ISSN 0956-9979 JULY 1994

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

Editor: **Richard Ford**

Technical Editor: **Fridrik Skulason**

Consulting Editor: **Edward Wilding**,
Network Security Management, UK

CONTENTS

EDITORIAL
Fire! 2

VIRUS PREVALENCE TABLE 3

NEWS
Junkie Mail 3
VB '94 Conference on Track 3
Chill Out! 3

IBM PC VIRUSES (UPDATE) 4

FEATURE
The Ludwig Collection 6

VIRUS ANALYSES
1. Stealth.B: Invisible Fire 8
2. Argyle: Viruses and the i386 10

COMPARATIVE REVIEW
The Review, Reviewed 12
VB Scanner Review: July '94 14

PRODUCT REVIEW
Norton on *NetWare* 20

REVIEW
Virus: Prevention, Detection, Recovery 23

END NOTES & NEWS 24

IN THIS ISSUE:

- **Scanner Update.** The latest comparative review includes 27 anti-virus products. Find out which products have kept up and which have fallen behind on pp.14-19.
- **Viruses for Sale.** Mark Ludwig has released a CD-ROM containing thousands of viruses, including many new ones. What are the contents of *The Collection*, and what are the implications for the industry?
- **Information Junkie.** According to *Reflex Inc*, the latest virus to hit the streets is the 'new and dangerous' Junkie virus. An evaluation of the genuine threat to systems is given on p.3.

VIRUS BULLETIN **©1994 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)235 555139. /94/$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

Fire!

Last month's article on the Pathogen virus, which criticised the way it had been publicised, elicited a squeal of discomfort from Dr Alan Solomon, who believes that the *S&S International* press release was far from 'wholly unnecessary' (see *VB*, June 1994, p.3). Claiming that his actions were in the users' interest, he insists that the alert was legitimate. However, whether or not this is the case, it seems likely that the publicity generated did more harm than good.

When a new virus is first discovered, issuing a virus alert to product users is an attractive course of action, as it allows a new and unquantified threat to be dealt with in the quickest possible manner. If a user pays for protection from a company, he has some justification in expecting interim product updates if and when a new threat is discovered.

Whether or not one should inform the popular press about a new virus is a much more difficult decision. At what level of threat does a virus alert become more than simple market manipulation? One sample found in the wild? Ten? One hundred? When a new virus is discovered, there is rarely any data available on how widespread it is. This was the case with Pathogen - the outbreak could have been an isolated incident, or it could have been the tip of an iceberg. With little information on prevalence, press releases were put out by three different UK companies.

“ The new and dangerous Blancmange virus, undetectable by all known anti-virus software, has been discovered running wild on UK machines ”

Once a decision to send out a press release has been made, things become still more difficult, as control will pass from the technical department to the marketing or PR department. An honest description of a new virus out in the wild might read as follows: 'We at *Acme Virus Detection Inc.* have discovered a new virus named Blancmange at approximately 15 sites. The virus is not detected by the current version of our anti-virus software. Although there is no cause for alarm, users are urged to add the following driver file…' Such a report is factual, causes no panic and stands little or no chance of being published. However, thirty minutes after entering the PR office, the report might appear in a slightly different form: 'The new and dangerous Blancmange virus, undetectable by all known anti-virus software, has been discovered running wild on UK machines by *Acme Virus Detection Inc.* Currently, no other product except *Acme's* own *AcmeScan* can…'

This puts an anti-virus software developer in a difficult position. Assuming he genuinely wants to alert the public to a potential threat, he will only get coverage by dramatising the situation. A calm, measured response is not news. Therefore, the 'spiced up' release is distributed, and the marketing machine grinds away, the original motive for the warning long forgotten.

There is no easy way out of this loop. *S&S International* is not the only company ever to obtain press coverage by releasing news of a new virus, and is certainly not the worst offender - it is in the company of several other major players in the industry. *Secure Computing* described Pathogen as 'spreading widely', while *Reflex Inc* (the USA vendors of *Disknet*) claims that the latest threat to security, the Junkie virus, is a 'new generation' and of 'special concern'. Such claims are counter-productive at best, and at worst leave the companies open to accusations of scare-mongering and market manipulation. When warning of a new virus, vendors should stick closely to the facts.

One possible solution to this problem is that all companies take a measured response to new viruses. When the threat is believed to be small, nothing need be done until the next product update. When the threat is unknown, a factual warning should be issued to all product users. Finally, only when the threat is *known* to be large, with a significant number of sites infected, should a press campaign be launched. Until the extent of a problem is known, there is no justification for spreading alarm.

The current situation is rather like that of the company fire officer who holds too many fire drills: the people in the building will eventually assume that the cry of 'Fire!' does not require immediate action, merely a slow wander from the premises. The virus industry is putting itself in the same position by sending out misleading stories concerning the 'latest and greatest' virus which is in the wild. How hard will it be to convince people when the threat is much larger?

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

Junkie Mail

The latest virus to hit the headlines is Junkie, which, according to a press release sent out by *Reflex Inc*, the American reseller of *Disknet*, is 'of special concern'. The press release (entitled ' "Junkie" - New Generation of Viruses Discovered: Nearly Undetectable, Dangerously Infectious') details how the virus was discovered in the wild, in Ann Arbor, Michigan, USA. However, *Reflex* has had no other reports of the virus in the wild since.

Analysis of the Junkie virus reveals that it is mildly polymorphic, and capable of infecting both COM files and the Master Boot Sector of fixed disks. The virus contains no trigger routine, and is described by Fridrik Skulason, author of *F-Prot*, as 'unremarkable'. This description is entirely at odds with the semi-hysterical comments made by *Reflex.*

Commenting on the press release, Mr Frank Horowitz of *Reflex Inc* said, 'It was never our intention to cause panic. We didn't mean to imply that Junkie would bring down every computer in the world, but to alert the user to the dangers of the new generation of viruses. If we succeed in getting this message across to just one user, that's great!' He went on further to say that such a press release could just as easily have been written about Pathogen or Chill, and was meant to be a warning to both vendors and users. Horowitz stated that he had tried to point out that, despite the fact that this virus was 'no Michelangelo', the computer community had been lulled into a false sense of security ■

VB '94 Conference on Track

The fourth annual *VB Conference* will be held in Jersey on 8/9 September. Internationally recognised experts will speak on topics ranging from 'Viruses in the Wild' (Joe Wells), through 'The Computer Underground' (Dr Alan Solomon), to '*NetWare* Security' (Stephen Cobb). Delegates from every corner of the globe have registered: more than 25 countries are now represented, including the first-ever delegates from Hong Kong and Japan.

Linked to the event this year is an exhibition by anti-virus product developers. Particular interest has been shown in the fact that manufacturers will be able, for the first time, to target the *VB* readership directly, and to promote their products at an international venue.

Conference costs are £595.00 (with a £50.00 discount to *VB* subscribers), and, once again, *Expotel International Travel Group* has been appointed to co-ordinate travel arrangements and accomodation for delegates. Conference proceedings are available at £50.00 per copy from mid-September, for those unable to attend (apply to *VB* offices).

For further information, please contact Petra Duffield.
Tel. +44 (0)235 531889 Fax +44 (0)235 559935 ■

Virus Prevalence Table - May 1994

Virus	Incidents	(%) Reports
Form	21	40.4%
Stoned	7	13.5%
JackRipper	3	5.8%
Barrotes	2	3.9%
Cascade	2	3.9%
Exebug.4	2	3.9%
NoInt	2	3.9%
Spanish_Trojan	2	3.9%
V-Sign	2	3.9%
Anti-CMOS	1	1.9%
Dinamo	1	1.9%
DIR_II	1	1.9%
Form.B	1	1.9%
Michelangelo	1	1.9%
Pathogen	1	1.9%
PS-Dropper	1	1.9%
Tequila	1	1.9%
YMP	1	1.9%
Total	52	100.0%

Chill Out!

A new virus has been found in nine files stored in the PBS Forum on *ZiffNet*. According to an Email message sent out by Katherine Prouty, *ZiffNet* Forums Manager, the virus, known as Chill, was only on the forum between the dates of 3 June and 14 June.

The Email goes on to explain that the virus did not originate in these files; versions of the programs downloaded before 3 June are absolutely fine. When asked how it was that the files were uploaded uninfected, and later infected by the virus, Prouty was unwilling to comment, saying that it would be 'a few weeks' before a complete story could be put together.

The Chill virus is a simple memory-resident COM-file infector, which contains code intended to reformat part of the hard drive. It is 544 bytes in length, and is encrypted with a simple XOR algorithm. Updates of both *F-Prot* and *Norton Anti-Virus* which can detect the Chill virus are available from *ZiffNet*.

There is little cause for concern except for those worried that they may have downloaded an infected file, although full record of downloads have been kept, and all recipients contacted. Further information is available from Prouty, whose Email address is 72241,1511@compuserve.com ■

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 20 June 1994. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

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

Burger.560.AS

CN: Another minor variant of this silly overwriting virus. Detected with the Burger pattern.

Freddy 2.1

CER: This is a variably-sized polymorphic virus, which is widespread in South America, and Brazil in particular. No search pattern is possible.

Grog

CN: The Italian Grog family is not a ‘real’ family, but a collection of different viruses written by the same person(s). All contain the word ‘Grog’, and many have text strings in Italian. Some are destructive, others are not. They may infect COM or EXE files, and have varied internal structure. Many seem based on older viruses, but are heavily modified, e.g. using different registers for indexing. Current Grog viruses include Aver_Torto (CN, 647 bytes), Bruchetto (CN, 474 bytes, overwriting), Delirious (CN, 304 bytes, overwriting), Hop (480 bytes, overwrites COM files), Il_Mostro (CN, 660 bytes, detected with the Darth_Vader pattern, but a different pattern is included below) Joe_Anthro (CN, 589 bytes), Metafora (replaces COM files, renames original to EXE), Noncemale (CN, 796 bytes), Ovile (CER, 1417 bytes), Public_Enemy (CN, 800 bytes), Razor (CN, 801 bytes), Sempre (CEN, 373 bytes, overwriting), Trumpery (CN, 202 bytes, overwriting), Villino (CN, 547 bytes) and Wild_Cards (CN, 798 bytes).

```
Grog.Aver_Torto     B802 3DBA F6D6 CD21 8BD8 B43F B903 00BA FD00 CD21 803E FD00
Grog.Bruchetto      B802 3DCD 2193 33ED 33C9 33D2 B802 42CD 2183 FA00 7403 E9B6
Grog.Delirious      B802 3DCD 2172 3093 B904 00BA 4B01 03D5 8BF2 B43F CD21 ADAD
Grog.Hop            B800 3DBA 89EA CD21 93BA 60E6 E8E9 0080 3E60 E6E8 74BF BE63
Grog.Il_Mostro      B802 3DCD 218B D8B9 0300 BAFD 00B4 3FCD 2180 3EFD 00E9 7401
Grog.Joe_Anthro     B800 3D8D 966E 03CD 2193 53B8 2012 CD2F B816 1226 8A1D CD2F
Grog.Metafora       B800 3D61 720D 9362 6360 B960 E9BA 0001 B43F C333 C050 60C3
Grog.Noncemale      B802 3DCD 2172 9393 B43F 8DBC 0601 8BD7 B904 00CD 2172 5380
Grog.Ovile          80FC 3D74 1280 FC4B 740D 3D00 6C75 0580 FB00 7403 E9B2 0006
Grog.Public_Enemy   B800 3D8D 9641 04CD 2193 53B8 2012 CD2F EB12 4707 7207 6F07
Grog.Razor          B800 3D8D 9642 04CD 2193 53B8 2012 CD2F EB12 4707 7207 6F07
Grog.Sempre         803F E974 0F80 3FE8 740A 59E2 D083 06FE 0003 EBA7 8BF3 BF76
Grog.Trumpery       B800 3DCD 2172 3F93 53B8 2012 CD2F B816 1226 8A1D CD2F 26C6
Grog.Villino        B800 3DCD 218B D8B9 0300 8D95 4502 B43F CD21 B802 4299 5259
Grog.Wild_Cards     B800 3D8D 963F 04CD 2193 53B8 2012 CD2F EB12 4707 7207 6F07
```

Gusano

EN: This virus is also known as Buen_Dia. Both names derive from a text in the virus: ‘BUEN DIA!!! Yo soy un GUSANO’.

```
Gusano              B802 3D8D 16FF 01CD 218B D8E8 C200 7403 E82A 00B4 3ECD 21B4
```

Helloween.1063

CER: A 1063-byte variant. Awaiting analysis.

```
Helloween.1063      B43F EB02 B43E E815 0072 022B C1C3 33C9 33D2 B802 41EB 0733
```

Hiperion.249

CR: Awaiting analysis.

```
Hiperion            9C50 80FC 4B75 1153 5156 1E52 5533 EDE8 0D00 5D5A 1F5E 595B
```

HLLO.Orion

EN: A simple overwriting virus which does not seem to work properly. No search pattern will be given, due to the high risk of false positives.

