ISSN 0956-9979 APRIL 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

IN THIS ISSUE:

• **Linking Up.** Those subscribers who are not yet familiar with Link viruses will find this month's tutorial a useful source of information.

• ***VB* Survey.** Readers from far and wide responded to the questionnaire - the findings are on pp.14-15.

• **Artificial Life.** The latest theory presented by certain members of the computer underground is that viruses are a life form, albeit artificial. Mark Ludwig's new book on the subject claims to be *The Little Black Book of Computer Viruses II*. Does it deserve the same criticisms as his first book on the subject? See page 23.

CONTENTS

EDITORIAL

NEWS

INSIGHT

VIRUS ANALYSES

TUTORIAL

FEATURE

PRODUCT PREVIEW

PRODUCT REVIEWS

BOOK REVIEW

VIRUS BULLETIN ©1994 **Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)235 555139. /90/$0.00+2.50**

EDITORIAL

Room for Improvement?

One of the (many) tasks given to the *VB* Editor is to write new papers for conferences and presentations. Often such talks are based upon familiar but important themes, such as 'The Role of User Awareness' or 'Developing an Anti-virus Policy which Works'. However, as often as possible, the subject matter is new, and attention can be turned to more technical issues. In the case of the latest talk given (at the Washington-based *NCSA* conference), the subject under discussion was the infection of 'unusual objects'.

The idea that COM, EXE and BAT files are not the only programs on a computer which can be infected by a virus will not be a new one to regular readers of *Virus Bulletin*. Any object which either is executable, or under certain circumstances can become executable, or which represents a pointer to executable code, is a potential attack point. Some of these issues were discussed in last July's *VB*, and the subject raised its head again last month, in the form of an OBJ-infecting virus.

> *“ those involved in generic virus detection face a very different problem: they must be proactive in their approach ”*

It seems that the issue of unusual infection targets has yet to be addressed by those vendors who sell products which provide 'permanent protection against known and unknown viruses'. To those who do not believe that any vendor would make such a tall claim, this quote was taken from one of the products in the *Virus Bulletin* product library.

The number of objects on the PC which can be infected grows with every new enhancement to its operating system. *Windows'* screen saver files, DLLs, OBJ files… monitoring all possible infection targets on a machine is rather like cutting the heads off a Hydra. Keeping up with new viruses as they are written is hard enough - keeping up with new *ideas* is much more difficult.

These problems are of far greater import to those vendors who claim to provide a 'past, present and future' solution. The way in which an anti-virus scanner manufacturer operates is purely reactive: a new virus is found, and the product is altered in order to take this into account. However, those involved in generic virus detection face a very different problem: they must be proactive in their approach. Loopholes in DOS must be plugged before a virus is written which takes advantage of them; one cannot simply sit back and wait for the computer underground to act.

To illustrate this point, let us consider the humble checksummer. How many checksummers check every potential executable on the system by default? This will include DLLs, WIN.INI, WIN.COM… the list is sufficiently long that few (if any) products provide complete protection. Therefore the user is purchasing protection from a *particular type* of virus, not future-proof, all-round detection. This may be exactly what the user wants, but is often not what he thinks he has bought.

Approaches which do not rely on virus-specific information are of increasing interest to users as the number of individual viruses continues to climb. Generic virus detection is a powerful additionto the industry standard technique of 'scan and forget', but at this time, the large vendors seem uninterested in pouring time and money into further development. There are a number of possible reasons for this - however, the largest stumbling block is not a technological problem, but a financial one.

Clearly, if a vendor feels that further development will not improve sales, then such enhancements will be shelved until they become a priority. Users are not aware of the issues raised by the infection of previously unconsidered objects, and most policies are not centred around generic protection. If development does not pay, it will not be done - it would be naïve to think otherwise.

Improving the checksummers supplied with many products would be a comparatively simple exercise, certainly when compared to the contortions necessary to detect some of the new highly polymorphic viruses. However, few checksummers are designed to run in such a way as to examine the *structure* of files on the disk: does the file have an executable form? The fact that so many vendors are blasé about such problems is an issue which users should raise. Anti-virus software is costly, and many users pay premium prices for 'the best'. It is time they demanded it.

NEWS

The '2nd International Virus Writing Competition'

According to an announcement in the *Computer underground Digest* (*CuD*), Mark Ludwig has launched the '2nd International Virus Writing Competition'. The article takes the form of a four-page entry form and an introduction which requests programmers to 'write a virus which is itself a political satire'.

The text of the introduction explains that the contest is sponsored by *American Eagle Publications Inc*., and *The Crypt Infosystems BBS*. Prizes for the winning entry include US$100 cash and a year's subscription to Ludwig's virus magazine, *Computer Virus Developments Quarterly*.

As an example of how the winning entry might function, *American Eagle* gives the following example:

```
The PCV
This virus is a memory-resident boot sector
virus which maintains a list of politically
incorrect words on your computer system. It
also hooks the keyboard interrupt and monitors
every keystroke you make. If you type a
politically incorrect word into the computer,
the PCV springs into action... The virus also
uses powerful means to prevent disinfection,
so that, once you get it, you can't get rid of
it without a major effort.
```

Such competitions play directly into the hands of those who wish to strengthen American legislation on the subject of computer viruses. Although the virus authors may claim that their work should never be released 'in the wild', it is all too easy for this to occur, especially in the case of a supposedly 'amusing' virus ▮

Macintosh Developments

Another *Apple Macintosh* virus has been discovered 'in the wild' in Italy. The virus, named INIT-9403 (alias SysX), is believed to have been distributed on an altered version of pirated commercial software. When executed, it installs the virus on the affected system.

The virus is thought to be widely spread on systems running the Italian version of *MacOS*. It infects the Finder file, and may insert copies of itself into various file compression and archiving utilities.

INIT-9403 contains a malicious trigger routine: after a certain number of files have been infected, it will attempt to erase the contents of all hard drives which are connected to the system. All the major *Macintosh* anti-virus software vendors are planning to release updates to their products, which will be available through the usual channels, in order to detect/eliminate the virus ▮

Virus Prevalence Table - February 1994

Virus	Incidents	(%) Reports
Form	15	32.6%
New Zealand 2	5	10.9%
Parity Boot.A	4	8.7%
Spanish Telecom	3	6.5%
Amse	2	4.4%
Disk Killer	2	4.4%
Exebug.4	2	4.4%
Form.B	2	4.4%
NoInt	2	4.4%
Anti-CMOS	1	2.2%
JackRipper	1	2.2%
Black Monday	1	2.2%
Form.II	1	2.2%
Keypress	1	2.2%
PrintScreen	1	2.2%
Joshi	1	2.2%
Tequila	1	2.2%
Stoned.O	1	2.2%
Total	46	100.0%

Acorn Problem Grows

Although the virus problem on the *Acorn Archimedes* is much smaller than that on the PC, the number of *Archimedes* viruses continues to rise. The latest new virus, Dratsab, brings the total to 56, and marks a new trend in viruses previously observed on this platform.

The text within the virus claims that Dratsab is a 'mutating' virus (i.e. is polymorphic). Compared to the complexity of the PC polymorphic engines such as the Mutation Engine or TPE, this boast has little meaning. However, the virus does, to a limited extent, vary its appearance from one infection to another. This is achieved by the technique of altering its overall length by including between one and a hundred calls to a particular procedure. It also chooses a random filename and filetype in which to store its code. Due to its simple-minded approach, it presents no great problem to vendors.

Dratsab was discovered 'in the wild' but the extent of its distribution is at present unknown. Anti-virus researchers hope to be able to prevent it from becoming widespread.

Commenting on the virus, Alan Glover, author of the *Archimedes* virus scanner *Killer*, said, 'The *Archimedes* scene is rather like the early days of the *IBM* virus problem. As time goes on, it seems likely that more ideas will be transferred from one machine to the other.' ▮

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 17 March 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

Abraxas.1214

EN: An overwriting virus which has practically no chances of spreading.

```
Abraxas.1214    CD21 B43C 33C9 BA9E 00CD 21B7 4093 BA00 01B9 BE04 CD21 C3B4
```

Appelscha

CER: A Dutch 2161-byte polymorphic virus. No simple search pattern is possible.

Baron

CR: A 255-byte virus recently reported ‘in the wild’ in the UK. It does nothing but replicate, and contains the following messages: ‘GERM. (C) The Black Baron U.K 93’, and ‘Better SMEG than dead’.

```
Baron           1E50 5352 B802 3DCD 210E 1F93 B800 57CD 2151 52BA FA01 B905
```

Cascade.1701.P

CR: Detected with the Cascade-YAP pattern.

Civil_War.281

CN: A small, unremarkable 281-byte variant.

```
Civil_War.281   E800 005D 81ED 0901 BA00 FEB4 1ACD 21BF 0001 8DB6 EF01 B906
```

Dark_Avenger.1799

CER: This variant is also known as Francis, because the text at the beginning has been replaced with the message: ‘Francis lives…in Hong Kong’. Apart from the fact that it is one byte shorter than the standard 1800 variant, the code is practically identical. Detected with the Dark_Avenger pattern.

