocuLAN for NetWare

Detection Results

Test-set[1]	Viruses Detected	Score
In the Wild	307/312	98.4%
Standard	393/409	96.1%
Polymorphic	8853/10000	88.5%

Overhead of On-access Scanning:

Tests show time taken to copy 63 EXE files (4.6MB). Each is performed ten times, and an average taken.

	Time	Overhead
NLM not loaded	25.1	n/a
NLM unloaded	29.8	18.5%
Incoming/Outgoing		
Real-time FAST		
NLM loaded, no manual scan	33.9	35.0%
NLM & Alert loaded; no scan	34.5	37.4%
NLM loaded; manual scan	36.3	44.4%
Real-time SECURE		
NLM loaded, no manual scan	34.0	35.3%
NLM loaded; manual scan	36.0	43.6%
Incoming Only		
Real-time FAST		
NLM loaded, no manual scan	34.1	35.7%
NLM loaded; manual scan	36.8	46.7%
Real-time SECURE		
NLM loaded; no manual scan	34.4	37.2%
NLM loaded; manual scan	36.9	47.0%

Technical Details

Product: *InocuLAN for NetWare v4.0*.

Developer/Vendor: *Cheyenne Software*, 3 Expressway Plaza, Roslyn Heights, New York 11577, USA. Tel +1 516 484 5110, fax +1 516 465 5115.

Distributor UK: *Cheyenne Software*, Stonecroft, 69 Station Road, Redhill, Surrey, RH1 1DL. Tel +44 1737 775500, fax +44 1737 775520.

Price: US$995 per server. A European agreement includes a licence for 25 workstations per server; elsewhere, workstation licences must be purchased separately. Volume discounts and other options are available, and free monthly updates (on the 15th of each month) are included.

Hardware Used: Server – *Compaq Prolinea 590* with 16MB of RAM, 2 GB of hard disk, running under *NetWare 3.12*. Workstation – *Compaq 386/20e* with 4MB of RAM, 207 MB hard disk, running under *MS-DOS 6.22, Windows 3.1*.

[1] **Test-sets:** For a complete listing of all the viruses used in this review, see *Virus Bulletin*, July 1996, p.22. For a complete explanation of each virus, and the nomenclature used, please refer to the list of PC viruses published regularly in *VB*.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

PRODUCT REVIEW 2

Backing a WINner?

Dr Keith Jackson

F/WIN claims to be ‘A heuristic Windows virus scanner, detects and cleans known and unknown Word macro viruses, NE-EXE and PE-EXE viruses (Windows 3.x and 95 viruses)’. Note the emphasis on *Windows*, and on macro viruses. This review did not test the product under *Windows 95* and OS/2. *F/WIN* only claims to be able to detect macro viruses in v6.0 and v7.0 of *Microsoft Word* documents.

Heuristics

F/WIN uses heuristics to decide whether or not a file is infected. The advantages of this approach are that unknown viruses can be detected, and that updates may be required less frequently than for other scanners.

Using a heuristic scanner can have disadvantages. Files can be incorrectly reported as infected (a ‘false positive’) with greater frequency than a conventional scanner – *F/WIN* even lists known false positives in a file called FWIN.TXT! When an infected file is found, a heuristic scanner may ‘know’ the file is infected, but may not know what the infection is. Depending on circumstances, this may or may not be a limitation. On the plus side, it does enable the scanner to pick up viruses about which it does not explicitly know.

Installation

F/WIN was provided as a ZIP file sent by email. This is their normal method of distribution; requesting a floppy disk version incurs extra charges. The key file included can be updated by obtaining a key file from *F/WIN* developers to change from a shareware to a full-featured version.

Installation was merely a matter of unzipping the *F/WIN* archive into a suitable subdirectory, reading the documentation, and using the executable program FWIN.EXE. Batch files can be created to produce individually-tailored scans.