HOT

CN: This 130-byte-long virus contains the text ‘This is hot!’, but is otherwise unremarkable. It overwrites any file with a name matching *.C*, that is COM files, C source files and others. Like other overwriting viruses, it has practically no chance of spreading.

```
Hot                 B842 3DCD 2193 B420 D0E4 B182 BA00 01CD 21C3 B42C CD21 8ACA
```

Icelandic.655

ER: The first new variant of this family to appear in a long time.

```
Icelandic.655       2EC6 0686 020A 5053 5152 561E 8BDA 4380 3F2E 740D 803F 0075
```

Jerusalem.1808.Standard.AO **CER:** A minor variant, detected with the Jerusalem-US pattern. The first few bytes of infected COM files seem to be corrupted.

Jerusalem.Sunday.Satan **CER:** A minor variant, with little but text strings changed. Detected with the Sunday pattern.

Jerusalem.Tarapa.B **CER:** This 2064-byte variant is detected with the Jeru-1735 pattern.

Junkie **MCR:** A 1027-byte encrypted multi-partite virus which is in the wild. It contains the text 'Dr White - Sweden 1994', which might indicate that it was written in Sweden. Junkie received some media attention recently, mostly undeserved - it is not a technically remarkable virus.

```
Junkie (File)       B9F4 0126 8134 ???? 4646 E2F7
Junkie (MBS)        33FF BE00 7CFA 8BE6 8ED7 FB8E C7B8 0202 BB00 7EB9 0400 BA80
```

Khizhnjak.642 **CN:** A 642-byte virus of Russian origin.

```
Khizhnjak.642       B43D B002 CD21 1F73 03E9 9000 EB14 901E 2EA1 2C00 8ED8 BA08
```

Natas **MCER:** A 4744-byte multi-partite virus written by the author of Satanbug. It is polymorphic, with no search string possible. Currently reported as a significant problem in Mexico.

PS-MPC **CER, EN:** This month's encrypted variants are 606.D (CER), ARCV-1.731 (CER), Powermen.718 (CER) and Tim.500 (EN). The non-encrypted variants are 212 and Tim.405 (EN).

Satan **CN:** Two variants, 602 and 612 bytes long, related to the virus reported as 'Liquid'.

```
Satan.602           8BD5 81C2 0402 CD21 B900 00B8 0242 8BD1 CD21 2D03 008B F581
Satan.612           8BD5 81C2 2102 CD21 B802 42B9 0000 8BD1 CD21 2D03 008B F581
```

Slash **ER:** A 457-byte virus. Awaiting analysis.

```
Slash               3D00 4B74 133D 013D 7405 80FC 3D74 099C 2EFF 1E05 01CA 0200
```

Slub **CR:** A 1024-byte virus of East European (probably Polish) origin. Not fully analysed, but contains the strings 'c:\autoexec.bat' and 'c:\slubdestr.n23'.

```
Slub                F8F9 5053 5152 061E 80FC 4C74 2E80 FC4B 7429 EB18 90B4 2C9C
```

Sluknov **CR:** This 871-byte long virus uses encryption, with variable instructions, making the extraction of a simple search pattern impossible.

SMEG **CER:** Two variants, Pathogen and Queeg, are known. Pathogen has already been described - Queeg is closely related to it, using a slightly more advanced polymorphic engine.

Smoka **ER:** A 1024-byte virus containing a long encrypted message in Polish.

```
Smoka               B959 01BB A600 03D9 2E8A 0751 B104 D2C8 2E88 0759 E2ED 0E1F
```

Sofia_Terminator **CR:** A couple of closely related Bulgarian viruses, 839 and 887 bytes long, containing the text 'Sofia 1993 by TERMINATOR'.

```
Sofia_Term          3D00 4B74 1A80 FC3D 7412 80FC 4174 0D80 FC56 7408 80FC 4374
```

Split **CN:** A 250-byte virus containing the string 'SPLIT'. Awaiting analysis.

```
Split               B43D B002 8D96 2102 CD21 8BD8 B43F B904 008D BEDF 018B D7CD
```

Stimp **CN:** Yet another Polish virus. This one is encrypted, 248 bytes long and contains the string 'STIMP-VIRUS made in Poland'.

```
Stimp               8B16 F601 BB05 01B9 5800 9031 1790 83C3 0290 E2F6 C3
```

Suriv_2.I **ER:** A very minor variant, detected with the Suriv_2.01 pattern.

Swedish_Boys.Headache.441 **CN:** Closely related to the 457-byte virus originally named 'Headache'. Contains the text 'Severe Head-Ache Virus V2.oo)+- Created by The Vile One & MaZ Copyright (c)1992 The BetaBoys Development Corp. -Sweden 04/19/92-'.

```
Headache.441        BB01 018A 27BB 0201 8A07 86C4 8BF0 B41A 8D94 B802 CD21 33C9
```

Sze **CN:** Two closely-related viruses, 314 and 351 bytes long. The viruses are of East European, possibly Hungarian, origin.

```
Sze                 B802 3DBA 9E00 CD21 8BD8 B002 E8E8 FFA3 0300 8BCA 8BD0 83EA
```

Vienna.Violator.707 **CN:** Another Vienna variant, not significantly different from those already known. Detected with the Violator pattern.

Yankee_Doodle **CER:** New members of this old family still appear. The Login.2967 variant is detected by the pattern published for a virus originally named CZ2989, which should be named Yankee_Doodle.Login.2989. The TP.44.D variant is detected with the Yankee pattern. There are also two new encrypted variants, 2189 and Warlock, which require new patterns. The Warlock variant contains the text 'Revenge of WARLOCK!'.

```
Yankee_Doodle.2189  44A6 FE44 A78B FEB9 5007 2E8A BC20 0802 FEE8 B600 071F 58C3
Warlock             5B1E 068B F381 C621 008B FE0E 1F0E 0753 B97B 038B 5F01 FCAD
```

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

The Ludwig Collection

For several years, the anti-virus software industry has anticipated that somebody might turn the distribution of computer viruses into a money-making enterprise. The first glimmer of such a trend began as early as 1987, with the launch of Ralph Burger’s book *Computer Viruses: A High Tech Disease*, which featured live virus code. Soon after, others began to cash in, and there are now several different books in a similar vein.

The next step in the process was the launch of Mark Ludwig’s magazine *Computer Virus Developments Quarterly*. Claiming to be a publication aimed at the virus-aware MIS manager, the magazine included material which would help a reasonably inexperienced programmer develop his own virus code. *American Eagle Inc* also offers accompanying disks to the magazine, which include complete commented source code for viruses such as Jerusalem.

The most recent step in this chain of events is the publication of Ludwig’s $100 CD-ROM, blandly titled *The Collection, Outlaws from America’s Wild West*. Although this is not the first time virus code has been available to the general public (*VB*, February 1993, p.2), the disk represents the largest and most complete collection of virus information ever made available to a wide audience.

Weasel Words

One of the first things which one notices about the CD-ROM is that the word virus is mentioned nowhere on the black and white cover or its flipside. Each cover is serial-numbered, and contains the following message:

```
NOTICE

Retain this card! We anticipate updates to
this CD-ROM, and this card is your proof of
ownership. To obtain updates at a special
price, you will be asked to return this card
with your order.
```

Clearly, Ludwig has plans to continue his virus service, although the regularity or cost is not outlined anywhere. The only other point of note about the cover is the following short disclaimer: ‘The software on this CD-ROM is provided as-is without warranty or liability of any kind. The user assumes full responsibility for anything that happens as a result of executing any programs on this disk.’

Contents

The disk’s contents are organised into a number of subdirectories off the root directory. The only file in the root directory is named READ_OR_.DIE, and contains a lengthy disclaimer and a description of each of the principal subdirectories. Ludwig has grouped the files on the CD into fourteen different subjects, as follows:

- **ALIFE** Programs and documentation on non-viral artificial life
- **ANTI-VIR** Anti-virus programs and utilities
- **HOSTS** Examples of sacrificial goat programs
- **LIVE-VIR** The main body of the CD; the live viruses directory, subdivided by virus family name
- **NEWSLETR** Newsletters and other literature
- **OTHER-OS** Viruses for non-DOS systems
- **NEW_VIR** ‘New’ viruses
- **SOURCE** Source code listings for viruses
- **TESTBED** A test-set of polymorphic viruses against which to test virus scanners
- **TOOLS** A number of virus handling tools
- **TROJANS** Trojan horse programs
- **V-SIMUL** Virus simulation programs
- **VIR-INFO** More text and virus information
- **VIRTOOLS** Virus writing toolkits etc.

The most interesting of these sections will now be considered in turn.

Live Viruses

The virus collection itself is very complete, and has 573 subdirectories leading off it, each labelled with the name of a particular virus family. Within each of these directories are a number of individual samples, attached to an *American Eagle* goat file. Each goat file contains a copyright message, and displays the text ‘Host #1 - You have just released a virus’ when executed.

It is worth noting that every virus in this part of the collection has been transferred to a new host file - a long and time-consuming process, which when done properly prevents files which are incapable of replicating being included in the virus collection. However, at least eight ‘junk’ files have somehow slipped through. The cataloguing here follows the usage of the scanner *F-Prot*, and is sufficiently accurate that it classifies subvariants of each virus (e.g. Cascade.1701.B).

Each infected file has a purely numerical name. The convention used is as follows: a three digit name (e.g. 002.COM) indicates that the file is a replicating sample of the virus. A four digit name (e.g. 0003.COM) indicates that the file appears to be infected, but does not seem to replicate.

Ludwig takes an unusual amount of care with the boot sector viruses included in *The Collection*. These are supplied in one of three forms: a virus dropper program, an image file of an infected disk, and a binary copy of an infected boot sector. Ludwig even goes to pains to make clear that the final group will not replicate or function, and that the virus code is often incomplete.

Although this live virus section will be of immediate interest to anti-virus software developers, the section NEW_VIR is the one which will require the most analysis. According to the documentation supplied on the CD, these viruses 'are largely untested, and there is no telling what executing one will do'. To make things even more confusing, the samples in the directory do not conform to Ludwig's normal naming convention. Each file has a text name, and it is relatively easy to run one of the samples accidentally.