Dark_Avenger.1800.Platina

CER: A minor variant, 1800 bytes long. Most of the differences are in the text strings, which have been changed to ‘THE LITTLE BEETLE - PLATINA BOYS’ and ‘It’s written in Hradec Kralove, Czechoslovakia (C)1990 [Fuck,fuck,fuck]’. Detected with the Dark_Avenger search string.

Datalock.828.B

CER: As the 828.A variant reported last October, detected with the Datalock pattern, which also detects Datalock.1150. The third new variant is 1740 bytes long, and requires a new search string.

```
Datalock.1740   C31E 8CD8 488E D881 2E03 008C 0040 8ED8 A102 002D 8C00 A302
```

Dracula

ER: Awaiting analysis. 827 bytes long.

```
Dracula         FB50 5351 5256 5755 1E06 9C3D 004B 7408 80FC 3D74 03E9 F101
```

Gotcha.605

CR: Detected with the Gotcha-E pattern. It appears to be based on the same source code, as it includes a number of search strings from other viruses. These seem intended to fool certain scanners, in particular *McAfee’s SCAN*. However, that misidentification problem was fixed some time ago.

Grog.1089

CN: This 1089-byte virus uses polymorphic encryption, making extraction of a simple search string impossible. It contains the text ‘JoeLEsquimese (C) '93 by GROG - Italy’.

Grog.1200

CR: Another polymorphic virus, probably by the same author. It contains the text ‘GROG v3.1 (C) '93 by GROG - Italy’.

Helloween

CER: Four new variants (1228, 1401, 1430 and 1684 bytes) are now known. The first three are detected with the Helloween pattern, but the 1684-byte one requires a new search string.

```
Helloween.1684  B43F EB03 90B4 3EE8 1600 7202 2BC1 C333 C933 D2B8 0241 EB08
```

Intruder.1555

EN: Detected with the Intruder pattern.

Jerusalem.Tarapa

CER: A 2064/2069-byte variant, detected with the Jeru-1735 pattern.

Little_Red

CER: 1465 bytes long. There have been isolated reports of this virus in the wild.

```
Little_Red      3D00 4B74 1D80 FC30 740F 80FC 1175 03E9 07FF 80FC 1274 F8EB
```

Metallica.2620

CR: This is clearly related to the Metallica.1739 virus, but possibly also to the Emmie family - further analysis is required to determine the exact relationship.

```
Metallica.2620  86E0 3C3D 7432 3D6C 0074 183C 4B74 353C 4374 253C 5674 2186
```

Murphy **CER:** Two new variants: Murphy.1477 and Murphy.1521.B. Both are detected with the HIV pattern.

Mystic **CN:** Two closely-related viruses, 377 and 379 bytes long, containing the text ‘'Mystic' by Digital Alchemy’. Although encryption resembles that of a VCL-constructed virus, decrypted code is different.

```
Mystic.377      B9AE 0081 37?? ??83 C302 E2F7 C3
Mystic.379      B9AF 0081 37?? ??83 C302 E2F7 C3
```

Particle_Man **CN:** Much of the body of this 690-byte virus is taken up with a long text message, starting with the words ‘Particle man, particle man Doing the things a particle can....’.

```
Particle_Man    518B B63F 018D BE5F 01B9 4201 3135 83C7 02E2 F959 C3E8 E8FF
```

PCBB.1683 **CR:** No single simple search pattern is possible for this virus, although it is detectable with a small set of patterns containing wildcards.

Pixel.251 **CN:** A small variant which does nothing but replicate.

```
Pixel.251       BF00 01F3 A42E C706 0001 0001 2E8C 1E02 0153 582E FF2E 0001
```

Pixel.761 **CN:** Detected with the Pixel-936 pattern. This variant contains the text ‘LiquidCode<tm>’.

Predator.1154 **CR:** Similar to Predator.1137, but slightly longer. It is encrypted, and contains the text: ‘Predator virus (c) Mar. 93 In memory of all those who were killed...Wookies ain't the only ones that drop! Priest’.

```
Predator.1154   BA35 02B1 ??FA 8BEC BC?? ??58 F7D0 D3C8 50EB 01?? 4C4C 4A75
```

PS-MPC As expected, there are several new PS-MPC-generated viruses this month. No search patterns will be published for these viruses, as most are encrypted, but any good virus scanner should be able to detect them. The list this month includes: 150 (CN), 425 (CR), 569 (CER), 594 (CER), 639 (EN), 691 (CEN), 739 (CER), 749 (CEN), 2668 (CEN), Abominog (2011, CN), Actifed (725,CER), Alchemy (700, CEN), Argent (762, EN), Birthday (1104, EN), Blender (578, CEN), Doggy (538, CEN), Fred (720, CEN), G2.572 (CEN), G2.573.A (CEN), G2.573.B (CEN), G2.574 (CEN), G2.575.A (CEN), G2.575.B (CEN), G2.576 (CEN), G2.582 (CEN), G2.584.A (CEN), G2.584.B (CEN), G2.584.C (CEN), G2.585.A (CEN), G2.585.B (CEN), G2.588 (CEN), Joana.942 (CEN), Justice (1151, EN), McWhale.1023 (EN), McWhale.1124 (EN), Mojave (626, CEN), Projekt (918, CEN), Ranger (44, CN), School (473, CN), Shock (401, CN), Skeleton.542 (CEN), Sorlec.597 (CR), Steeve.672 (EN), Steeve.686 (EN), Swansong.1719 (EN), Swansong.1772 (CEN), Swansong.1773 (CEN), Swansong.2062 (CN), Walt.311 (CN), Walt.355 (CN) and Warez.1805 (CEN).

Timid **CN:** Three new variants have appeared (298, 299, and 301 bytes long), and are detected with the Timid (originally named Timid-305) pattern.

Tolbuhin.1004.B **CN:** Detetcted with the Tolbuhin (previously SK) pattern.

VCL **CN:** Five new VCL-generated viruses have been reported. As in the case of the PS-MPC viruses, search patterns will not be published for the encrypted viruses. They are: Angel (436), Dial (599), Julian (2737), Muu (610) and Suck (677). In addition, the unencrypted variant VCL.Assassin (756) is detected with the VCL.VoCo and VCL-non pattern.

VCS **CN:** Four new variants are now known: VCS.Standard.Darkside, VCS.Standard.Parity VCS.Standard.Test and VCS.Standard.VDV. All are detected with the VCS 1.0 search pattern.

Vienna.608, Vienna.610 **CN:** Two similar variants, detected with the Vienna-4 and Dr. Q patterns.

Vienna.700.A **CN:** This is really a variant of the 648-byte Vienna.Lisbon virus, and just like that virus, sometimes overwrites the beginning of COM files with the word @AIDS. Detected with the GhostBalls and Vienna-1239 patterns.

Vienna.814 **CN:** Due to an error in the code, almost all infected files will not work properly. Detected with the Vienna-4 and Dr. Q patterns.

Vienna.Violator.803 **CN:** An unremarkable Vienna-variant, 803 bytes long. Four other members of the Violator group have been reported recently, and can be found using previously published search patterns. They are: Vienna.Violator.909 (detected by the Vengeance search string), Vienna.Violator.957 (detected by Infinity), Vienna.Violator.801 (by Violator.C) and Vienna.Violator.5286 (by Xmas Violator).

```
Violator.803    ACB9 0080 F2AE B904 00AC AE75 EEE2 FA5E 0789 7C4A 8BFE 83C7
```

Vienna.W-13.507.E **CN:** Minor variant, detected with the W13 pattern.

Warsaw **CN:** An 850-byte Polish virus, which contains the text ‘FBC Warsaw - virus 1990’.

```
Warsaw          7305 8CC0 408E C08B FB33 C926 8A25 80FC 2E74 0A47 4183 F90C
```

Yankee-Doodle.Login.3045.C CER: Minor variant, detected with the Yankee-login string.

INSIGHT

Kaspersky: East goes West

Megan Palfrey

Eugene Kaspersky is one of the best-known Russian anti-virus specialists, both in the East and the West. This position has taken a relatively short time to attain: his first job in computers began only seven years ago, as a young programmer in a State company, and led to his discovery of the world of computer viruses.

At the Outset

The first company at which Kaspersky worked had several computers, amongst them an *ES-1033* and *-1060* (*IBM 360/370* clones), an *SM-4* (*PDP-11* clone), and one *IBM PC/XT* - an *Olivetti M24*. This last machine was not specifically assigned to anyone, but Kaspersky was placed in the department where it was located, and soon became the resident PC expert. He was able to demonstrate that many applications could be transferred from mainframes to PCs, with the result that his superiors chose to standardise on PCs within the entire company.

Along with PCs came the problem of PC viruses. The first to appear in Russia were Vienna-648 and Cascade, followed by Vacsina, Yankee Doodle, and Jerusalem. Kaspersky's love of experimentation helped him disassemble Vienna-648, after which he wrote a rudimentary virus scanner, *-V.EXE*. These were modest beginnings; *-V.EXE* could detect only the Vienna virus! However, he enjoyed the subject, and resolved to specialise in developing anti-virus software.