F/WIN, although it places great emphasis on *Windows*, is a DOS program – curious, given its *raison d’être* is detecting *Windows* viruses. The documentation justifies this by stating that in this form, *F/WIN* could be used if crucial *Windows* files were damaged or corrupted. The product is said to have been tested using several OSs (see Technical Details), and copes with the long file-names used by *Windows 95*. Upgrades to subsequent *Windows*-specific versions are promised.

Documentation

The documentation, provided in the form of disk-based TXT files, is thorough, easy to understand, and contains explanations of all possible error messages and an extensive glossary. There is also technical detail on *Windows* viruses, including a description of how they first developed. In reality, two types of documentation are included, one for ordinary users, and for more technical users, interleaved explanations. I like this style.

The documentation is prescient when it states ‘macro viruses which infect documents are fairly new’, then explains that other products, such as *Excel* and *Word Perfect*, have their own built-in macro languages, and are thus vulnerable to being targeted by their own type of macro virus. The discussion in *VB* [*August 1996, p.8*] about an *Excel* macro virus has realized these fears – an *F/WIN* update is now in the pipeline.

Legal

F/WIN comes accompanied by some of the most impressive legal gobbledygook I have ever seen. A prospective user must beware that ‘authorized distributors of *F/WIN Anti-virus* accept no responsibility in the event that *F/WIN* malfunctions or does not function’. The developers state that *F/WIN* is not guaranteed to be error-free, only agree to limited attempts to fix problems, and disclaim warranties ‘including but not limited to implied warranties of merchantability, fitness for a particular purpose and noninfringement of third party rights’.

If the developers take legal action as a consequence of a dispute over supposedly illegal distribution and/or use of *F/WIN*, you are directly liable for their legal costs. This seems quite curious, but often happens in such cases.

I have omitted many other bits of legal jargon, but you probably get the picture. It appears that the norm for American legal documents is becoming more and more restrictive as time goes on.

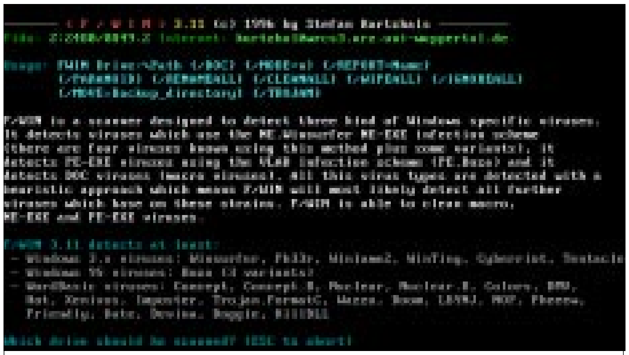
Scanning

F/WIN documentation states that there are currently about thirty macro viruses. The product claims to be able to detect all these, and, using heuristic methods, many future *Windows* viruses. It cannot detect macro viruses in password-encrypted *Microsoft Word* documents: this is unsurprising, but it should be noted that this restriction applies to all systems that detect macro viruses, and may add to their spread.

F/WIN can scan an entire drive, or all files in a selected subdirectories path. A subdirectory scan must be specified as a command-line parameter: if no scanning location is specified, *F/WIN* displays a help screen and waits for a drive letter to be specified before commencing a scan of the named drive.