Due to the large number of viruses contained on the CD-ROM, it has not yet been possible to ascertain exactly how many of the files in this directory are new viruses or already-known samples. However, *Dr Solomon's AVTK* (v6.64) found 215 infected files in the NEW_VIR directory out of a possible 469. Similarly, *Sophos' Sweep* (v2.62) detected 232 in quick mode, and 273 in full mode. An automated junk search found 86 of the missed files not to be viruses.

Shareware and Hosts

The Collection is not entirely dedicated to virus writing information - also included are some of the standard tools for virus detection and removal. The directory ANTI_VIR contains a large number of shareware and freeware virus detection utilities. Most notably, reasonably up-to-date versions of *F-Prot* (v2.11), *McAfee Scan* (v113), and *ThunderBYTE Anti-Virus* (v6.12) are included.

The HOSTS directory is another useful source of information for the would-be virus researcher, containing copies of Ludwig's standard host files, and executable code which can be used for generating large numbers of files ready for virus infection.

The Best of the Rest

The rest of the CD covers many other related topics. A large collection of computer underground publications is contained in the directory NEWSLETR, covering publications ranging from *40Hex* (issues 1 to 11) to the *Virus-L* archives. A complete set of virus creation toolkits is included on the CD. Included in this group are VCL, G2, etc. - all the well-known construction toolkits are represented. Finally, there is the obligatory collection of polymorphic virus engines such as TPE. These appear on other areas of the disk too: the TESTBED directory contains 1000 copies each of MtE and TPE-infected files, apparently so that the user can test his anti-virus product against them.

The sections described in the preceding sections make up the bulk of the information on the CD. However, with over 157MB of data in *The Collection*, a cursory analysis will only scratch the surface.

Conclusion

It is important to draw the reader's attention to a number of points. Firstly, those who had hoped that Ludwig's long-awaited collection would be the usual computer underground collection of crippled executables and renamed text files are in for a disappointment. Nothing could be further from the truth. A vast amount of work has been carried out transferring viruses to standard host files, forming a relatively well-ordered and (with the exception of the NEW_VIR directory) 'de-junked' collection.

Secondly, the collection is well organised and classified according to the *F-Prot* naming convention (close to, but not identical to that of *CARO*), making it quick and easy to find and extract a particular virus. Had *The Collection* been produced by a member of the more generally accepted anti-virus community and given a controlled distribution, it could well have been greeted as a gift from the gods, tying up a number of naming issues, and becoming a standard collection used within the community.

Unfortunately, Ludwig's collection does pose many problems for the industry. The greatest of these is that, with large amounts of virus source code available, it is likely that an upturn in the number of variants of particular viruses will occur - even something as trivial as using a different assembler could lead to a new virus.

How the industry and the computing community chooses to deal with this CD-ROM is a subject which needs discussion. If any industry action is to be taken, it needs to be done soon: at the end of the READ_OR_.DIE file, Ludwig leaves the reader in no doubt as to how he is prepared to obtain samples:

```
********************************************************************************
*                                                                              *
*                    WANTED: DEAD OR ALIVE ---> Your viruses!                  *
*                                                                              *
*   Anti-virus researchers, virus writers, hackers, system administrators,     *
*   we want your viruses and we actively trade and buy new viruses. Please     *
*   contact us for details. PLEASE USE THE TOOLS IN THE \TOOLS DIRECTORY       *
*   FOR SAVING BOOT SECTOR VIRUSES. FAILURE TO DO SO COULD DAMAGE THEM!        *
*                                                                              *
```

[*Closing remarks deleted. Ed.*]

VIRUS ANALYSIS 1

Stealth.B: Invisible Fire

Joe Wells
Symantec

Very visible fires were consuming much of southern California. The nearest and largest of these was raging on the hills behind my home, so I was up late monitoring its progress. To pass the time, I disassembled a virus we had recently received from several sites in the southeast United States; most were from the Miami, Florida area.

The virus, named Stealth.B, infects the DOS boot sector on floppies and the Master Boot Sector (MBS) of the first physical hard disk (drive 80h, usually the C: drive). It is also, as its name implies, a stealth virus.

Evolution of a Virus

The virus was reportedly based on the Stealth boot virus source code contained in *The Little Black Book of Computer Viruses*, by Mark Ludwig. Therefore I had tracked down a copy of the book to borrow earlier in the day, so I could compare it to the samples we had received.

A quick difference analysis revealed that, apart from some instruction swaps and slight offset differences, the infected MBSs of the two viruses were effectively the same, except for one byte. A comparison of the complete viruses revealed a difference of less than 5% (172 bytes different in 3584 bytes of code and data).

The Genealogy of Computer Viruses

Trying to ascertain which virus derives from which is not an exact science. If two viruses are very similar, it is of some academic interest to identify how the variants relate to each other. Simply because sample *x* was found before sample *y* does not mean that *x* was written before *y*. They may share a common ancestor, or one may be a simple mutation of the other.

Deriving a virus' family tree involves a detailed analysis of the instructions which make up the virus code. Many i8086 instructions have two different binary forms which are functionally identical - changes to these instructions, for example, would indicate the use of a different assembler, or that the older sample had been disassembled to a source code form, and reassembled. In the case of Stealth.B, even a detailed analysis of the code does not make it clear how it is related to Stealth.A, as the differences in the virus code are small. Only the virus author knows for sure, and with Stealth.B common in the USA, he is not telling.

So Ludwig's Stealth and Stealth.B are not completely identical. There are, however, many striking similarities between the two. Both infect 360k and 1.2M floppies by formatting an extra track and placing five sectors of virus code followed by the original boot sector. On 720k and 1.44M floppies, however, both use the last cluster, head 1, to store the code and boot sector, and mark these sectors as bad to protect them.

Additionally, both viruses infect the Master Boot Sector and use track 0, head 0, sectors 2-7 on the hard drive to store the additional sectors. Finally, they both hide infected sectors by returning either the original sector (if the floppy boot or MBR sector is being read), or a null buffer (if a storage sector is being read).

One would assume that the virus in the book uses two methods of infecting floppies to demonstrate the two methodologies. In the wild, however, the tactic serves no purpose, and is probably just an evolutionary remnant inherited from its progenitor.

Operation

Stealth.B infects a system in the usual manner: someone unintentionally boots from an infected floppy. Like most boot viruses, Stealth.B immediately checks the MBS and, if it is not already infected, infects it by direct action. Discussing this strategy in his notes on Stealth.A, Ludwig states that 'The infection mechanism for moving from a floppy to a hard disk must take advantage of this little mistake on the user's part to be truly effective. This means hard drives should be infected at boot time.'

The virus reserves 4K of memory. Thus, on a 640K machine, running CHKDSK will report 651,264 bytes rather than the normal 655,360 bytes, and using DEBUG to dump the word at 0000:0413h, one will find the value 027Ch (as bytes this will appear as 7C 02). Running CHKDSK on an infected 3.5-inch floppy (720k or 1.44M) will also report 3072 bytes in bad clusters.

The virus stealths the infected boot sector on floppies and the infected MBS by returning an image of the stored original on disk reads. The other six sectors are stealthed on the hard drive by returning a buffer filled with nulls. On floppies, however, these six sectors are not stealthed.

The method of checking disks for previous infection is also identical to Ludwig's virus. In the *Little Black Book of Computer Viruses*, he writes: 'The Stealth virus uses its own code as an ID. It reads the boot sector and compares the first 30 bytes of code (starting after the boot sector data area) with the viral boot sector. If they don't match, the disk is ripe for infection.' The routine he then presents is identical to that found in Stealth.B.

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.

Interestingly, there is a minor difference in the 30 bytes checked between Stealth.B and Ludwig's prototype. Thus, Stealth.A and Stealth.B cannot recognize one another. The difference consists of only two swapped instructions, but in the event of an infection by both viruses, the system would become unbootable.

Damage

Stealth.B does not contain any intentionally damaging code, but has been reported as wreaking havoc with some memory managers. In my testing I found that, on my 386 test machine, starting *Windows* would bounce back to the DOS prompt. Interestingly, the same test on my 486 machine produces an error message I cannot recall seeing before.

On the 486 machine, I have a permanent *Windows* swap file which uses 32-bit access. When WIN386.EXE attempts to load its disk driver, the following message appears:

```
The Microsoft Windows 32-bit disk driver
(WDCTRL) cannot be loaded. There is
unrecognizable disk software installed on this
computer.

The address that MS-DOS uses to communicate
with the hard disk has been changed. Some
software, such as disk-caching software,
changes this address.

If you aren't running such software, you
should run a virus-detection program to make
sure there is no virus on your computer.

To continue starting Windows without using the
32-bit disk driver, press any key.
```

Pressing a key leaves you back at the DOS prompt. This 'feature' of the virus will have an obvious impact on today's enterprise environment, which depends so much on *Windows* productivity software.

Damages, as with most viruses, will be measured in wasted time, shaken users, and more concrete clean-up costs. I would remind the reader that this last item (cost of clean-up), depends on the number of PCs on the site rather than the number of PCs infected. For example, if two computers at a site with 200 machines are infected, then 200 PCs and all their associated floppies must be examined for infection. This is where time is wasted and costs soar.

Like Wildfire

In the few months since my initial analysis late last year, the virus has spread to many other parts of the US apart from Florida. In the month of April alone, Stealth.B was one of the ten most reported viruses in the United States according to our statistics here at *Symantec*, with reports received from states as far apart as New York and California.

A Stealth.C variant has also been reported in the wild in the US. Moreover, the AMSE virus (*VB*, May 94, p.11) may also be closely related, since it and Stealth.B both contain the text string 'AMSESLIFVASRORIMESAEP' and the description of that virus is quite similar to the Stealth family. Unlike Stealth.B, AMSE prints a message and may be a hack of a Stealth family member.

Electronic Arson

The fire I was monitoring was the first in a series which swept through the Los Angeles area. It was set by an arsonist. Some of the other fires were called 'copycat' fires because they were set by arsonists mimicking the first; fires which would otherwise never have been set. In like manner, would Stealth.B and its ilk now be spreading like wildfire if the virus upon which it is based had never been written?

Watching the fire after analysing Stealth.B, I could see little difference between an arsonist and a virus programmer. Both start something which spreads uncontrollably and can cause extensive damage.

In fact, I began wondering if there are books available in the United States written by arsonists, for arsonists. I can even imagine how one might start with a mock disclaimer: 'Never set fire to anyone's home or business and this is exactly how you should do it.'

Stealth.B

Aliases: STB, stelboo, AMSES.

Type: Master Boot Sectors of fixed disk drives, and boot sectors of diskettes.

Self-recognition on Disk:

Compares 30 bytes (0Fh words) of code in the boot sector with code in memory.

Self-recognition in Memory:

None.

Hex Pattern: The following pattern is that used by the Stealth.B virus as its own self-recognition string:

```
FA33 C08E D08E D88E C0BC 007C
FBB1 06A1 1304 D3E0 2DE0 078E
C083 2E13 0404
```

This differs from that used by the virus published in the *Little Black Book*:

```
FA33 C08E D08E D88E C0BC 007C
FBB1 06D3 E0A1 1304 2DE0 078E
C083 2E13 0404
```

Intercepts: Int 13 for infection and stealth.

Trigger: None.

Removal: Disinfection possible by replacing original Master Boot Sector under clean system conditions.

VIRUS ANALYSIS 2

Argyle: Viruses and the i386

Eugene Kaspersky

Every week, a large number of archives claiming to be virus libraries are sent to anti-virus researchers, each containing new viruses, Trojan programs, corrupted programs and data files. The vast majority of these new viruses are written for the PC running standard *MS-DOS*, and are compatible with machines ranging from the humble XT to the speedy Pentium. Occasionally, one encounters a virus which uses i286 instructions, such as PUSHA/POPA and PUSH/POP IMMEDIATE, but by and large, much of the functionality of the more powerful *Intel* processors is left unused.