-V.EXE was Kaspersky's 'baby' for two years, despite the fact that he did not market it at that time (autumn 1988), distributing it only to friends. This early version was a virus scanner only, capable of detecting just two viruses. Within a relatively short period of time, this number grew, as new viruses were written. Unsurprisingly, Kaspersky soon became known as *the* PC computer virus expert. By then, (late 1989/early 1990), *-V.EXE* was a freeware package which included on-line help, an anti-virus monitor, and memory-browsing utilities. Kaspersky feels an obligation towards those early users: 'Responsibility for end-users and my interest for viruses - these are the points which lead me.'

He is, however, not averse to the 'perks' which come from being the owner of such knowledge: 'It was quite pleasant to become well-known, and even more so finally to start making money from my product!' he explained.

The Middle Years

Kaspersky defines as his 'middle period' the time from 1990 to the present - since 1990, his product has changed markedly, and his career has progressed proportionately. He finally produced *-V.EXE* commercially, in 1990/91, launching it as *Dr Kaspersky*. In 1993, this led to the production of v1.0 of the *Antiviral Toolkit Pro* (*AVP*), with a database editor, new versions of the monitor and utilities, and hypertext help. Version 2.0 will soon be available, and will purportedly be able to scan compressed files and archives.

During this time, he also started working for *KAMI*, a computer trade company. He had known the president of the company, Alexey Remizov, for some time, having met him two or three years before the company was born. Then (1983/84), Remizov was a young mathematics teacher, and Kaspersky, a student. They lost touch, but met up again at a conference ('I don't remember which one,' admitted Kaspersky sheepishly). Remizov told him about all the new developments at *KAMI*, and it was not long before he was persuaded to work for them.

I asked Kaspersky exactly what he does at *KAMI*: 'My role? It's a difficult question. *KAMI* is not a typical company. Really, I'm another person in a company of friends. Also, I am a well-known programmer in Russia, which is advantageous to the company - my bosses never forget to mention to clients that I work there. So, I am a "face" for *KAMI*.'

Kaspersky's main function, however, is not as a goodwill ambassador; rather, he is an expert in the development of anti-virus software. His laboratory has expanded and improved over the years: he started with a 286, then upgraded to a 386. Now his team has three 486s, two 386s, two test computers, hardware, modems - and the list goes on. 'I see that my laboratory is quite good,' said Kaspersky.

KAMI does not specialise in anti-virus equipment; it sells computers, and other related hardware. This, according to Kaspersky, is the company's bread and butter: 'They work with soft- and hardware development too, but that brings less revenue than trade in computers.'

The Changing Face of Moscow

There have been incredible changes in Russia recently, which have already influenced Kaspersky's life.

'Life is much easier. There are no problems with such day-to-day matters as buying petrol, food, or clothes. I remember great queues at petrol stations in the days before Glasnost: now I wait perhaps two minutes each time I want to fill the car up. Moscow looks like a western city now - shops, cars, shops, shops, cars… It has not yet attained the standards of a western city, but it's moving that way. Life is not difficult, if you have money - but this is an international problem!'

However, although much has eased, it is still not all 'plain sailing'. He would very much like to develop his research and his products outside his own country, but finds it practically impossible to publish software in the West. He has started to distribute his software as shareware by ftp sites, BBSs, and mailservers - but, he feels, this is not enough. ‘The best solution would be to find a software company which will buy my product and sell it, or insert my engines into their own anti-virus software,’ he explained.

The Virus Writers

Kaspersky is not worried about virus writing becoming overly prevalent in Russia: ‘There are only about thirty new viruses per month. Sometimes, though, a virus author will give us a real present - I recently came across the Phantom_1 virus, which belongs in this category. It’s one of the most complex viruses I’ve ever seen.’ [*See pp.8-9. Ed.*]

> *“I think the way ahead lies in database-oriented scanners with local technical support sites”*

He believes that whoever wrote Phantom_1 was familiar with polymorphic engines such as the MtE, seeing technical similarities between them. He also feels that it will not be long before other similar viruses are released into the wild: ‘Even if we don’t see another virus from this person, there will be variants and hacks of Phantom_1. As long as *MS-DOS* exists, viruses will continue to be written, and will become steadily more clever, difficult, and ingenious.’ Even Phantom_1, however, which Kaspersky views as one of the best-written viruses he has seen, is not perfect: ‘This virus has bugs. Sometimes it infects files incorrectly. They consequently do not execute; therefore the virus will not replicate. The world’s “best” virus cannot have such bugs.’

The things which motivate a Russian virus writer are similar to those found elsewhere, with the added pressures of life in Russia: unemployment, dissatisfaction with the work within a State company, not enough money to live comfortably… ‘He is unhappy, therefore he becomes malicious.’

Trends in Russia mimic, to an extent, those in the West. Kaspersky cited Burger’s book on viruses and sources: ‘This resulted, in the West, in a lot of Burger-based viruses. Some time ago, a Russian called Khiznak also wrote such a book - the result, a lot of Khiznak-based viruses!’

Kaspersky is convinced that viruses are generally written by people who are bored, who can find nothing more constructive to do with their time. Eighty percent of viruses, he thinks, are written by those ‘natural hooligans’, teenagers, and probably only 5% by competent programmers. ‘Why write a virus? I think everyone has a criminal side. One man may have a soul which is 1% criminal, another, 99%! The programmer with a 1% criminal soul will never write a virus; he who has, say, a rating of 40% might write a virus but never distribute it. The programmer with a soul which is 99% criminal will write a virus, distribute it, and be happy at resultant damage. And there will always be such people.’

Russia has not yet seen the rapid growth of polymorphic engines and virus construction toolkits which are rampant in the West, despite the fact that such things are available on BBSs in his country. This is fortunate, as Kaspersky is aware of only four anti-virus scanners which are produced there: ‘Too few,’ he says sombrely, ‘for a country such as mine.’

Forging Onwards

Where does anti-virus research go from here? The latest version of Kaspersky’s *AVP* has a heuristic element, but he does not feel that heuristics are necessarily the route to take.

‘Heuristic scanners say “This file looks like a variant of virus AAA”, with *n*% success. Heuristic scanners are “first-alarm” software only. I think the way ahead lies in database-oriented scanners with local technical support sites.’

He explained his theory further: any user who discovers an infection of his computer sends an infected file/floppy to a local support site. Experts there would be notified, and the virus would be analysed, added to their database, and a ‘cure’ provided for the original user. Kaspersky uses these techniques in Russia and in Italy, where he has a number of such sites in operation.

Tongue in cheek, Kaspersky claimed that he was fed up with hearing about viruses: ‘Every day it’s viruses, viruses, viruses. Let’s stop here and now. Stop writing viruses, stop writing about viruses. Let’s... Give me time to relax!’ In a more serious vein, he is completely immersed in the field of anti-virus research: Kaspersky sees computer viruses as an out-of-the-ordinary theme, bringing him into contact with out-of-the-ordinary people and situations.

‘I can remember someone asking me if they themselves could become infected with a computer virus if they worked with computers!’ he chuckled. ‘But really, viruses are intriguing to the computer specialist, from a technical point of view. I simply find the subject passionately interesting.’

From Today to Tomorrow

Kaspersky sees himself as a ‘dyed in the wool’ researcher: ‘I will work with viruses as long as they are around. If *MS-DOS* dies next year, I will work with viruses for only one more year. I’m an 8086 man, and wouldn’t jump to another platform now; I’ve been here far too long.’ He wants to continue working with programming until the end of the century - then, he says, he will look for a change.

‘I’m a team leader now; I am acquiring experience in that area. I would like to be in charge of a project of some sort - if computer viruses are still an issue then, perhaps I could use my expertise there. But, maybe I will be a businessman! Maybe... I just don’t know yet what will happen.’

Wherever his future takes him, one thing is certain: there are enough viruses around to occupy him full-time for the present, and for the foreseeable future. Kaspersky plans to continue in anti-virus research for as long as this holds true.

VIRUS ANALYSIS 1

The Phantom Flies

Eugene Kaspersky

Spring 1994 brings an early Easter present for anti-virus software vendors: Phantom_1, a new polymorphic virus. The virus presents one of the biggest challenges to researchers to date, and may lay claim to the dubious honour of being the most polymorphic virus in the world!

Phantom_1 is known to be ‘in the wild’, and spread rapidly in Moscow at the beginning of March. Somebody (possibly the virus author) infected the latest version of the most popular Russian virus scanner (AIDSTEST), and uploaded it to many local BBSs. When it was downloaded and used, the file infected the user’s machine. The following day, there was a storm of phone calls to anti-virus technical support sites and their BBSs. Users asked for software updates, but were told that because the virus was very difficult to detect, it would take some time before detection and disinfection routines would be ready.