When *F/WIN* is used under *Windows*, and an invalid drive letter is specified (for instance, no diskette is present in the floppy drive), *F/WIN* produces an error message, then returns to *Windows* so quickly that it is impossible to read the text. This is not very useful.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.



```
──────── [ F / W I N ] 3.11 (c) 1996 by Stefan Kurtzhals ────────
Fido: 2:2480/8849.2 Internet: kurtzhal@wrcs3.urz.uni-wuppertal.de

Usage: FWIN Drive:\Path (/DOC) (/MODE=x) (/REPORT=Name)
       (/PARANOID) (/RENAMEALL) (/CLEANALL) (/WIPEALL) (/IGNOREALL)
       (/MOVE=Backup_directory) (/TROJAN)

F/WIN is a scanner designed to detect three kind of Windows specific viruses.
It detects viruses which use the NE.Winsurfer NE-EXE infection scheme
(there are four viruses known using this method plus some variants), it
detects PE-EXE viruses using the VLAD infection scheme (PE.Boza) and it
detects DOC viruses (macro viruses). All this virus types are detected with a
heuristic approach which means F/WIN will most likely detect all further
viruses which base on these viruses. F/WIN is able to clean macro,
NE-EXE and PE-EXE viruses.

F/WIN 3.11 detects at least:
 - Windows 3.x viruses: Winsurfer, Ph33r, WinlamerZ, WinTiny, Cyberriot, Tentacle
 - Windows 95 viruses: Boza (3 variants)
 - WordBasic viruses: Concept, Concept.B, Nuclear, Nuclear.B, Colors, DMV,
   Hot, Xenixos, Imposter, Trojan.FormatC, Wazzu, Boom, LBYNJ, NOP, Pheeew,
   Friendly, Atom, Divina, Doggie, BIIIDLL

Which drive should be scanned? (ESC to abort)
```

F/WIN is designed to supplement another scanner.

Scanning Speed and Detection

Under DOS, *F/WIN* scanned the hard disk of my test PC in forty seconds. In comparison, *Dr Solomon's AVTK* did the same scan in 38 seconds, and *Sophos Sweep*, in 1 minute 18 seconds. When the same scans were performed from within *Windows* (and with the same scanners), *F/WIN* increased its scan time by three seconds, whilst the other two increased by ten. Given *F/WIN's* emphasis on *Windows*-specific viruses, this is a plus.

After discussion with *VB's* editor, a set of *Windows* and macro viruses was created for this review. It comprised 28 different samples (see the Technical Details section for a description). *F/WIN* detected 26 of the 28 test samples, missing only WinVir1.4 and Twitch [*neither of which can survive in the wild anyway. Ed.*].

When presented with a sample of Wiederoffnen, a *Word 2.0* macro Trojan, the product stated that it was unable to check inside *Word 2.0* files. In spite of its various levels of detection, the scanner always missed the same two test samples, no matter what level was chosen.

On the face of it, *F/WIN's* detection rate is not as good as other, 'ordinary', scanners. For instance, *Dr Solomon's Anti-virus Toolkit* found that all 28 test samples were infected.

Disinfection

F/WIN's documentation states that the product is able to remove viruses it detects from their host files – it transpires that it can only do this with macro viruses, however, and not with *Windows* viruses.

This is a sensible decision on the part of the developers – in the case of *Windows* EXE infectors, it is always better to restore from backups, and as the EXE files on an average system do not change very frequently, these are more likely to be available than for the rapidly-changing DOC files in which *Word* viruses reside. Cleaning these is essential, as up-to-date backups are very unlikely to be available.

F/WIN's developers acknowledge that trying to remove a virus can go wrong: 'If on the outside chance the cleaning process leaves it unreadable…' To their credit, they do insist on creating a backup before cleaning, and the product will not disinfect if it cannot create this backup. However, the shareware version of *F/WIN* does not include file disinfection capabilities – only NORMAL.DOT can be cleaned!

Conclusions

F/WIN's emphasis on *Windows* viruses may not entirely be reflected in actual virus incidents – certainly Concept heads the prevalence table, but boot sector viruses are still a problem in the real world, and conventional parasitic viruses should not be forgotten. To be fair, *F/WIN's* developers acknowledge this, and do not claim that *F/WIN* should be the sole scanner. It is promoted as a supplement to other scanners.

F/WIN is fast at scanning a disk, and has a notable advantage over other scanners when searching for macro viruses. It opens every file on disk to see if it is a *Word* file, and if so checks it for viruses. This makes it run much more slowly than one might expect (a speed comparable to a product checking for 'all' viruses), but means it will catch macro viruses in files which other products will, in default mode, not scan.