Towards the end of 1993, viruses written for the i286/i386 and above began to turn up. The first such virus reported to *Virus Bulletin* was PMBS, the protected mode boot sector virus (*VB*, October 1993, pp.9-11), which runs the i386 processor in its protected mode. This was quickly followed by Pure, which will only install itself into Upper Memory Blocks, and Pink Panther, which uses i286-specific code during installation. The next sample to be added to this class of viruses is Argyle, which uses i386 commands in order to hide its presence in memory.

Chinese Roots

One of the developments in the anti-virus world is that most vendors are attempting to standardise on a single naming convention. This means that it is unacceptable to name new viruses at random: the name should reflect some attribute of the virus. If the new virus is a variant of an older 'parent' such as Jerusalem or Vienna, it takes its name from that sample. Thus, a new Jerusalem variant would be named Jerusalem.B, and so on.

If the virus is completely new, the name is extracted from any internal text messages stored within the virus code. If there are no messages, the name is drawn from the virus effects, or from some other feature.

In the case of the Argyle virus, there is a text string at the end of the virus code, but it is in Chinese, and therefore not wholly suitable for a virus name. As the virus contains no other features apart from its stealth tricks, its name was taken from the host file in which it was distributed.

Installation and Hiding in Memory

The virus is a memory-resident parasitic EXE file infector which uses a polymorphic engine in order to make it more difficult to detect. When an infected file is executed, the virus decrypts itself by XORing double words of the virus code with an encryption key, before passing control to its installation routine.

The installation routine first carries out an 'Are you there?' call to check whether the virus is already loaded and active. This consists of calling Int 21h with the contents of EAX set to FFFFFFFFh. If the virus is already loaded, the virus returns 12345678h in the same register.

If the call is unanswered, the routine attempts to install itself into system memory. It checks the list of Memory Control Blocks, and searches for the last memory block, which it decreases in length. The virus then copies itself into this newly created space and points the necessary interrupt vectors to this area.

Into the UMB…

One of the unusual features of the virus is its ability to install itself into the Upper Memory Blocks (UMB). These are blocks of memory which have addresses above video memory. When first executed, the virus attempts to install itself into free UMBs. If there are no free UMBs, the virus installs itself into conventional memory. The virus uses the same routine for creating and maintaining its area of both conventional and upper memory.

> *"anti-virus utilities cannot gain access to the real addresses of the 'virus alarm' interrupts such as Int 13h, 21h, 25h and 26h"*

The second feature of Argyle is its use of an i386 'trick' in order to hide its own code. After the virus has become memory-resident, it checks the processor mode. If this is set to real mode (that is, DOS was loaded without a memory manager in place, and the DOS session is not under *Windows* or *OS/2*, etc.) and the virus is loaded into conventional memory, it calls a special routine in order to make memory detection more difficult. This stealth routine consists of copying the entire interrupt vector table into the virus' own memory block, and loading a pointer to this copy into the interrupt descriptor table.

As a result, the processor will use the virus' own copy of the interrupt table when searching for the address of an interrupt vector, instead of using the original table stored at 0000:0000-03FFh. Once this operation is complete, any changes made to the 'usual' interrupt vectors have no effect whatsoever: the entire table could be filled with zeros, and the machine would still function!

This trick means that standard debugging and anti-virus utilities will not work correctly, as the trace vectors Int 01h and Int 03h can no longer be set. Additionally, anti-virus utilities cannot gain access to the 'virus alarm' interrupts

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.

such as Int 13h, 21h, 25h and 26h, as most utilities of this type directly access the standard interrupt table, or use the Get Interrupt Vector (Int 21h, subfunction 25h) and Set Interrupt Vector (Int 21h, subfunction 26h) calls.

The final part of the installation routine traces and hooks the BIOS disk handler, Int 13h, Int 21h, and Int 09h, the keyboard handler. Once this process is completed, control is returned to the host program.

Fishing for Interrupts

When memory-resident, Argyle intercepts several of the Int 21h subfunctions. These are 3Dh (Open Handle), 4B00h (Load and Execute), 4B01h (Load), 56h (Rename File), 3Eh (Close Handle), 4Eh (Find First), 4Fh (Find Next), 57h (Get/Set File Date and Time), 25h (Get Interrupt) and 35h (Set Interrupt). Finally, it checks for an Int 21h call with EAX=FFFFFFFFh, the virus' 'Are you there?' call.

The virus infects executable files when it intercepts a Rename, Execute or Load call. In order to avoid multiply infecting files, it checks the file date and time stamp, and only infects files with a file date below 2080. This method of infected file location has become almost standard, and was first used by the 4K virus.

Stealth Routines

In order to check that the file is an EXE file, Argyle examines the first two bytes of its potential victim. If these are 'MZ', indicating that the file is in the EXE file format, the infection routine is called. This alters the EXE header, calls the polymorphic code generation routine, and appends the encrypted virus code at the end of the file. Finally, 100 years is added onto the file date. During the infection process, Argyle hooks Int 1Bh (Ctrl-Break) and Int 24h (the DOS Critical Error handler).

Files are also infected during the Close function. This is carried out in the same way, except for the fact that the virus uses information stored in the undocumented System File Tables in order to complete the process.

When a call to the Open File function is intercepted, the virus calls its stealth routine. If the file is infected, the virus loads it into memory, traces the execution path, and decrypts the attached virus body, including the original EXE header of the infected file. The virus then restores this copy of the EXE header and truncates the file to its original length. Thus, the file is disinfected before being passed back to the calling function.

The virus uses the Find First/Find Next and Get/Set File Date/Time stamp functions for a stealth routine which substitutes the length and file date of infected files with that of the original host file.

The interception of the Get/Set Interrupt Vector function is used as an addition to the memory stealth algorithm. The virus executes these calls in order to prevent passing control to other programs or to memory managers. I see no reason why the virus does this, although it is possible that the author included this feature in order to ensure compatibility with a tool he was using.

Trigger Routine

Argyle contains two different trigger routines. One intercepts Int 13h read and write sector requests. During every 256th call, the virus sets a randomly-selected bit of data to its complementary value. This will cause gradual corruption of data stored on disk.

The second trigger routine monitors the Int 09h handler. If the user presses Ctrl-Alt-Del in order to reboot his machine, the virus checks an internal counter and the system timer. Depending on their values, a message is displayed in Chinese. The only part of this text which is in standard ASCII characters is the date, 'Dec 1993'.

Conclusions

Although Argyle does not pose too many problems for antivirus vendors, the general trend of i386 viruses is cause for some concern. By utilising the extra functionality of the more powerful *Intel* processors, it is possible to increase vastly the degree of stealth which can be applied.

Virus writers have yet to gain much experience in writing i386-specific code, which opens up a plethora of ways in which to subvert the PC. What will be the next development? We can only wait and see.

Argyle

Aliases: None known.

Type: Memory-resident, parasitic file infector, stealth and polymorphic.

Infection: EXE files only.

Self-recognition in File:

It the year value in file date stamp is above 2080 (plus 10 years), the virus does not hit the file.

Self-recognition in Memory:

'Are you there?' call consists of calling Int 21h, with EAX=FFFFFFFFh. 12345678h is returned in EAX.

Hex Pattern: No search pattern is possible in files.

Intercepts: Int 21h for infection and stealth, Int 13h for damage, Int 09h for trigger routine.

Trigger: Corrupts data on disks, displays message, and reboots computer.

Removal: Under clean system conditions, identify and replace infected files.

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.

COMPARATIVE REVIEW

The Review, Reviewed

Dr Keith Jackson

One of the regular items published in *Virus Bulletin* is the anti-virus product review, eagerly anticipated by developer and user alike. Do these articles serve a useful purpose, or are they simply a waste of time? It is virtually impossible to answer such questions for reviews of individual products: the answer depends far too much on the particular product being reviewed. Comparative reviews, however, are a different matter altogether.

When reviewing anti-virus products, it is easy to test how well a scanning program checks a set of virus-infected files and from there write a results-based review: many reviews written in computer magazines seem to be produced in this way. Writing a fair, objective and (most importantly) useful review is more difficult. If conducted well, a comparative review of several scanners may be the fairest way of gauging the performance level of individual scanners.

Timing Comparisons

When I review a scanner, I measure the time taken to scan the hard disk on my test computer, and provide similar results for two other well-known scanners. This is not done to publicise the other scanners, but as recognition of the fact that an absolute measurement of the time taken to carry out a scan does not mean a great deal on its own.

Scan time is affected by the processor, processor clock speed, the type of hard disk, the partition structure on the hard disk, how files are fragmented on the hard disk, the operating system, what software is executing concurrently, etc. The list is so long, and so intertwined, that what is quickest on one computer may well not be so on another.

So, is the time taken to scan a disk of any relevance? Alone, it means little: what is important is whether the user perceives it to be excessively long. As is the way with such things, it is a subjective criterion. However, comparative measurements of scanning speed do allow users to select products with a reasonable scan time, and (if speed is important) to reject the slowcoaches.

Scan Time for Virus Test-Sets

The time taken to scan a set of virus-infected files should not be a prime factor. Overall scan time may be important, but that the scanner slows down and cogitates on a file suspected to be virus-infected seems of little consequence, unless the time delay is exorbitant, and the file is in fact clean. Given that these measurements are contained in the comparative scanner review, it merely means that the anti-virus developers who score well in this test are keeping well ahead of the game, and pouring in the necessary resources to keep their scanner speed acceptable high while still identifying increasingly polymorphic virus code.

The Standard Test-Sets

If I had my way, the 'standard' test-sets would either be varied so frequently that nobody would be certain as to what was in them or, most likely, would be scrapped altogether. *VB* gets many calls from developers asking for copies of their test-set [*there was even one request for 'regular updates' of all test-sets used! Ed*]. The fundamental issue is who chooses the test-set contents. The individual with this responsibility can bias test results - this may not even be a conscious decision.

VB actually goes to great lengths to be fair in its reviewing process, but, as must be expected, some people have easier access to test-sets than others. Were I a complete unknown, and asked *VB* for a copy of the standard test-set, in order to develop an anti-virus program, would they oblige? The answer must be no: a malicious person could request a copy of all currently known 3000-plus viruses, and propagate them to far-flung corners of the globe. Conversely, if the person involved were known in the industry, the response to such a request would almost certainly be 'yes'.

> *"any manufacturer who performs particularly badly in polymorphic tests is almost certainly having difficulty devoting the required manpower to the problem"*

Ultimately, use of a standard test-set probably leads to the development of anti-virus software which scores highly against that particular test-set, and which may well react differently when used in the 'real' world. The existence of standard test-sets seems to reinforce a hegemony among the leading product developers, which at best is unhelpful, and at worst will lead to sterile 'numbers-game' competition amongst a few companies, with newcomers excluded. Users deserve better than this.

Detection Rate 'In the Wild'

The 'In the Wild' test-set is a continuously updated set of viruses with examples of viruses known to have been found on a user's site. Such a test-set is a necessary precaution: there are individuals who, on writing a virus, stake their claim to fame by sending a copy to an anti-virus company. They do not often bother to release a copy of the virus to let it wend its merry way around the world.

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.

Such a virus is rarely seen outside laboratories of anti-virus developers, who unfortunately must take receipt of every virus seriously: there is no immediate way of telling whether a virus is in the wild. A virus may be the most dangerous ever written, but is not harmful (apart from inducing panic) unless released into the computer community.

Only viruses which have been detected with certainty on a few user sites are included in the 'In the Wild' test-set. It is a telling comment on the psyche (and software capability) of virus writers that this test-set is but a small subset of the number of viruses actually known to exist.