It is possible that this Trojanised AIDSTEST file is still available on some BBSs: whoever uploaded it masked the virus using PKLITE, making it even more difficult to find. AIDSTEST.EXE checks the integrity of its own host file before processing other functions - the Trojanised copy was patched so that it does not display any warning message.

Phantom_1 is a fast infector, hitting files on execution or opening. Like other Russian viruses (for example, Penza or SVC), it could easily become prevalent worldwide.

Installation and Infection Routines

The virus is a memory-resident parasitic COM and EXE infector, 7000 bytes long. On execution of an infected file, processing immediately passes to the start of the decryption routine. In COM files, this is achieved by a simple JMP instruction to the virus code; in EXE files, the entry point is altered to point to the decryption routine.

Once the main body of the virus is decrypted, control passes to the virus’ installation routine. Phantom_1 first checks whether a copy is already memory-resident by means of an ‘Are you there?’ call. This consists of calling Int 21h with the value ABCDh loaded in the AX register. If the call is returned with FFFFh in the same register, Phantom_1 assumes that a copy is already resident. In this case, the virus repairs the memory image of the host file, and passes control to it.

If the call goes unanswered, processing passes to the installation routine. The method employed is reminiscent of that used by Jerusalem: Phantom_1 copies itself to the beginning of the memory block allocated to the infected program and executes the host file, staying memory-resident by taking advantage of the Terminate_and_Stay_Resident function (Int 21h, AH=31h). The virus does this in a slightly more elegant manner than Jerusalem, using code which is better optimised.

Once installed in memory, the virus tunnels the Int 21h and Int 13h addresses, and obtains direct access to the true DOS interrupt handlers. The virus then hooks Int 1Ch for the trigger routine, and Int 21h, for file infection.

Whenever the DOS calls Load_and_Execute (AX=4B00h) or Open_File (AH=3Dh) are intercepted, Phantom calls its infection routine. This first checks the filename, and specifically excepts the files AIDSTEST.EXE and SCAN.EXE from infection. This test complete, the virus checks the target file’s extension, and ensures that it is either EXE or COM. Thereafter, the infection routine begins in earnest.

> *“The decryption loop consists of …randomly selected instructions such as XOR, ADD, SUB, DEC, NOT, ROR and ROL”*

Phantom_1 first hooks several different interrupts: Int 24h (the Critical Error Handler), Int 01h and Int 03h (two interrupts used by debuggers) and Int 2Ah. The last three of these routines are all replaced with a simple IRET instruction. In the case of Int 2Ah, this disables a certain memory-resident anti-virus program.

The next action taken is a check of the amount of free disk space available - Phantom_1 is large, and if several executable files are infected on a single floppy disk, it is possible that an ‘out of disk space’ error will be generated. File time and date stamps are stored for later use, and file attributes are reset and restored after infection has completed.

In order to prevent multiply infecting files, Phantom_1 makes two checks on target files. Firstly, if the file extension is COM, the first byte is checked for the value E9h (JMP). Secondly, if the file has an EXE structure, the SP register field of the header is checked for the value 1000h. If either of these conditions is satisfied, the infection routine aborts.

The last precaution taken by the virus writer is to check the target file’s length: if this is shorter than 4096 bytes, infection does not take place. Similarly, if a COM file is longer than 58368 (E400h) bytes long, it is deemed unsuitable. In the case of EXE files, Phantom_1 compares the real file length with the module length (calculated from the EXE header), and does not infect if these values are different (this would indicate the presence of an overlay file).

Finally, control is passed to the polymorphic routine, and the decryptor and the encrypted virus code are written to the end of the host file.

As Polymorphic as they come

There are many viruses classified as ‘polymorphic’, i.e. viruses which encrypt themselves and save different decryption routines (such as MtE-, TPE-, NED-based viruses, Phoenix, Tremor, Uruguay and so on) or hide themselves in the middle of a file with different entry code sequences (like Commander Bomber). Some of these viruses use polymorphic engines capable of producing very complex code, using many different instruction types (e.g. Uruguay), and others are comparatively simple (for example, Phoenix). Thus, different viruses have different degrees of polymorphism; Phantom_1 produces some of the most variable code I have seen to date.

The polymorphic engine within Phantom_1 is approximately 3K in length, and highly complex, divided into approximately 40 subroutines. The generation of the variable decryption routine (the polymorphic code) is split into two parts. The first routine generates the entry code, and the second creates the decryption loop.

The decryption loop consists of a variable number of randomly selected instructions such as XOR, ADD, SUB, DEC, NOT, ROR and ROL. The entry code loads the registers ready for use in the decryption loop, but contains a large number of ‘dummy’ instructions. Practically every 8086 instruction is present in this junk code, including instructions to access the Interrupt table, direct port IO and Int 21h calls.

Although Phantom_1’s polymorphic generator is highly advanced, it is not free from its fair share of errors. Under certain circumstances, the virus generates a valid decryption loop, but does not store its body encrypted. When such an unencrypted file is executed, the decryption loop *encrypts* the virus, producing garbage code. When control is passed to the virus body, the computer will crash. This is reminiscent of the MtE, which also produces code incapable of decrypting the body of the virus.

Animated Trigger

When the virus is active, it continually checks the contents of the keyboard buffer. If no characters are entered for about 20 minutes, the trigger routine is executed. This consists of a message display program, which hangs the computer when it is completed.

The screen effects used by the virus are on a par with its polymorphic routine. It only works on computers with VGA graphics, and utilises several of the features of the VGA card. When the routine is called, it first slowly fades the current screen image - just like the start of many computer games. Next, a skull appears, which blinks its eyes, followed by the text ‘PHANTOM 1’ in large red letters.

After a short period of time, the skull fades, and the following message is scrolled across the bottom line of the screen:

```
Congradulations!!! Your computer is now
infected with a high performance PHANTOM
virus! Coming soon: next virii based on the
_C00LEST_ mutation engine all over the world:
the Advanced Polymorphic Engine! Enjoy this
intro! (C) 1994 by Dark Prince.
```

The last message of the virus’ video effect begs the question of whether Phantom_1 has been compiled using a new, linkable polymorphic module. Should vendors be ready to encounter new viruses which use the same mutation engine?

Sadly, careful analysis of the virus shows that this is a strong possibility. Firstly, there are four different blocks of code in the virus: installation and infection code, polymorphic routine, trigger routine, interrupt tracing and hooking routines. The location of these blocks and some ‘programming signs’ seem to indicate that the four source files were compiled into different object modules which were subsequently linked together to form a dropper program.

Secondly, access to the polymorphic routine looks like a standard call to TPE, MtE, and NED polymorphic engines: there are several instructions to load registers with parameters of encryption, followed by a call to the polymorphic generator, which stores the encrypted virus body in the file. Both observations make it likely that this *is* a new engine - the world’s most complicated polymorphic engine to date.

Phantom_1

Aliases: None known.

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

Infection: COM and EXE files.

Self-recognition in Files:
E9h (JMP instruction) at the start of COM files. SP register field in EXE header set to 1000h.

Self-recognition in Memory:
Via ‘Are you here?’ call. Int 21h called with AX=ABCDh returns FFFFh in the same register.

Hex Pattern: No search pattern is possible.

Intercepts: Int 21h (infection), Int 1Ch (trigger routine). Int 01h, 03h, 24h, and 2Ah during infection process.

Trigger: If no keys entered via keyboard for 20 minutes an animated video sequence is run, and the computer hangs.

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

VIRUS ANALYSIS 2

Jack the Ripper

Benjamin Sidle

Jackripper is yet another new boot sector virus known to be at large both in the UK and in the rest of Europe. It is intentionally destructive, slowly corrupting the data on the hard disk. The message 'FUCK EM UP!' encrypted within the virus leaves no doubt as to the aims of its author. Jackripper's name is taken from another string within the virus body [*thankfully! Ed.*].

Initialisation

Jackripper infects the Master Boot Sector (MBS) of hard disks when the PC is booted from an infected floppy. On booting, the virus decrypts part of its boot sector in memory (if indeed a simple XOR-ing process can truly be considered encryption). It then decreases the system memory by 2K, and copies itself to this newly created free space.

The virus subsequently jumps to the high memory copy of itself. It then stores the address of the original Int 13h routine and reads the second sector of the virus code into memory. This is stored in sector 8, cylinder 0 on a hard disk, and in the penultimate sector of the root directory of a diskette (both 3.5- and 5.25-inch).

Next, it installs the address of the new Int 13h handler and reads the original MBS of a hard disk (which is stored at sector 9, cylinder 0), or the boot sector of a floppy (stored in the last sector of the root directory). This is loaded into memory at the location 0000:7C00h (i.e. where the boot code would normally be loaded). Its last act before jumping to the original boot code is to re-encrypt that part of the virus boot sector which was decrypted on booting.

Infection and Corruption

When the virus is active in memory, it uses stealth techniques to avoid detection. All read and write requests are redirected to the stored copy of the original sector. The second sector of the virus code, or the sector where the original Master Boot Sector is stored, will also be hidden from view. On a read request, a sector full of zeroes is returned. When a write request is made, it is not acted upon, and the virus copies its own boot sector to the Master Boot Sector. This causes the drive light to flash and indicate the expected activity.