It also provides more in-depth information about each virus than most other products [*a task made fairly easy by the current limited numbers of macro viruses. Ed.*]. Even so, I'm at something of a loss as to just *who* should use *F/WIN*. Scanners I used in comparison are as good at detecting *Windows* and macro viruses, so why bother using *F/WIN*?

Its advantage is its ability to detect new macro viruses, which other scanners cannot. [*Indeed, testing against macro viruses which have appeared since the product's release, F/WIN detected them all, something no other product did. Ed.*]

In summary, *F/WIN* will find a place if you are concerned about contracting a new macro virus, or about the ability of your anti-virus product to detect and remove macro viruses. Nonetheless, do not use *F/WIN* until it imposes reasonable legal conditions: the current ones are impossibly onerous.

Technical Details

Product: F/WIN v3.11E, serial number 45585669E8714FA7.

Developer: Stefan Kurtzhals, Durrenberg 42, 42899 Remscheid, Germany. Email: kurtzhal@wrcs3.urz.uni-wuppertal.de.

US Vendor: *Computer Virus Solutions*, Gary Martin, PO Box 30802, Gahanna, OH 43230, USA. Tel +1 614 337 0995, email: fwin_sup@ix.netcom.com, WWW: http://www.gen.com/fwin/.

Availability: A PC operating under *PC DOS*, *MS-DOS*, *Windows 3.x*, *Windows 95* (DOS 7.0), or *OS/2 Warp* (from a DOS window).

Price: Cost per PC: 1 – US$30; 2-9 – US$20; 10-24 – US$15. The price with the number of PCs to be protected: 5000+ PCs cost US$0.50 per unit. Free updates if downloaded from the Internet; US$75 if shipped on diskette (maximum of 10 floppies).

Hardware: A 33MHz 486 PC containing 12 MB of RAM and a 1.2 GB hard disk, under *MS-DOS v5.00* and *Windows v3.1*.

Viruses used for testing purposes:

Macro viruses: Atom, Boom, Colors.B, Concept, Concept.Fr, Date, Divina, DMV, Doggie, Friendly, Guess, Hot, Imposter, LBYNJ, NOP, Nuclear, Nuclear.B, Pheeew, Polite, Wazzu, Trojan.FormatC, Xos. *Windows 3.1* viruses: CyberRiot, Tentacle, Twitch, WinTiny, WinVir14. *Windows 95* virus: Boza.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

ADVISORY BOARD:

Phil Bancroft, Digital Equipment Corporation, USA
Jim Bates, Computer Forensics Ltd, UK
David M. Chess, IBM Research, USA
Phil Crewe, Ziff-Davis, UK
David Ferbrache, Defence Research Agency, UK
Ray Glath, RG Software Inc., USA
Hans Gliss, Datenschutz Berater, West Germany
Igor Grebert, McAfee Associates, USA
Ross M. Greenberg, Software Concepts Design, USA
Alex Haddox, Symantec Corporation, USA
Dr. Harold Joseph Highland, Compulit Microcomputer Security Evaluation Laboratory, USA
Dr. Jan Hruska, Sophos Plc, UK
Dr. Keith Jackson, Walsham Contracts, UK
Owen Keane, Barrister, UK
John Laws, Defence Research Agency, UK
Roger Riordan, Cybec Pty Ltd, Australia
Martin Samociuk, Network Security Management, UK
John Sherwood, Sherwood Associates, UK
Prof. Eugene Spafford, Purdue University, USA
Roger Thompson, ON Technology, USA
Dr. Peter Tippett, NCSA, USA
Joseph Wells, IBM Research, USA
Dr. Steve R. White, IBM Research, USA
Dr. Ken Wong, PA Consulting Group, UK
Ken van Wyk, SAIC (Center for Information Protection), USA

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

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
CompuServe address: 100070,1340
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

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

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

This publication has been registered with the Copyright Clearance Centre Ltd. Consent is given for copying of articles for personal or internal use, or for personal use of specific clients. The consent is given on the condition that the copier pays through the Centre the per-copy fee stated on each page.