The detection rate on the 'In the Wild' test-set is probably the most important measurement in the entire comparative review. I contend that any scanner which cannot detect 100% of such viruses (excepting viruses which have just come to light) should be viewed as less than adequate.

Companies which develop anti-virus scanners regularly exchange virus sets, and know well which viruses occur in the wild. If a company cannot, or will not, update its scanner to deal with a known problem, this may be taken as indicative of its attitude as a whole: avoid its products like the plague. Having said that, it is noticeable that most scanner developers do indeed try hard to be very close to 100% detection for viruses known to be in the wild.

Polymorphic Detection

Not all viruses can be detected by scanning for a fixed sequence of bytes. Those which encrypt themselves with a unique key each time they replicate and which can change their structure are known as polymorphic. They require a scanner to use techniques beyond pattern matching.

The *VB* comparative review shows how well scanners can detect extremely polymorphic viruses. It is in reality a measure of the amount of research a company puts into keeping its scanner up to date. Only developers who invest the requisite amount of effort to keep up to date with such problems will survive in the long term.

Improving the Test-Sets

The shortfalls associated with using standard test-sets were discussed earlier in this article: what improvements could be made to a comparison of various scanners?

Some products are prone to the erroneous detection of the presence of a virus in a file, an error known as a 'false positive'. It has been requested that *VB* measures the incidence of such 'false positives' in its product reviews. Unfortunately, factual information about false positives can only be obtained if a product is tested against all known software packages, which is somewhat difficult to achieve.

There is no unique solution; however, most reputable anti-virus developers maintain at least one system containing samples of every software package they can obtain. Scanners are tested against this store of files before being released: it should not be beyond the wit of man to find a way of pooling this resource across several developers when comparative reviews are being considered. Such action would tend to alleviate the problem of who chooses the content of the 'false positive' test-set. As false alarms waste more money than viruses, the developers of the best scanners should be keen to contribute to such a scheme.

Which Product is the Right One?

There are serious considerations when it comes to the choice of an anti-virus product. Some of the most important are:

- If a scanner performs badly against the 'In the Wild' test-set, consider it no further - its developer is obviously shirking his responsibilities.
- If a scanner is very slow, users will not operate it. There is thus little point in purchasing it. Speed measurements are subjective, and can only really be assessed by an individual user when measured on his own computer system(s). Try before you buy.
- If a scanner continually throws up false positives despite being used correctly, ditch it.

Beyond these rules (which admittedly do not form a sufficient basis for a purchasing decision), the choice is more difficult. Many scanners show signs of falling behind in the game of 'catch' played between virus writers and anti-virus companies. Such effects are difficult to measure accurately, but any manufacturer who performs particularly badly in polymorphic tests is almost certainly having difficulty devoting the required manpower to the problem. Deducing how to detect a polymorphic virus accurately is not always the greatest feat of Sherlock Holmes-like detection: it is, however, time-consuming, fiddly, and manpower intensive.

Conclusions

There are few positive gains to be had from comparative scanner reviews, only negative ones. For example, it would be ridiculous to choose one of the best (i.e. most accurate) scanners on the grounds that it detected 1% more viruses than its closest competitors. However, deciding not to purchase one of the less efficient scanners because it was 20-30% worse than the best scanners currently available would be a reasonable decision. A high detection rate is a prerequisite for a scanner.

You should check that the scanner you choose achieves an acceptable detection rate when running in the mode which gives acceptable speed - not just in a fearfully slow 'high security' mode.

The protocol used for these tests is published in detail, and any software developer can have his product included by contacting the editor of *VB*. The magazine always attempts to provide objective evidence about anti-virus scanners, and, within the constraints discussed above, the results can be positively illuminating.

COMPARATIVE REVIEW

VB Scanner Review: July ’94

Mark Hamilton

In the six months since *VB’s* last comparative scanner review, there have been more than the usual amount of changes in the anti-virus world, from withdrawals of products, to births of new ones, and takeover bids.

Total Control has withdrawn its *VIS*, although the company has told *VB* it will continue to support existing users in the short term. Further afield, *Symantec* has acquired *Fifth Generation* and signed an agreement to buy *Central Point.* All this makes the future of *Untouchable* and *Search and Destroy* (both from *Fifth Generation*) rather uncertain. The fate of *CPAV* is also as yet undecided.

Symantec’s Norton Anti-Virus is noticeable by its absence: despite their initial willingness to participate, and numerous telephone calls and faxes to both its UK and US offices, nothing was forthcoming. As testing was completed, a package did arrive from *Symantec* - unfortunately, it was the *NetWare* version, and could not be included.

Testing Protocol

Products were pitted against four test-sets: ‘In the Wild’, with 109 samples of file infectors known to be at large; a Boot Sector set containing nine commonly found viruses; the ‘Standard’ test-set, consisting of 227 file infectors; and a set of 750 polymorphic viruses. For further details, see the table at the end of this article.

Disk scanning speed on both an uninfected hard drive and a clean floppy drive was tested on all products. For the first time, a speed test on an infected hard drive was also included, giving users some idea of how long a scan might take when checking a machine with a number of infected files on it. The Polymorphic test-set was chosen for this speed degradation test, since that type of virus represents the worst case for most anti-virus products. Tests were performed on a 16MHz *Dell* 386SX with 4MB memory, which is a typical office machine.

Product speeds are given in kilobytes per second. This represents the times taken to scan an uninfected hard drive containing 165 executable files spread across 14 directories and occupying 6,917,984 bytes, an infected hard drive containing 750 executable files spread across 8 directories and occupying 8,510,417 bytes, and a clean high-density floppy disk containing 51 executable files occupying 1,400,626 bytes in a single directory.

Conclusions about products were made according to detection rate: the higher the overall score on detection, the better the overall placing. Although scanning speed is an important consideration, it is of paramount importance that a product correctly detects as many different viruses as possible. Anti-virus software manufacturers were asked to supply the version of their scanner which was shipping at the beginning of May. No special releases were accepted, unless otherwise stated.

Avast! Version 6.20

In the Wild:	97.2%
Boot Sector:	0.0%
Standard:	99.6%
Polymorphic:	90.0%

The failure of *Avast!* (*Alwil Software*, Prague) to detect boot sector viruses is somewhat puzzling, given that its other results were most encouraging, particularly in the Polymorphic test-set. It is, however, one of the slower packages tested, with a scanning speed of only 109KBytes per second.

ASP Integrity Toolkit Version 3.7.9

In the Wild:	67.9%
Boot Sector:	44.4%
Standard:	55.1%
Polymorphic:	0.0%

This package, originally written by Dr Fred Cohen, is now distributed and maintained by *Sikkerheds Radgiverne ApS* in Copenhagen. Results were obtained from running an early version of Fridrik Skulason’s *F-Prot* (March 1992) which is included on the installation disk, but not installed. Unsurprisingly, this product had the worst detection results, missing boot sector viruses Parity Boot, BFD-451, Monkey, Jack the Ripper and Quox.

AVScan Version 1.51a

In the Wild:	100.0%
Boot Sector:	100.0%
Standard:	100.0%
Polymorphic:	82.1%

AVScan from *H+BEDV* is one of the best packages available in terms of detection, and as an added plus, it is freeware! It gets perfect scores in all the principal tests but, as do so many others, it fails on a number of the polymorphic viruses. It is a great pity that the commercial version of the product, *AntiVir IV*, is only available in German.

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.

Central Point Anti-Virus V2.0

In the Wild:	75.2%
Boot Sector:	44.4%
Standard:	92.5%
Polymorphic:	Failed to complete

Central Point submitted three packages for review: *Central Point Anti-Virus for DOS v2.0*, *Central Point Anti-Virus for Windows v2.0* and *Central Point Anti-Virus for DOS v2.1* (see p.18). The scores for the two v2.0 products are identical and uninspiring, especially against the boot sector viruses.

The Doctor Version 94.04

In the Wild:	96.3%
Boot Sector:	100.0%
Standard:	97.3%
Polymorphic:	80.0%

This product is developed by Roger Thompson, of *Thompson Network Software* (formerly of *Leprechaun Software).* The package is being updated and sold in the USA, and is one of the more secure, if slower, available.

F-Prot Professional Version 2.12a

In the Wild:	99.1%
Boot Sector:	100.0%
Standard:	100.0%
Polymorphic:	81.7%

A very high detection rate puts this package, from *Frisk Software International* in the ‘top 10’. Surprisingly, after detecting 100% of polymorphic infections in the last comparative review, tests this month showed a significantly lower detection rate, which has marred the product’s previous excellent results. However, it is still one of the top packages available.

Iris Anti-Virus Version 4.20.25

In the Wild:	89.0%
Boot Sector:	66.7%
Standard:	97.8%
Polymorphic:	80.3%

A set of mediocre test results from *Iris*. Like several other products, the detection of boot sector viruses lets *Iris Anti-Virus* down, missing BFD-451, Jack the Ripper and Quox. All the boot sector viruses used in the test-set are in the wild. This deficiency needs to be addressed immediately.

Microsoft Anti-Virus (with MS-DOS 6.2)

In the Wild:	68.8%
Boot Sector:	44.4%
Standard:	90.7%
Polymorphic:	Failed to complete

This product was written by *Central Point*, and although released after *Central Point’s* version 2.0, it has a worse detection record in most categories. It too misses Parity Boot, Monkey, Jack the Ripper, Quox and BFD-451, and suffers the same problems regarding polymorphic viruses as its big brother. Poor.

IBM Anti-Virus for DOS Version 1.05

In the Wild:	95.4%
Boot Sector:	100.0%
Standard:	98.7%
Polymorphic:	80.0%

Depending on where and when you buy *IBM’s PC-DOS 6.3*, this may be the version of its anti-virus product included (*PC-DOS* sold in Europe may have a slightly earlier version of the scanner). The review version was supplied by an *IBM* value-added reseller in the US. As an add-in extra to vanilla DOS, this product represents terrific value for money, and sets the standard for *Microsoft*.

PC Vaccine Pro (PCVP) version 2.0

In the Wild:	96.3%
Boot Sector:	88.9%
Standard:	95.1%
Polymorphic:	79.1%

This product, from *Computer Security Engineers*, missed four of the new viruses found in the wild, in addition to the Jack the Ripper boot sector virus. It is, however, the fifth fastest at scanning a clean hard drive.

Scan 9.25 Version 114

In the Wild:	97.2%
Boot Sector:	88.9%
Standard:	98.7%
Polymorphic:	89.6%

This is the version of *McAfee Associates’ Scan* which was shipping at the time of testing, although plans are afoot for *McAfee Scan* version 2.0, which is already in Beta release (see overleaf). Similar scores to last January’s test.

	In the Wild (109)	Boot Sector (9)	Standard (227)	Polymorphic (750)	Overall (100)
Avast!	106	0	226	675	72.0
ASP	74	4	125	0	42.0
AVScan	109	9	227	616	96.0
CPAV/DOS 2.0	82	4	210	Failed to complete	53.0
CPAV/DOS 2.1	93	5	211	Failed to complete	58.5
CPAV/Win 2.0	82	4	210	Failed to complete	53.0
The Doctor	105	9	221	600	93.4
F-Prot Professional	108	9	227	613	94.8
IBM Anti-Virus	104	9	224	600	93.5
Iris Anti-Virus	97	6	222	602	83.5
Microsoft Anti-Virus	75	4	206	Failed to complete	51.0
PCVP	105	8	216	593	90.0
Scan v114	106	8	224	672	93.6
Scan v2.0 Beta	100	8	220	279	78.7
Search and Destroy	99	7	219	500	83.0
SmartScan	75	7	215	0	60.3
Sweep	109	9	227	674	97.5
S&S AVTK	108	9	224	682	97.2
ThunderBYTE	108	9	227	725	99.0
VET	107	9	224	672	96.6
Virex-PC	100	9	218	598	92.0
Virus Buster	85	8	210	607	85.1
Virus Buster Lite	84	8	209	607	85.0
Virus Control	107	9	221	581	93.3
ViruSafe	90	9	214	550	88.0
Vi-Spy	109	9	227	675	97.5

The final scores! Overall scores out of 100 have been calculated as follows: viruses known to be in the wild (the results from the In the Wild and Boot Sector tests) account for 50% of the final score. The remaining 50% is made up by combining the scores from the Standard and Polymorphic test-sets. In this comparative, nobody escaped without missing at least 26 viruses.