Reads and writes to all other sectors are also intercepted. In the case of a write, there is a 1 in 1024 chance (based on the low byte of the clock count from Int 1Ah) that two words from the sector will be swapped before the write is completed. This corruption does not actually take place if the sector concerned is one where either the virus' own boot code, or the original boot code, is stored. A lower word of the clock count is read, and a value stored at a particular memory location within the virus is subtracted from this - the original word is then stored at this memory location. The new value is used in deciding whether to try to infect the drive being accessed.

If the infection process fails to write to a floppy disk (presumably due to write-protection), the carry flag is cleared, and no error condition is displayed.

Conclusions

The virus code is somewhat erratic, and the fact that a part of the virus boot sector is encrypted is no barrier to its disassembly. The true purpose of the encryption seems to be to hide the two text strings within the virus boot sector.

When a floppy disk is infected, the messages at the end of the boot sector are preserved within the new virus boot code; thus, a casual glance at the boot sector will show nothing amiss. By the same process, the partition table is included in the new MBS on hard disks.

As the virus only corrupts on writing, the files most likely to be affected are data files. Therefore, by the time an infection is discovered, it is possible that data stored on disk has been slightly corrupted.

Jackripper

Aliases:	None known.
Type:	Memory-resident, Master Boot Sector infector.
Infection:	Boot sector of bootable media.
Self-recognition on Disk:	
	40 bytes of new Int 13h routine.
Self-recognition in Memory:	
	Compares contents of disk with image in memory.
Hex Pattern:	
	8BFE 0E1F 0E07 AC34 AAAA 5781 E7FF 0081 FFDF 005F 75F0 33C0
Intercepts:	Int 13h Read and Write requests.
Trigger:	Gradual clock-triggered data corruption.
Removal:	Easily removable under clean conditions. Data recovery difficult.

VIRUS ANALYSIS 3

Misis: Interrupt Interruption

Jim Bates

This month's nomination for 'Singleminded Simpleton 1994' is the creator of a boot sector virus reported from a university in the UK Midlands. The Misis virus is only 279 bytes long (excluding what seems to be a foreign language message), but the design is so incredibly careless that during analysis I had to check and recheck my results in order to be absolutely sure of what I was seeing!

What makes Misis slightly unusual is that it is capable of remaining memory-resident on an infected PC without changing the available memory. The feat is achieved by the simple (but incredibly unreliable) trick of storing the virus code in the interrupt table of the machine.

Installation

The Misis virus infects the Master Boot Sector of fixed disks and the boot sector of floppy disks. When a machine is booted from an infected disk the virus becomes active, and the following sequence of events occurs:

After initialisation of various register values, a request is issued to read the Master Boot Sector (Track 0, Head 0, Sector 1) of the first fixed disk on the system. If the contents stored at offset zero of the boot sector is C933h, this disk is assumed to be already infected.

If the disk appears to be uninfected, the original contents of the Master Boot Sector are written to Track 0, Head 0, Sector 6 and the virus code is written to the Master Boot Sector in its place. If the fixed disk is already infected, the infection routine is skipped, and processing jumps to the installation code described below.

The virus then relocates its code in memory and installs its own routine to intercept calls to the system disk services (Int 13h). Finally, a soft reboot call is issued.

It is the relocation position which caused such concern during detailed analysis of the virus, because it displays such a cavalier disregard for proper programming practice. The complete virus code is loaded into the memory normally used for storing the upper half of the system interrupt table. Any subsequent insertion of vectors for interrupts 94h to B2h will destroy the integrity of the virus code and cause unpredictable system malfunction.

It seems that the code was located in this way to avoid decreasing the memory available to DOS. Memory stolen in this way is easily detected, and several anti-virus programs use it as an indication of the presence of boot sector virus code. Whilst the interrupt vectors overwritten are infrequently used, they do form part of the vital architecture in all machines, and interference with them makes a system crash almost inevitable.

Operation

As with most boot sector viruses, the whole operation of Misis centres around its Int 13h interception routine. This intercepts all requests for disk access and checks to see if the request is for read access to the boot sector of the floppy drives, or the MBS of the first fixed disk drive (the virus ignores the possibility of additional fixed drives). If the call does not fulfil these conditions, processing passes to the original Int 13h handler.

Once interception is properly under way, the virus completes the request call and loads the Master Boot Sector into the caller's buffer. The address of this buffer is stored by the virus for later reference. Processing then branches, depending upon whether the intercepted request was for access to the fixed disk or one of the floppy drives.

If it was for the fixed disk, a routine is called which decides whether to invoke the virus trigger routine (see below). After this has completed (with or without the trigger display), the original Master Boot Sector is collected from Track 0, Head 0, Sector 6, and returned to the calling routine without error.

"The complete virus code is loaded into the memory normally used for storing the upper half of the system interrupt table"

If the original request was for access to a floppy drive, processing attempts to verify the existence of Track 0, Head 0, Sector 12. This effectively distinguishes between high density and low density floppy disks. The virus uses this information to set a target address of Track 0, Head 1, Sector 3 for low density or Track 0, Head 1, Sector 12 for high density disks. This target address is used to store the original boot sector after the virus has ensured that the floppy is not already infected. Infection of the disk is thus accomplished by simply copying the infected MBS of the fixed disk to Track 0, Head 0, Sector 1 of the floppy disk.

Once this process has been completed, processing passes to the trigger checking routine.

Trigger Operation

The actual trigger routine is preceded by a check of the system timer. This is done in such a way that the trigger has a chance of operating approximately one in every 16 times any disk's boot sector is accessed. The trigger process displays a message at the top of the screen, but since the message text appears to use a foreign character set, I am unable to quote exactly what its contents are. There is no attempt at encryption of the message (or any of the virus code), and two areas of the code contain the text messages 'Soft 236-25-35' and 'NIKA!'

Once the message has been displayed, processing waits for a key press before returning control to the calling routine. On machines equipped with a colour monitor, the message will appear as flashing yellow on a red background. The length and start of the message text is randomised so that not all of the message will be seen at any one time.

It should be mentioned that while the virus conducts a simple check of video screen mode, this is only to determine the address of current video memory. Thus, if the controlling program is operating in graphics mode, the message will appear as slight corruption on the top few lines of the screen.

Conclusions

Although the Misis virus is in no way outstanding, the mere fact of its existence makes it yet another straw on the back of anti-virus software developers. There is no encryption or stealth capability so recognition is simple. Similarly, disinfection is easy, although the usual caveats concerning the removal of boot sector viruses should be observed.

Misis

Aliases: NIKA.

Type: Memory-resident, Master Boot Sector.

Infection: First fixed disk drive and all write-enabled floppy disks.

Self-recognition in Memory:

Checks for presence of virus code at memory location 0000:0253h.

Self-recognition on Disks:

Contents of offset zero of the Master Boot Sector is C933h.

Hex Pattern:

```
C0B0 538B F8BE 537C B175 F3A4
BE4C 00A5 A5A3 4C00 8C06 4E00
```

Intercepts: Int 13h - redirects requests for Master Boot Sector.

Trigger: Displays apparent garbage on UK machines. This may represent a message when using different character sets.

Removal: Disinfection possible under clean system conditions.

TUTORIAL

Link Viruses

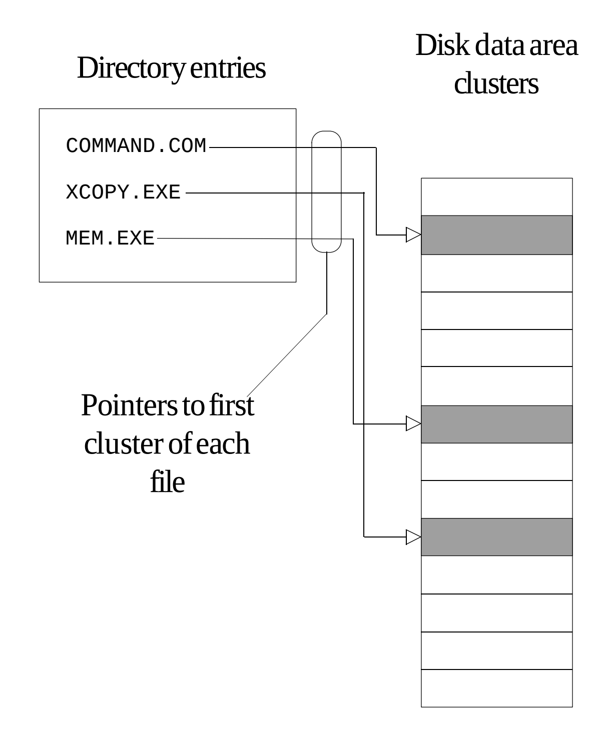
In the autumn of 1991 a new virus was discovered, which used a new technique to infect target files. The virus, named DIR-II, spreads extremely quickly, infecting all executable files at once. However, several years later, there is still a great deal of confusion amongst users about precisely how the DIR-II virus infects files on a disk.