END NOTES AND NEWS

S&S International is presenting **Live Virus Workshops** at the *Hilton National* in Milton Keynes, Buckinghamshire, UK on 2/3 September and 7/8 October 1996. The company has also announced a new version of the *AVTK for Macintosh*. The revamped product is said to offer complete detection of *Macintosh* viruses, as well as DOS boot sector and macro viruses. Details are available from the company: Tel +44 1296 318700, fax +44 1296 318777.

***Sophos Plc's* next anti-virus workshops** will be on 25/26 September 1996 at its training suite in Abingdon, UK. The two-day seminar costs £595 + VAT. One single day may be attended at a cost of £325 + VAT (day one: Introduction to Computer Viruses; day two: Advanced Computer Viruses). For further information, contact Julia Line, Tel +44 1235 544028, fax +44 1235 559935, or on the company WWW site; http://www.sophos.com/.

Readers are reminded that the *6th Annual Virus Bulletin Conference and Exhibition* takes place at the *Grand Hotel* in **Brighton, UK, on 19/20 September 1996**. Contact conference coordinator Alie Hothersall, Tel +44 1235 555139, for details.

The *NCSA* is hosting the ***Web, Internet Security and Firewall Conference***, which will be held in San José, California from 30 September to 1 October. Details on the event can be obtained from the *NCSA*; Tel +1 717 258 1816, fax +1 717 243 8642, or email fwcon96west@ncsa.com. Information is also available from their WWW site: http://www.ncsa.com/fw96west.html.

IBM Corp has launched *IBM AntiVirus v2.5*, which has **the ability to scan Internet documents and to check for macro viruses**, and now also incorporates a VxD. In linked work, researchers at the *IBM T J Watson Research Center* are studying viruses using an epidemiological approach, which they hope will help them create 'an immune system for cyberspace which will protect the world's computers from viruses', including not-yet-extant infectors. For information on this and other developments, contact Andrea Minoff at *IBM*; Tel +1 914 759 4713, or email minoff@vnet.com.

The *Computer Security Institute* (*CSI*) *23rd Annual Computer Security Conference* is to be held from 11–13 November in Chicago, Illinois, USA. The event will feature a program of over 120 sessions, including presentations on **Internet security, viruses, email**, etc. It will also include an exhibition of computer security products, for which free passes are available from the *CSI*. Those wishing to receive further information on attending can contact contact Patrice Rapalus at the *CSI*; Tel +1 415 905 2310; email prapalus@mfi.com.

Reflex Magnetics has another **Live Virus Experience** scheduled for 9/10 October 1996. Further information is available from Rae Sutton: Tel +44 171 372 6666, fax +44 171 372 2507.

International Data Security is holding a series of ***Security and Network Management Seminars*** at venues throughout the UK, on 10 and 23 September, 8 and 22 October, 5 and 19 November, and 3 December. The one-day seminars focus on BS7799 and its implementation. Information from Miralle Bonne, Tel +44 171 209 2222.

At the Old Bailey in London, a man has been cleared of blackmailing *Sun Alliance*: it had been alleged that Keith Lamb, from Bournemouth, **threatened to infect the company's computers with 'computer bombs and polymorphic codes'** if a claim he had made (which had been rejected) was not paid. Lamb claimed that it had been an 'April Fool's prank' designed simply to panic *Sun Alliance* technical staff.

Compsec 96, the 13th world conference on computer security, audit, and control, is being held in London from 23–25 October 1996. Contact Alex Verhoeven on Tel +44 1865 843654 for details.

VIRUS BULLETIN **©1996 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /96/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.