Scan Version 2 (Beta)

In the Wild:	91.7%
Boot Sector:	88.9%
Standard:	96.9%
Polymorphic:	37.2%

Version 2 of *McAfee's Scan* is claimed to scan more quickly than its predecessor. This is true on clean drives but, as results show (see table p.17), it is very slow on an infected system. The boot sector virus Quox foxed both versions, and polymorphic detection is much lower than in version 1.

SmartScan Version 3.05

In the Wild:	68.8%
Boot Sector:	77.8%
Standard:	94.7%
Polymorphic:	0.0%

Visionsoft's contribution missed many viruses: this fact, coupled with its slow scanning speeds, contribute to making it one of the less efficient anti-virus software packages on the market. The boot sector viruses which it failed to detect were BFD-451 and Quox. Possibly falling behind?

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.

	Floppy Read (Kb/sec)	Clean HD Read (Kb/sec)	Infected HD Read	Degradation
Avast!	15.2	109.0	27.2	4.0 x slower
ASP	6.5	58.2	58.1	n/a
AVScan	17.6	139.6	32.1	4.3 x slower
CPAV/DOS 2.0	14.4	99.4	Failed to complete	n/a
CPAV/DOS 2.1	15.0	99.4	Failed to complete	n/a
CPAV/Win 2.0	11.8	87.7	Failed to complete	n/a
The Doctor	13.8	51.8	17.0	3.0 x slower
F-Prot Professional	20.7	182.6	21.4	8.5 x slower
IBM Anti-Virus	13.4	91.3	40.7	2.2 x slower
Iris Anti-Virus	12.3	24.0	49.2	2.0 x faster
Microsoft Anti-Virus	27.9	114.5	Failed to complete	n/a
PCVP	20.1	257.9	96.6	2.6 x slower
Scan v114	13.1	37.1	13.5	2.7 x slower
Scan v2.0 Beta	21.3	57.6	3.5	16.5 x slower
Search and Destroy	21.5	339.5	40.8	8.3 x slower
SmartScan	12.5	112.8	38.1	3.0 x slower
Sweep	7.6	39.7	26.3	1.5 x slower
S&S AVTK	27.9	196.7	2.4	82.8 x slower
ThunderBYTE	59.5	750.6	9.7	77.4 x slower
VET	16.4	119.6	86.5	1.4 x slower
Virex-PC	7.0	35.9	13.4	2.8 x slower
Virus Buster	25.8	153.5	3.3	46.5 x slower
Virus Buster Lite	40.2	293.7	3.3	89.0 x slower
Virus Control	23.7	148.5	42.0	3.5 x slower
ViruSafe	27.4	270.2	129.9	2.1 x slower
Vi-Spy	23.6	150.1	26.0	5.4 x slower

Once again, the faster products are not necessarily the least accurate. *ThunderBYTE*, the most accurate product on the test-sets used, is also the most fleet-footed, scanning at an impressive 750KBytes per second on a clean hard drive. For the first time, scan times on an infected hard drive have been included - these give some measure of the time taken on a machine containing several infected files.

Virus Buster Version 4.03.05

In the Wild:	78.0%
Boot Sector:	88.9%
Standard:	92.5%
Polymorphic:	80.9%

A disappointing result for *Leprechaun Software*. The company also submitted a package called *Virus Buster Lite* (see tables). Scores for this were identical to the full version, with the exception of the In the Wild test-set, where one extra virus was missed.

VET Version 7.63

In the Wild:	98.1%
Boot Sector:	100.0%
Standard:	98.7%
Polymorphic:	89.6%

Cybec's VET suffers less from speed degradation when scanning infected drives than most other packages, only slowing by a factor of 1.4. This is the top package from the southern hemisphere, and has improved considerably since the last review.

Search and Destroy Version 28.02

In the Wild:	90.8%
Boot Sector:	77.8%
Standard:	96.5%
Polymorphic:	66.7%

Search and Destroy was licensed last year by *Novell* from *Fifth Generation*, since when the company has been acquired by *Symantec*: whether the product continues to be developed or supported has yet to be seen. When this scanner runs, it displays an internal version date of 22 July, 1993, going some way to account for its indifferent detection capabilities.

Sweep Version 2.60

In the Wild:	100.0%
Boot Sector:	100.0%
Standard:	100.0%
Polymorphic:	89.9%

Sweep from *Sophos* is, as always, an efficient and dependable product, despite the slow scanning speeds in its ‘full sweep’ mode.

Virex-PC Version 2.93

In the Wild:	91.7%
Boot Sector:	100.0%
Standard:	96.0%
Polymorphic:	79.7%

Although it is not one of the faster packages, this product from *Datawatch* ended up in the top half of the table. Better ‘in the wild’ detection is needed.

ViruSafe Version 6.1

In the Wild:	82.6%
Boot Sector:	100.0%
Standard:	94.3%
Polymorphic:	73.3%

This package from *EliaShim* will not allow scanning of more than one floppy: the program fails to note that the disk has been changed and attempts to locate files from the previously scanned floppy. The program must be exited between scans and, as it appears to leave virus signatures in memory, the PC rebooted or memory scans disabled. Checking an infected boot sector results in the reporting of a corrupted text string instead of a virus name.

Central Point AV/DOS Version 2.1

In the Wild:	85.3%
Boot Sector:	55.6%
Standard:	93.0%
Polymorphic:	0.0%

This is the latest version of *CPAV*, phased in while this review was underway. It failed to detect Monkey, BFD-451, Jack the Ripper and Quox boot sector viruses. This package has always suffered from bugs - the inability to scan polymorphic viruses without crashing the PC, and its propensity to leave virus signatures in memory.

The results in this current review show that nothing has changed: all three versions submitted for testing crashed when run against the Polymorphic test-set. These points have been mentioned time and again: it is totally unacceptable that the problems still exist. Whether this will be fixed or not remains to be seen - at this time, the entire future of *CPAV* seems uncertain.

S&S AVTK for DOS Version 6.54

In the Wild:	99.1%
Boot Sector:	100.0%
Standard:	98.7%
Polymorphic:	90.9%

Another set of good results from *S&S International*, placing the product well up in the overall rankings. Due to changes to the way in which the *AVTK* indentifies polymorphic viruses, it slows down greatly on an infected machine. However, this is easily made up for by its ability to carry out precise virus identification on such files.

ThunderBYTE Anti-Virus Version 6.20

In the Wild:	99.1%
Boot Sector:	100.0%
Standard:	100.0%
Polymorphic:	96.7%

ThunderBYTE from *ESaSS* was rated the best package in the last comparative review (*VB*, January 1994, pp.14-19). Overall, it is still the highest-scoring package, and also has the highest polymorphic detection rate.

Its author, Frans Veldman, was very keen to have this version reviewed, which features a new detection engine specifically geared to polymorphic viruses. The product is capable of examining the encrypted code within a polymorphic virus, making more precise identification possible. Still blindingly fast on a clean machine, *TBAV* slows down by a factor of over 75 on an infected one.

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 Control Version 3.42

In the Wild:	98.2%
Boot Sector:	100.0%
Standard:	97.4%
Polymorphic:	77.5%

This product, from *Norman Data Defense Systems*, was first reviewed in *Virus Bulletin* earlier this year (May 1994 pp.17-19). The detection rate of polymorphic viruses has already improved considerably: where will it go from here? Scanning speeds were quite acceptable; indeed, faster than some of the other ‘high scorers’.

Vi-Spy Version 12.0

In the Wild:	100.0%
Boot Sector:	100.0%
Standard:	100.0%
Polymorphic:	90.0%

RG Software’s Vi-Spy emerges as one of the top scoring anti-virus packages tested, equal second with *Sweep*, and outdone only by *ThunderBYTE*. Speeds are comparable to the last review; in the top half, but not the fastest. Reliable.

Final Comments

Anti-virus products are constantly changing; new ones appear as old ones fade away. Only the best can hope to stay the course. Unsurprisingly, results varied wildly, from the sublime to the ridiculous. Every product should be able to detect *all* viruses in the In the Wild test-set: in fact, only three (*Vi-Spy*, *Sweep*, and *AVScan*) managed this.

It is also important that products are capable of detecting polymorphics other than Mutation Engine-generated viruses. It is incredible that six products detected no polymorphic viruses at all (all three from *Central Point*, and the product from *Microsoft*, crashed every time the test was attempted), and that a further six detected less than 80% of this test-set. Even though the Polymorphic test-set has been overhauled and updated for this review, the only very new virus included is Pathogen, and this accounted for less than 7% of the samples.

Certain manufacturers will need to make radical changes to their products if they wish to remain contenders: a significant number appear to have become ‘lazy’, not only on scanning speed but, most importantly, on detection rates. Certain other developers have improved considerably since the last review, giving more competition to the vendors, and (as a direct consequence) more choice to the user. Anti-virus software is often the main line of defence against virus attacks: if a product cannot detect reliably and consistently, it cannot be said to be protecting the user. No excuses are good enough.

The Test-Sets

1. In the Wild

4K (Frodo.Frodo.A), Barrotes.1310.A, BFD_451, Butterfly, Captain_Trips, Cascade.1701, Cascade.1704, CMOS1-T1, CMOS1-T2, Coffeeshop, Dark_Avenger.1800A, Dark_Avenger.2100.DI.A, Dark_Avenger.Father, Datalock.920.A, Dir-II.A, DOSHunter, Eddie-2.A, Fax_Free.Topo, Fichv.2.1, Flip.2153.E, Green_Caterpillar.1575.A, Halloechen.A, Helloween.1376, Hidenowt, HLLC.Even_Beeper.A, Jerusalem.1808.Standard, Jerusalem.Anticad, Jerusalem.PcVrsDs, Jerusalem.Zerotime.Australian.A, Keypress.1232.A, Liberty.2857.D, Maltese_Amoeba, Necros, No_Frills.843, No-Frills.Dudley, Nomenklatura, Nothing, Nov_17th.855.A, Npox.963.A, Old_Yankee.1, Old_Yankee.2, Pitch, Piter.A, Power_Pump.1, Revenge, Screaming_Fist.II.696, Satanbug, SBC, Sibel_Sheep, Spanish_Telecom, Spanz, Starship, SVC.3103.A, Syslock.Macho, Tequila, Todor, Tremor (5), Vacsina.Penza.700, Vacsina.TP.5.A, Vienna.627.A, Vienna.648.A, Vienna.W-13.534.A, Vienna.W-13.507.B, Virdem.1336.English, Warrior, Whale, XPEH.4928.

2. Boot Sector

Brain, Form, Italian, Michelangelo, Monkey, New_Zealand_2, Quox, Spanish_Telecom, and V-Sign.