Disk Structure

The data stored on disks is kept in units known as clusters which are stored on the disk in a group of Sectors. DOS gives each cluster a number (an address) by which it distinguishes the different parts of the disk. The cluster number is translated into a call to read a particular area of the disk by the BIOS.

The information stored on a fixed disk is stored in four primary blocks: the Boot Sector, the Root Directory, the File Area and the File Allocation Tables (FAT). Each of these structures serves a different purpose.

The Boot Sector of the disk contains executable code which loads the operating system. The Root Directory, which is created when a disk is formatted, contains a series of 32-byte



On an uninfected hard disk, each directory entry points to the first cluster of the appropriate file.

directory entries, each of which contains the name of a file, a subdirectory or a disk volume label. The File Area makes up the last and largest part of a disk, and contains the data files and subdirectories which make up the data stored on the disk. Each file name contains the cluster number at which the file starts.

Under ideal conditions, a file stored in this area is made up of one contiguous block. However, because files are continually erased and created on the disk, the space available does not always consist of contiguous sectors. The ‘breaking up’ of a file is known as file fragmentation. This slows down the speed at which information can be retrieved from a disk, as the hard drive has to access information from numerous different locations, and it takes a finite time for the heads to move around the disk. Such disk fragmentation is repaired by such programs as *Norton Speed Disk* or the *DOS 6.0* DEFRAG utility.

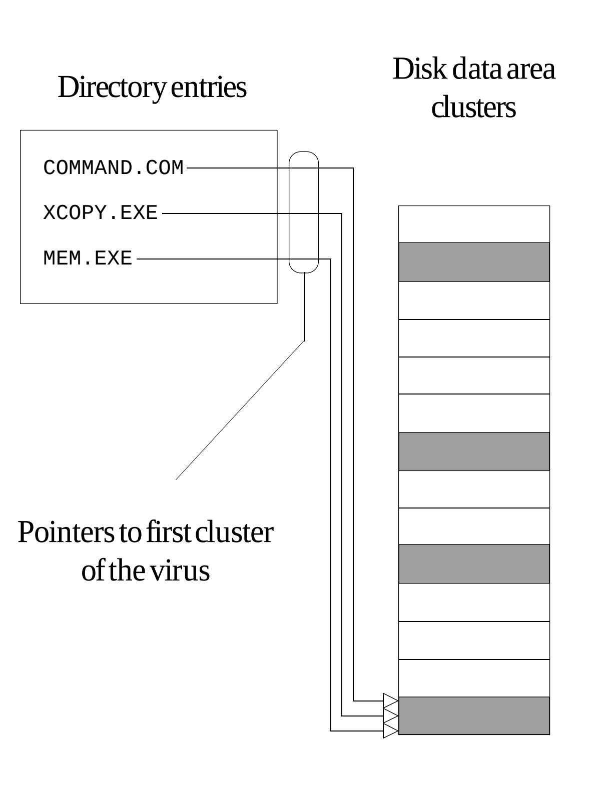
The sequence of clusters used to make up a file is recorded in the FAT in the form of a ‘linked list’. A linked list is one in which each entry in the list contains the information needed to locate the next entry. When a file is accessed by DOS, it uses this information to identify and load the sectors which make up the file. Anyone familiar with the DOS program CHKDSK will have come across the expression of a lost ‘chain’. This is simply an area of the FAT which is marked as allocated, but which is not pointed to by any of the directory entries.

Daisy Chains

It is possible to take advantage of the way DOS locates files to infect them without changing any of the actual code within the file. If the starting cluster of a file is reset to point to virus code rather than to the file itself, the virus will be loaded instead of the file. In the case of DIR-II, the virus code subsequently loads the original uninfected file into memory. This technique is easily expanded to every file on the disk: the virus sets the starting cluster of each executable to point to a cluster which contains the virus code. When the virus is executed, it correctly loads the file using data stored within the virus.

This technique is very effective for three main reasons. Firstly, it is only necessary for virus code to be stored in one location on the disk, massively reducing the amount of disk space used when infecting a single disk. Secondly, virus propagation is very quick, as the entire disk can be infected in one pass, guaranteeing that the virus is memory-resident for the maximum possible time. Thirdly, it is possible to make file disinfection an extremely difficult and time-consuming task.

In the case of the DIR-II virus, recovering the information stored on the disk is trivial. The virus only changes the directory entries in the case of executable files, so data is unaffected. However, if the machine is booted from a clean system disk, the directory entry for every executable file on the machine points to the same 1024 bytes of virus code.



On an infected disk, the virus alters all the directory entries so that they all point to the cluster which contains the virus code. Once the virus has received control, it loads in the appropriate code.

This ‘cross linking’ (where several directory entries point to the same piece of disk space) of executable files led many people to believe that DIR-II caused a vast amount of damage to hard drives. In fact, the truth is quite the opposite, as DIR-II can actually be removed completely from an infected disk drive without using any anti-software at all!

With the virus memory-resident, rename all COM and EXE files on the hard drive to a non-executable extension. The virus does not cross link these renamed files, as it no longer considers them to be executable. Once every executable on the affected machine has been renamed, the machine can be turned off and clean booted. Every directory entry on the disk now points to its corresponding code - all pointers to the virus code have been eradicated.

Conclusions

When DIR-II was first discovered, it was seen to be a major threat. However, in the last two years, only a handful of new viruses which use this ‘linking’ technique have been discovered. The reasons for its scarcity seem to be that the infection process is rather more subtle than the brute force approach of parasitic file infection. Additionally, the programming techniques used are a little more complicated, and rely on an understanding of the disk and its structure. Link viruses may seem to be a powerful new technique, but in fact they present no more of a threat than any other new virus; they merely operate in a different manner.

FEATURE

VB Survey: The Results

Readers will remember that a questionnaire on computer viruses, anti-virus software, and virus prevention policy was included in the January edition of *Virus Bulletin*, in order that the magazine might identify more clearly the requirements of its readership.

Replies were received from all over the globe, from organisations ranging from a site with 3 PCs and no network, to a multinational corporate with 35,000 PCs and even more minicomputers and mainframes. Every reply was used when compiling the following statistics, and *Virus Bulletin* would like to thank all those who took the time to complete and return the survey.

Readers' Sites

The replies received by *Virus Bulletin* were grouped into three sets, classified by size. The smaller sites (which were classed as those with fewer than 100 PCs) were represented by 20% of the replies; medium-sized sites (100 - 999 PCs) by 37% of the completed questionnaires, and the larger sites (with over 1000 PCs), by 43%.

IBM-compatible PCs are by far the most widespread choice, with the *Apple Macintosh* lagging behind in second place - this was fairly constant, and did not vary with company size. Many sites use minicomputers, and (particularly the larger companies) mainframes.

The great majority of sites were networked: unsurprisingly, only some of the smallest (about 14%) were not. Of those sites which were networked, the most popular system was *NetWare 3.x*: this was used by 43% of smaller sites, 64% of medium-sized sites, and 75% of the larger sites. The most popular alternatives to *NetWare 3.x* were *NetWare 4.0* (used predominantly by the large sites), *LanManager*, *LanServer*, *PathWorks*, and *LanTastic*.

Anti-Virus Policies

Every organisation which completed the questionnaire took some anti-virus precautions. Only 86% of all companies who replied claimed to have a policy of scanning incoming disks; one hopes that this figure is a result of an omission from the form, rather than from the policy.

Apart from scanning incoming disks, many respondents implement other anti-viral policies, including scanning workstations, TSR virus protection, server-based scanning, disk authorisation, and checksumming. These, although the most popular methods, were not the only ones: some companies use access control, scan outgoing disks, and have server as well as workstation scans.

Scanners

There is a glut of scanners on the market, and this was well-reflected in the choices made by those who participated in the survey, with over twenty different products in use. In the smaller companies, the most highly-regarded anti-virus scanner was *Dr Solomon's Anti-Virus Toolkit*, with *F-Prot* and *McAfee SCAN* lying not far behind.

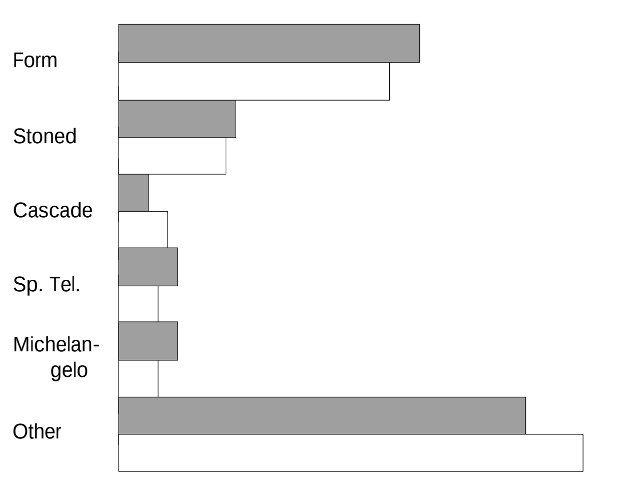
The larger companies claimed to use a similar selection of products, with many opting for more than one scanner. This provided some interesting comparative ratings of products: users unsurprisingly found the most popular products to be the best, with *MSAV* being almost universally criticised.

F-Prot, *Dr Solomon's AVTK* and *Sophos Sweep* were consistently rated highest within this group, with most users considering them to be excellent. Users of *McAfee's SCAN* found it to be adequate on the whole, while *Norton Anti-Virus* was generally considered rather mediocre, with some users finding it less than sufficient. Most of the larger companies use more than one anti-virus package, with many using a combination of up to four or five.

It was reassuring to note that approximately 50% of *VB* readers do update monthly, with the majority of the remainder going for quarterly updates. Astonishingly, there was one site which claimed it did not update any one of its four anti-virus packages at all, and one site which did not even have a virus scanner!

Checksumming and Disk Authorisation

The number of companies which use checksummers was fairly constant, with one in every three opting for this additional prophylactic. The range of checksumming



The table above represents interceptions (white bars) and infections (grey bars) experienced by respondents. It is clear from the diagram that Form is the most widespread virus today.

RESULTS SUMMARY

Site Size of Respondents:

0 - 99 PCs	33%
100 - 999 PCs	37%
1000+ PCs	43%

Use of Scanners:

McAfee SCAN	36%
F-Prot	23%
Sophos Sweep	23%

Average Cost of Virus Infection: US$150

Types of Measures Taken:

Scanning disks	86%
Scanning workstations	68%
TSR on workstations	50%
Server-based scans	47%
Disk authorisation	15%
Checksumming	30%

Of the larger companies, not one escaped the year unscathed: all have experienced actual infection. Ten percent of medium-sized, and 29% of small companies experienced no infections within the past 12 months, and 7% of medium-sized and 57% of small companies had no interceptions.

products was quite large, with the most popular being *ViVerify*, from *Dr Solomon's AVTK*, and *Sophos' Vaccine*. It was interesting to note that some of the larger companies had developed their own proprietary checksumming programs.

The use of disk authorisation software was much less widespread: 20% of all small and medium sized corporates claimed to use some form of software; this dropped to only 6% for the large companies. It is possible that the maintenance overhead of this technique is off-putting to sites which are very large.

It is difficult to say with any real certainty just how effective disk authorisation is as a virus preventative, due to the relatively small statistical sample. However, it is clear that it does not stop infections. Indeed, in every group, every company which had implemented disk authorisation had had a virus infection within the past twelve months. The probability of infection seemed unchanged, although the survey did not take into account the type of environment in which companies operated.

Virus Interceptions and Infections

Unsurprisingly, almost every respondent had intercepted a virus coming into a system in the past twelve months, even if they had not experienced an actual infection. The larger sites experienced proportionately the most infections: at least some of the small and medium-sized sites were able to report that they had had neither virus interceptions nor outbreaks, but this was not the case at large sites. None escaped completely unscathed, and 64% of all companies had experienced a virus infection in the last six months.

Those viruses which infected most frequently were Form, New Zealand, and Spanish Telecom. A table of the most common viruses is shown opposite. It is interesting to note that there is a pattern in the virus prevalence: three viruses are extremely common, with a large number of different viruses being seen only by one or two sites.

As was to be expected, the variety of viruses which were intercepted before infection was usually greater than those which actually infected, though the overall pattern was the same.

Several companies indicated that they felt either that viruses were not a problem for them, or that too much attention was being paid to viruses. Of these companies, only one was experiencing, in real terms, very few virus attacks.

The question of virus costs was also raised: readers were asked how much each post-infection clean-up cost a company per PC. Answers ranged dramatically, ranging from 'negligible' to US$1,500. The median seemed to be from US$50 to US$250 per PC, per incident. One company also commented that they had had eleven false alarms, and that they were 'just as expensive as the real thing'.

The Computer Misuse Act

Unfortunately, of the UK-based respondents, very few - approximately 25% - reported computer viruses to the *Computer Crime Unit* at *New Scotland Yard*. This was despite the fact that several of those who commented on the law thought that it should be strengthened. The Police allocates resources commensurate with the number of complaints they receive. If virus incidents go unreported, the *CCU* will find it increasingly difficult to justify its existence, when the truth is that it is badly needed. All reports to the *CCU* are treated in the strictest confidence, and further investigation into an outbreak will only be undertaken with the consent of the company concerned. Readers are strongly urged to take the time to report incidents.

Conclusions

Although the results of this survey held few surprises, it is clear that the virus problem is very real. Every large site which replied to the survey had experienced at least one virus infection and interception in the last twelve months. Additionally, over forty different viruses were reported as 'in the wild', including one incidence of Brain, the first *IBM* computer virus.

The statistics presented here show beyond any doubt that computer viruses are an everyday business problem. Most companies have experienced an attack in the last six months, and there is no reason to assume that the situation will improve. However, countermeasures seem to be working; continued vigilance must be the order of the day.

PRODUCT PREVIEW

Virus-Anti-Virus

Dr Keith Jackson

VB usually provides in-depth reviews of anti-virus products. However, *Virus-Anti-Virus* is very unusual, and it was decided that a brief preview of this product's capabilities could be worthwhile.

Virus-Anti-Virus (*V-A-V*) appears to use state-of-the-art virus technology to defeat viruses, a bold approach if ever there was one. The marketing blurb claims that 'techniques used in constructing viruses are now being put to good use in detecting them'. It also claims that *V-A-V* is certified to US *Orange Book* level D.

Installation and Overview

The product does seem to provide a radically new method of protecting PCs. *V-A-V* attaches itself temporarily to executable files, and relocates itself from one to another in turn. Whatever the PC user is doing, *V-A-V* worms its way through the disk in the background, any viruses found are eradicated, and warning messages are displayed.

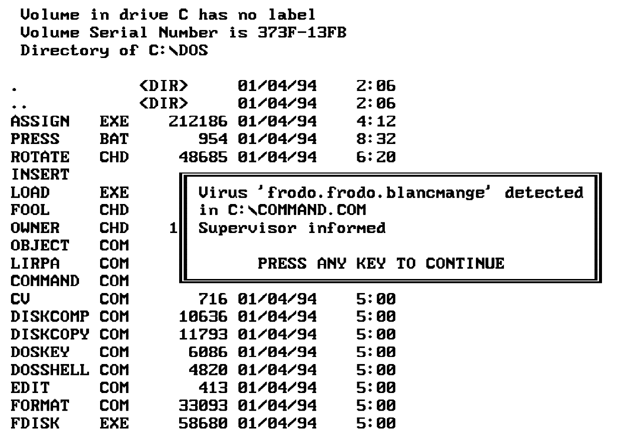
This anti-virus virus is not constrained to a single PC; it is capable of moving from one system to another using modems and telephone lines to find a likely target computer. It can take full advantage of wide area network connections using its exploratory Heuristic Internet Communications Connection Usage Protocol algorithm (*HICCUP*).

Getting Around

I started testing *V-A-V* by timing how long it took to move around my hard disk (534 executable files spread across 18.6 Mbytes), so that it had been attached to every executable file at least once, and had verified that they were virus free. This background 'hop & inspect' method of operation was completed in 4 minutes 55 seconds.

For comparison purposes, *Dr Solomon's Anti-Virus Toolkit* scanned the same hard disk in 55 seconds, and *Sweep* from *Sophos* took 1 minute 12 seconds for a quick scan, and 2 minutes 54 seconds for a complete scan. Given that while a conventional scanner operates it prevents other programs from executing, *V-A-V's* timings seem reasonable.

Licensing of *V-A-V* will come in various flavours, which are restricted in their sphere of influence according to cost. *VB* was provided with the local version which will spread only through the local telephone exchange, but it is possible to extend its scope to cover wider areas at increased cost. Comprehensive configuration options are available for *HICCUP*, including the ability to prevent it accessing UK premium-rate 0898 or, in the USA, 900 numbers. As *V-A-V* copies itself from one file to another, it leaves an indication of where it is going next. *V-A-V* upgrades are released bi-monthly, and are designed to use this so-called 'Hansel und Gretel' system. They follow the original release around the Internet, catch up with it, and update its virus database.



```
 Volume in drive C has no label
 Volume Serial Number is 373F-13FB
 Directory of C:\DOS

.            <DIR>     01/04/94    2:06
..           <DIR>     01/04/94    2:06
ASSIGN   EXE    212186 01/04/94    4:12
PRESS    BAT       954 01/04/94    8:32
ROTATE   CHD     48685 01/04/94    6:20
INSERT
LOAD     EXE          Virus 'frodo.frodo.blancmange' detected
FOOL     CHD          in C:\COMMAND.COM
OWNER    CHD     1    Supervisor informed
OBJECT   COM
LIRPA    COM                PRESS ANY KEY TO CONTINUE
COMMAND  COM
CV       COM       716 01/04/94    5:00
DISKCOMP COM     10636 01/04/94    5:00
DISKCOPY COM     11793 01/04/94    5:00
DOSKEY   COM      6086 01/04/94    5:00
DOSSHELL COM      4820 01/04/94    5:00
EDIT     COM       413 01/04/94    5:00
FORMAT   COM     33093 01/04/94    5:00
FDISK    EXE     58680 01/04/94    5:00
```

Virus-Anti-Virus seems very effective at detecting viruses, although it does require the temporary alteration of executable code.