3. Standard

1049, 1260, 1575, 1600, 2100 (2), 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 777, 800, 8888, 8_Tunes, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), AntiCAD (2), AntiPascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Butterfly, Captain_Trips (2), Cascade (2), Casper, Coffeeshop, Dark_Avenger, Darth_Vader (3), Datalock(2), Datacrime, Datacrime_II (2), December_24th, Destructor, Diamond (2), Dir, Diskjeb, DOSHunter, Dot_Killer, Durban, Eddie, Eddie_2, Fellowship, Fish_1100, Fish_6 (2), Flash, Flip (2), Fu Manchu (2), Halley, Hallochen, Helloween (2), Hide_Nowt, Hymn (2), Icelandic (3), Internal, Invisible_Man (2), Itavir, Jerusalem (2), Jocker, Jo_Jo, July_13th, Kamikaze, Kemerove, Kennedy, Keypress (2), Lehigh, Liberty (5), LoveChild, Lozinsky, Macho (2), Maltese_Amoeba, MIX1 (2), MLTI, Monxla, Murphy (2), Necropolis, Nina, Nomenklatura (2), NukeHard, Number_of_the_Beast (5), Oropax, Parity, PcVrsDs (2), Perfume, Pitch, Piter, Polish_217, Power_Pump, Pretoria, Prudents, Rat, Satan_Bug (2), Shake, Sibel_Sheep (2), Slow, Spanish_Telecom (2), Spanz, Starship (2), Subliminal, Sunday (2), Suomi, Suriv_1.01, Suriv_2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Syslock, Taiwan (2), Tequila, Terror, Tiny (12), Todor, Traceback (2), Tremor, TUQ, Turbo_488, Typo, V2P6, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virdem, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Willow, WinVirus_1.4, Whale, Yankee (7), Zero_Bug.

4. Polymorphic

The test-set consists of 750 genuine infections of: Cruncher (25), Coffee_Shop (500), Pathogen (50), Satan_Bug (100), Uruguay_4 (75).

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

Norton on NetWare

Jonathan Burchell

Norton Anti-Virus for NetWare is a virus scanner and program inoculator. It is designed as a complete stand-alone product, capable of protecting *Novell* file servers. The product does not rely on the presence of companion scanners and TSR programs on workstations to achieve file server protection.

The software is capable of scanning both DOS and *Macintosh* files. However, only an NLM version is supplied, so *NetWare 2.0* owners are left out in the cold. It is worth noting that *Novell* has officially announced a retirement day for *NetWare 2.0*: perhaps it is finally time to scrap any 286 file servers in your organisation. Additionally, *NAV for NetWare* requires at least one machine which has *Windows* installed on it for installation and configuration; no DOS utilities are provided.

The software claims to be compatible with all *NetWare versions 3.1x* and *4.0*. I have no reason to doubt the veracity of these claims, but the manual makes no mention of support for special *4.0* features such as file compression and backup media migration.

Documentation

The documentation is a single 60-page manual in the familiar *Symantec/Norton* colours and style. It covers installation requirements and gives an extremely good guide to using the software, clearly presenting each feature and control. Information about what viruses are, and methods of preparation for and dealing with an infection is also included - a useful addition.

Installation

The software arrives on just two 3.5-inch high-density disks (5.25-inch disks are available by contacting *Symantec* directly). Installation is started by typing A:INSTALL at the DOS prompt.

This automatically starts *Windows* on the hard disk, and runs from within *Windows* - a neat trick, which unfortunately failed on my system as I have several copies of *Windows*. The install program found the ‘wrong one’.

The documented method did, however, work if I started the install program from the home directory of the copy of *Windows* I wanted to use, or if I started *Windows* before running the install program. The manual should clarify this, as the organisation of *Windows* on my disk is not that different from a network install - I have one real copy of *Windows* and several local users’ versions.

Once the install program was running, it performed faultlessly, automatically copying the NLM to the SYS volume, making changes to AUTOEXEC.NCF, and installing the *Windows* workstation component. A custom install option allows complete control of the process.

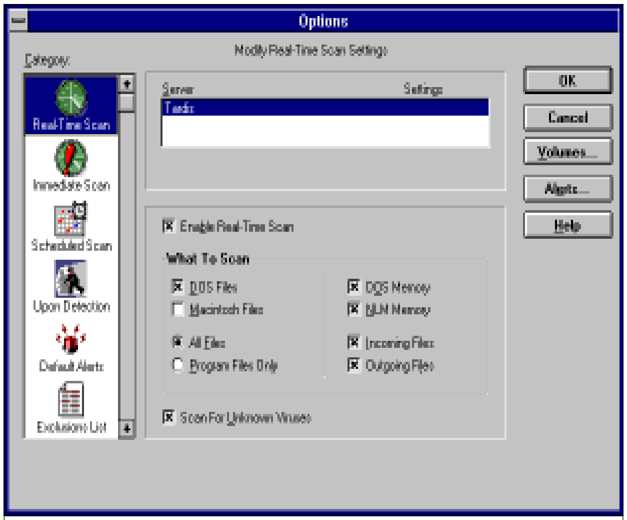
Before performing the install, the installation routine scanned all local drives for viruses. Apart from its dubious worth (scanning from within *Windows* almost entirely rules out a clean boot), it was a nice touch. Only the install software has the capability of local scanning; once installed, the feature is lost.

Some idea of the slickness of the installation routine is reflected in the fact that when it presented the dialogue box for my registration details, it had already filled in my name and company from the information stored in *Windows*!

Administration

Once installation is complete, the software is ready to be configured and used. The NLM has an informative console screen showing status, and keyboard entry allows the NLM to be enabled or disabled, scanning to be started or stopped, and the NLM to be unloaded. The unload feature is not password-protected, so anyone with console access could do this. Control via the console is meant to provide only the most rudimentary of options. The real way to control and configure the software is from a workstation via the administrator’s software.

The *Windows* interface provided is really very good. I found it totally intuitive to use, and in no way limited by being ‘constrained’ to a GUI. It should serve as a model (or is that an icon?) of ‘how to do it’ for other *Windows* products.



Modify Real-Time Scan Settings

Server: Tardis   Settings

☒ Enable Real-Time Scan

What To Scan

☒ DOS Files   ☒ DOS Memory
☐ Macintosh Files   ☒ NLM Memory
● All Files   ☒ Incoming Files
○ Program Files Only   ☒ Outgoing Files

☒ Scan For Unknown Viruses

NAV has one of the nicest user interfaces around - however, its virus detection results are somewhat disappointing.

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.

Concepts and Features

Administering the server (or servers) requires a valid supervisor privileged *NetWare* login. All servers in a domain can be administered at once: this is convenient for dealing with large networks with many servers, as configuration details will automatically (optionally) propagate from server to server within a domain, and infection and activity logs will be stored on the designated master server.

The main window presents summary status information for all the protected servers, together with a button-bar which allows access to configuration options, the virus encyclopaedia, the *NetWare* console and the activity logs. A more traditional menu-bar is also available.

Real-time scanning allows selection between scanning, DOS files, *Macintosh* files, and all files, or just programs, incoming and/or outgoing files and the server DOS and NLM memory space.

"in the 'all files' mode, detection of the polymorphic test-set was appalling, finding less than 1% of all samples"

The immediate scan options are almost identical to the real-time options, with the additional ability to set the maximum amount of server CPU time which can be consumed by the NLM. This should help lessen the impact of scanning on a heavily loaded server, at the expense of increasing the actual scan time. Like the real-time scan, it is possible to select between the global alert list and a custom list, and to select between scanning all servers or scanning particular items, which may be a server, a volume or even a particular directory (and, optionally, subdirectories).

The scheduled scan options are similar to the immediate scan, with the ability to specify a time. The scheduler is very flexible and will store a list of scheduled scans. Scans may be one time only or repeatable. One-time scans are erased from the list after execution. The scheduler allows timed scans, with frequency specified in statements like: 'Monthly on the 1st @ 3.00 AM'. Such simplicity is excellent, allowing a quick and easy interpretation of the current setup.

Actions to be Altered

The Exclusions list option allows the list of file extensions which represent an executable to be altered, and permits specific areas of servers (perhaps those to which no user has write access) to be excluded from a scan. These modifications are global and apply to all types of scanning.

The Upon Detection option controls the action to be taken if a virus is detected, including denying further access to the file, deleting the file, renaming the file, moving the file to a quarantine directory, loading an NLM, and forcing a workstation lockout. The mechanism which allows the quarantine directory to be chosen is wonderful, presenting a graphical browser which starts first with servers, then allows volumes and specific directories to be selected. The directory specified must exist. No option to create one is given.

As well as taking action on detection, *NAV* will send out alerts, which are subdivided into almost any combination of *NetWare* broadcast messages, MHS mail, and pager activation. The *NetWare* messages may be sent to any combination of all users, file user, file owner, file updater, system console, supervisor, or to a specified list of users.

It is not possible to modify messages sent: this is a serious omission, as few corporates want to risk inducing panic by allowing the vendor's virus detection message to propagate. One can, however, modify the pager and MHS messages. The pager option requires additional software and a modem.

Logging and Reporting

NAV provides a feature-rich logging and reporting system. Log files of configuration details, activity, and infections are produced. It is possible to configure the information which goes into the log file precisely, and a useful option allows specification in kilobytes of the maximum size log files may reach. Once they attempt to grow above this size, the oldest information will automatically be discarded.

In addition to good logging, the software provides excellent filtering and display functions, enabling log file reports to be generated. One disadvantage is that the format of the log file is not detailed anywhere, making it difficult to write software to examine them automatically. On the other hand, the configuration and filtering options cater for just about any possibility. The final report can be printed, sent to disk or mailed to a user. One thing I would like to see is the ability to run a report automatically, and mail the results.

Workstations equipped with *Norton Anti-Virus* are able to make use of the server-configured alert and logging mechanism, providing centralised collection of reports.

The on-line help system was extremely well laid out and informative. In addition, a fairly good electronic virus encyclopaedia is included. This is, in fact, the list of viruses for which the scanner searches, and includes basic information on virus types and triggers, in addition to other bits of general information. It is possible to remove a virus from the signature database, but I cannot really think of a good reason to do this: if a file were genuinely causing false positives it would be better to try and exclude it from the scanning, rather than to remove the virus signature from the database. No option to add signatures into the database is provided.

Inoculation and Scanning

As well as scanning, *NAV for NetWare* offers a second form of protection, which it calls 'inoculation'. This terminology is something of a misnomer, as the protected files (which

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.

must be executables) are not inoculated as such - inoculation usually refers to the process of adding code to a file which supposedly ‘protects’ it against infection, or which allows it to detect an infection. My guess is that in this case it means building a database of files, their locations and such details as size, dates, permissions and a checksum.

The inoculation feature could be a useful addition, as executables on a file server are extremely unlikely to change under normal circumstances. Also, the checking is being done on the server from an NLM, so it is not possible for a stealth virus to defeat the checking process (as happens with simple schemes on workstations). *Symantec* would be well advised to expand upon exactly what this feature does.

Scanning can be configured not only to check for signatures, but also to check for files which either are not in the inoculation database, or have changed their inoculation data. On detecting an inoculation variation, it is possible to configure options similar to those for virus detection.

The package also contains what is perhaps the most idiotic feature I have ever seen - automatic inclusion of the new file in the inoculation database, thus ensuring that any unknown file (which may be infected) is immediately marked as OK. I see very little justification for such a feature. If it is deemed an absolute necessity, it should be a ‘Hold the button whilst I do it’ option, not a tick box which can be set for ever. If you buy a guard dog, you must expect it to bark, and not muzzle it at the first noise.

Updates to virus signatures are available quarterly on floppy disk by subscribing to the update disk service. The manual mentions neither the cost of this service, nor whether or not the updates are available electronically.

Results were obtained by setting up the real-time scanner and copying the test-sets to the file server. When tested in the ‘all files’ mode, detection of the polymorphic test-set was appalling, finding less than 1% of all samples. However, once the files were renamed to their executable extension, detection results improved.

Conclusions

I experienced two problems with the software: every time I ran it with *Windows* in standard mode, I experienced a General Protect fault in *Windows*. This did not happen in enhanced mode. Additionally, the file server ‘fell over’ with a protection error twice when the NLM was loaded - this was far more worrying. Whilst I cannot pin it down to the NLM, this has never happened to me before with that particular file server, or with any other product.

This product has a simply brilliant user interface and features set. Unfortunately, although its virus detection results are not disastrous, they are hardly awe-inspiring. It seems to me rather odd that virus detection products tend either to be brilliant in detection and appalling in user interface design, or a joy to use but not particularly good at detecting viruses.

Norton Anti-Virus for NetWare

Detection Results (Secure mode):

NLM Scanner

Standard Test-Set [1]	220/229	96.1%
In the Wild Test-Set [2]	104/109	95.4%
Polymorphic Test-Set [3]	350/450	77.8%

DOS Scanner

No workstation software is provided.

Scanning Speed:

Speed results for an NLM product are inappropriate, due to the multi-tasking nature of the operating system. Full comparative speed results and overheads for all current NLMs will be printed in a forthcoming *VB* review.

Technical Details

Product: *Norton Anti-Virus for NetWare*

Developer: *Symantec Corporation (Peter Norton Group)*, 2500 Broadway, Suite 200, Santa Monica, California 90404, USA.
Tel. +1 503 334 6054, Fax +1 503 334 7400

UK Office: *Symantec Northern Europe*, Sygnus Court, Market Street, Maidenhead, Berkshire, UK.
Tel. +44 628 592222, Fax +44 628 592393

Price: £729.50, US$1093.50 for a single server with unlimited workstations, with monthly updates.

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

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

[1] **Standard Test-Set:** As printed in *VB*, February 1994, p.23 (file infectors only).

[2] **In the Wild Test-Set:** 4K (Frodo.Frodo.A), Barrotes.1310.A, BFD-451, Butterfly, Captain_Trips, Cascade.1701, Cascade.1704, CMOS1-T1, CMOS1-T2, Coffeeshop, Dark_Avenger.1800.A, Dark_Avenger.2100.DI.A, Dark_Avenger.Father, Datalock.920.A, Dir-II.A, DOSHunter, Eddie-2.A, Fax_Free.Topo, Fichv.2.1, Flip.2153.E, Green_Caterpillar.1575.A, Halloechen.A, Helloween.1376, Hidenowt, HLLC.Even_Beeper.A, Jerusalem.1808.Standard, Jerusalem.Anticad, Jerusalem.PcVrsDs, Jerusalem.Zerotime.Australian.A, Keypress.1232.A, Liberty.2857.D, Maltese_Amoeba, Necros, No_Frills.843, No_Frills.Dudley, Nomenklatura, Nothing, Nov_17th.855.A, Npox.963.A, Old_Yankee.1, Old_Yankee.2, Pitch, Piter.A, Power_Pump.1, Revenge, Screaming_Fist.II.696, Satanbug, SBC, Sibel_Sheep, Spanish_Telecom, Spanz, Starship, SVC.3103.A, Syslock.Macho, Tequila, Todor, Tremor (5), Vacsina.Penza.700, Vacsina.TP.5.A, Vienna.627.A, Vienna.648.A, Vienna.W-13.534.A, Vienna.W-13.507.B, Virdem.1336.English, Warrior, Whale, XPEH.4928

[3] **Polymorphic Test-Set:** The test-set consists of 450 genuine samples of: Coffeeshop (375), Cruncher (25), Uruguay.4 (50).

REVIEW

Virus: Prevention, Detection, Recovery

Following a spate of 'video nasties' from a number of different anti-virus software vendors, it was nice to have a vendor-independent choice on offer. The latest video is produced and distributed by *Commonwealth Films Inc*, and claims to be 'The virus awareness and protection video for the 1990's.'

After a lurid yellow banner informing the viewer that he is being treated to a video recorded in 'MacroVision' (whatever this may be), the video kicks off in a busy office, where Sarah, the secretary, has discovered her computer has a virus. Within minutes, the office grinds to a halt as she gathers an array of puzzled onlookers around her. Naturally, Sarah makes several elementary mistakes - the rest of the video goes on to explain what she should have done.

Target Audience

Clearly *Virus: Prevention, Detection, Recovery* is aimed at the average PC LAN user, and as such avoids many of the technicalities and problems associated with the subject. This simplification is generally a good thing, however it is important that the process is not taken too far. One piece of misinformation which the producers should never have allowed to be included was a throwaway remark about viruses 'jumping off' infected floppy disks. This is nonsense, and although it fits the simplistic tone of the other explanations, it is a myth which raises its head time and time again. A virus cannot do this: it must be executed, to spread.

> *"Particular emphasis is placed on four 'things to do' when discovering a virus"*

This criticism aside, the remainder of the video is enjoyable to watch. Viewers are treated to a meeting with the fictitious 'Leo', the author of the 'Zoo' virus. Under the guise of Leo's bragging about his latest creation, Zoo.2, the important facts about how viruses could get into the office are outlined, ranging from old favourites like 'the PC at home' right down to shrink-wrapped software.

Leo's character makes the video much more watchable, and saves the user from the interminable droning of an industry 'guru' - while MIS managers need much more hard technical information, the average user is probably better off with an entertainingly-presented checklist. A little bit more emphasis on facts here would not go amiss, but overall this section is good.

As the video is designed as a 'one size fits all' offering for a large number of different policies and companies, the pros and cons of different detection strategies are not touched upon. Rather, a feel is conveyed for the ways in which users can help solve the problem. Particular emphasis is placed on four 'things to do' when discovering a virus, namely:

- Do not spread panic
- Stop using the affected PC
- Get expert help
- Write down anything that appears on the screen

Good advice, and a procedure which anyone involved in PC technical support would welcome.

Cheap and Cheerful?

Perhaps the most memorable part of the video is its price. Training videos have seldom been cheap, and given that they can be used many times, spending a large amount of money on a video which actually works can often be a highly cost-effective solution to the problem. However, *Virus: Prevention, Detection, Recovery* is priced at £675, which works out at over £30 per minute.

Were it the only offering on the market, this price tag could be justified by a supply and demand argument. However, with equally watchable offerings featuring such industry luminaries as Wilf Hey (of *PC Plus* fame), Alan Solomon or Jan Hruska, it is extremely difficult to see why *Commonwealth's* offering costs so much.

On the plus side, the video is well produced, and does seem to emphasize the right ideas. Viewers are not swamped by vast amounts of technical information, and the short length of the film makes it ideal training material. In summary, *Virus: Prevention, Detection, Recovery* is a well-focused production, but has a large question mark hanging above its head in terms of value for money.

Video: *Virus: Prevention, Detection, Recovery.*

Format: VHS. Other formats available at additional charge.

Running time: 22 minutes.

Price: £675 (VAT free). 2 copies 10% discount. 3 to 5 copies 15% discount.

UK Vendor: *Commonwealth Films*, 6a Old Dunleary Road, Dun Laoghair, Co. Dublin, Ireland. Tel. +353 1 280 0506 (UK only 0800 387 458), Fax +353 1 284 2657.

Europe: *Commonwealth Films*, Stephanie Square, Avenue Louise 65, bte. 11, 1050 Brussels, Belgium. Tel. +32 2 535 7888 Fax +32 2 535 7700.

USA: *Commonwealth Films Inc*, 223 Commonwealth Avenue, Boston, MA 02116, USA.
Tel. +1 (617) 262 5634, Fax +1 (617) 262 6948.

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:

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
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
Dr. Tony Pitt, Digital Equipment Corporation, UK
Yisrael Radai, Hebrew University of Jerusalem, Israel
Roger Riordan, Cybec Pty, Australia
Martin Samociuk, Network Security Management, UK
Eli Shapira, Central Point Software Inc, USA
John Sherwood, Sherwood Associates, UK
Prof. Eugene Spafford, Purdue University, USA
Dr. Peter Tippett, Symantec Corporation, USA
Dr. Steve R. White, IBM Research, USA
Joseph Wells, Symantec Corporation, USA
Dr. Ken Wong, PA Consulting Group, UK
Ken van Wyk, CERT, USA

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire,
OX14 3YS, England

Tel. 0235 555139, International Tel. +44 235 555139
Fax 0235 559935, International Fax +44 235 559935
Email virusbtn@vax.ox.ac.uk
CompuServe 100070,1340@compuserve.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

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.

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

Command Software has just announced its latest addition to its customer list: *Microsoft*. Rather than adopting the company's own anti-virus software *MSAV*, the software giant has standardised on *NET-Prot* and *F-Prot Professional* to protect its PCs [*Is it possible that Microsoft has discovered a hidden security hole in MSAV? Users should be told! Ed.*]

China is facing losing its 'most favoured nation' trading status with the USA unless it cracks down on **piracy**, according to the newspaper *China Daily* on June 13. CD production in China presently stands at circa 100 million per year, including pirated disks selling at 10% of the original price.

The **call for papers** of the *1994 EICAR Conference* has been announced. The conference will concentrate on methods of improving small system security, and will be held jointly by *BP Oil* and *S&S International*. The delegate fee for the two-day conference, to be held in Hemel Hemstead, England, is £595+VAT. Tel. 0296 318700.

A new Swedish organisation aimed at developing a more positive image for BBSs has been set up. The group has proposed ethical guidelines for BBS operators, banning software piracy, computer viruses and other malicious software, and child pornography.

Cybec Pty Ltd have announced a **new boot sector virus in Australia** called Mongolian (due to the fact that it originated in that country). The virus overwrites the first 17 sectors of each partition on the hard disk, and then the Master Boot Record, if switched on on 30 May.

S&S International has once again announced a 'Trade-Up Programme', whereby users of competing products can change to *Dr Solomon's AVTK* at a 'significantly reduced price'. Users can now call a Freephone number to receive free advice on viruses. Full details on 0800 136657 (UK only). Outside UK, Tel. +44 (0)296 318700.

The *Computer Security Institute* (*CSI*) has released the 1994 edition of its official *Computer Security Products Buyers Guide*. It now offers a 'fax on demand' service from 1 June through 31 August 1994. Further information available from the *CSI.* Tel. +1 415 905 2626.

The UK *NCC* (*National Computing Centre*) is holding management seminars on IT security on 13 July in Birmingham, 14 July in Manchester, 19 July in London, and 21 July in Glasgow. Details from Jayne Howell. Tel. +44 (0)61 228 6333 Fax +44 (0)61 237 5330.

Microsoft* has announced the launch of *MS-DOS v6.22, which does not include the controversial disk compression software (called *Double Space*) integral in v6.0 and v6.2. *Microsoft's* latest version of *MS-DOS* contains a new disk compression system called *DriveSpace*, which does not infringe *Stac's* patents.

Racal Electronics has acquired *Airtech Computer Security Ltd*: the new group, *Racal Airtech*, will be headquartered in Oxford, England and will concentrate on the areas of system security, access control, terminal security, and link encryption.

Kevin Poulsen, known to the computer underground as 'Dark Dante', has pleaded guilty in a US federal court to charges of computer fraud, interception of wire communications, mail fraud, money laundering and obstruction of justice. It is alleged that he won two Porsches, two trips to Hawaii, and US$2000 under false pretences. He has also admitted accessing computers to identify undercover businesses used by the *FBI*, to locate *FBI* wiretaps and to eavesdrop on private citizens. Poulsen faces up to 40 years in prison and a US$1.7 million fine.

Sophos is holding a **Computer Virus Workshop** at the *Sophos* training suite in Abingdon, near Oxford, on 27/28 July. Cost for one day is £295.00 + VAT, and for both days £545.00 + VAT. For further information, contact Karen Richardson. Tel. +44 (0)235 559933.