Documentation

The user manual is supplied in machine-readable form, using its own Stochastic Non-Expanding Executable Zip-Extractor self-extracting archive technology (*SNEEZE*) coupled with a special Predictive Operational Objective Program module (*POOP*), which in turn uses the *HICCUP* module to anticipate *V-A-V's* movements and ensures that a copy of the documentation always arrives in advance of the software. As far as I could test them, both *SNEEZE* and *POOP* worked fine. My main gripe is that there is no sign of a decent index - surely an essential item in any self-respecting software package these days.

Reviewing Problems

Reviewing this product in detail is certain to prove challenging. Even with the local exchange-only version which sent itself to *VB* for review, I eventually had to spend several days driving around with a pair of binoculars and an Ethernet analyser trying to track it down.

[*Reports have been received of Virus-Anti-Virus spreading wildly on PCs attached to the Internet. VB has recommended that the developers of the software write a similar program which will chase unlicensed copies of V-A-V around the Internet and try to eradicate them before they automatically report unwitting users to the Federation Against Software Theft. Ed.*]

Product Details

Product: *Virus-Anti-Virus*

Developer: *Euvbinad Ltd.*, 1-4 Telephone Place, Southend-on-Sea SS1N 2ES, UK, Tel. +44 (0)702 8082, Fax +44 (0)702 8082

Price: TBA.

PRODUCT REVIEW 1

Intel LANDesk Virus Protect

Jonathan Burchill

Intel LANDesk Virus Protect is designed to protect all components of a *Novell* network, ranging from multiple file servers to nomadic laptops which only come into occasional contact with office machines. *Intel's* advertising literature states that the product 'is the most complete enterprise-wide protection available'. A tall claim, given the reputation of some competitors. How well does the product live up to it?

Parts and Requirements

Like most other *Novell* anti-virus packages, the software comes in two parts: one group of files for the server, and one for the workstation. The server software must be run under *Novell NetWare 3.11* (although *VirusProtect 2.1*, which is due for release in April, has additional support for *NetWare 4.0*), and requires at least 200K of free RAM. The product is capable of detecting both *IBM PC* and *Apple Macintosh* viruses, and workstation software is provided for both these platforms.

As I do not have access to a *Macintosh*, the remainder of this review will concentrate on the file-server and DOS-based parts of the package. The only specification for the DOS workstation is that its operating system must be DOS 3.3 or higher in order to run. Administration and configuration of the NLM is carried out from a workstation (anything from a 386 upwards) with at least 2 Mbytes of extended memory and 512K of free conventional memory. The manual states that the administration station must be using at least DOS 5.0, and I can testify that the 512K is a pre-requisite.

Product documentation is sadly deficient, limited to explaining options within the software. No general information on viruses or good anti-virus policy is included. There is also no virus encyclopedia, either in printed or electronic form.

Variable Interface

Versions of the software are included for both DOS and *Windows*. Not all programs have exact counterparts, and functionality between the two versions differs slightly. Programs for the DOS environment range from having no GUI at all, to (at least) two different styles of windowing interface. The administration program has a mousable, graphical user interface of the type which makes one double-check that one has not accidentally started *Windows*, whilst the configuration program for the execution monitor has an entirely different appearance.

This variety of interfaces neither helps give the package a cohesive look and feel, nor inspires overall confidence in the product. This is a shame, as elsewhere there are some nice touches. I personally prefer a good DOS interface to a *Windows*-based solution in anti-virus products: they are utility products, not applications.

Installation

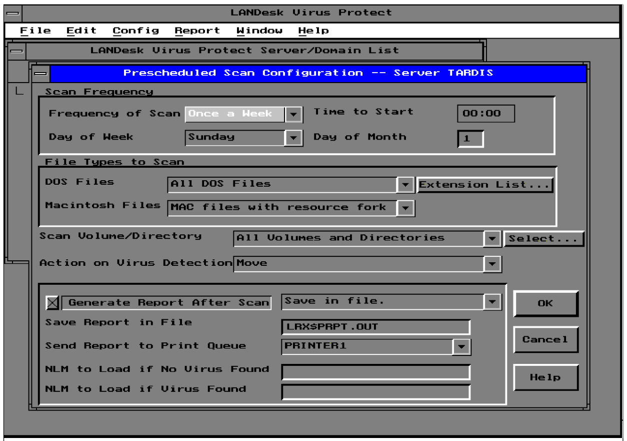
The product was supplied on four 3.5-inch floppy disks, each appropriately labelled as disk *n* of four. I mention this because if one opts for the DOS only installation, it is necessary to start with disk three of four (which, admittedly, was called DOS Install). The user is then asked for disk two, which contained the server code. Neither the quick setup guide nor the manual informed me of this fact.

One nice feature of the installation program is that it creates a file named TODO.TXT. The concept behind this file is that of an *aide memoire*: if options such as not allowing the install program to modify the server startup file were chosen, then referral back to this file would serve as a reminder of the tasks which still needed completing. This is an excellent idea. I hate to remember how often I have had to re-install a piece of software just to note down the terse message given after choosing 'Do not Modify'. One major drawback of TODO.TXT is the fact that it is generic, and does not reflect the particular install options and directories chosen.

When running in a multiple server environment, the software allows the server to be grouped into domains. The same configuration and settings are automatically shared amongst servers in the same domain, helping with control and set-up of large networks. The administration program has options for cutting and pasting selected options between domains.

Component Parts

Server protection consists of a traditional virus scanner, and a rules-based execution monitor which checks file read and write requests and looks for suspicious activity which might


LANDesk Virus Protect
File Edit Config Report Window Help
LANDesk Virus Protect Server/Domain List
Extension List...
Select...
OK
Cancel
Help

Prescheduled Scan Configuration -- Server TARDIS

Scan Frequency

Frequency of Scan: Once a Week
Time to Start: 00:00
Day of Week: Sunday
Day of Month: 1

File Types to Scan

DOS Files: All DOS Files
Macintosh Files: MAC files with resource fork

Scan Volume/Directory: All Volumes and Directories
Action on Virus Detection: Move

☒ Generate Report After Scan: Save in file.
Save Report in File: LRX$PRPT.OUT
Send Report to Print Queue: PRINTER1
NLM to Load if No Virus Found:
NLM to Load if Virus Found:

Virus Protect provides good management facilities, but is sadly let down by mediocre virus detection rates.

be the result of virus code. The documentation gives no information whatsoever on what might be regarded as suspicious activity. It is not possible to configure the server-based execution monitor (beyond who receives which alert messages) or to enable or disable it actively. I can only say that when I was trying it, I had no false alarms.

Without at least some information as to what it regards as suspicious activity, it is almost impossible to assess how much reliance can be placed on this feature. Also, I could not see any provision for allowing an exception list if the rule monitor is falsely triggered by, for example, an executable which stores the current configuration back to itself. Under such circumstances, this lack of flexibility would become rather a nuisance.

> *“the program will automatically download the latest software updates and signature files, and update itself”*

Server-based scanning can be configured to provide realtime scanning (of incoming or outgoing files, or both), prescheduled scans on almost any imaginable frequency (e.g. every Monday, or the first of every month), and manual scans. Options are provided to limit the scan by file type and extension, as well as by server volumes and directories. There is, however, one shortcoming: only one type of prescheduled scan can be defined. This prevents, for example, choosing a quick scan of only users’ directories on some days, with a more extensive server scan on others. Once prescheduled or manual scans have been completed, it is then possible to specify NLMs to be loaded according to the outcome of the scan.

Various actions may be taken if a virus is found. These include renaming its extension to ‘vir’, changing its execute privilege, deleting it, or moving it to a predefined directory. Notification of virus detection is limited to an optional custom message, sent to the offending user and/or to a specified group of users. These messages are normally sent as *NetWare* broadcasts, but may also be sent as MHS mail if so required. Messages are allowed to contain some runtime information - %V, for example, becomes the name of the virus, and %F, the file name: ‘Found %V in File %F’ could be displayed as ‘Found FRODO in 4K.COM.’

The administration program includes a realtime monitor of the *LanProtect* NLM activity. This shows CPU loading and scanning activity, and the most recent virus detections.

Messages from workstation and server software are sent to a centralised logfile. The administration software provides a viewer for this file which includes a degree of filtering. Unfortunately, it is not really comprehensive enough, and, as *Intel* declines to document the data format of the log file, it would be difficult to use a third part file viewer.

Workstation Protection

DOS and *Windows* workstation protection is provided by a combination of several ingredients, consisting chiefly of a virus scanner and two Terminate and Stay Resident programs. The on-demand virus scanner is capable of scanning both local and (optionally) network drives, and can be used from DOS or from *Windows*.

The TSRs consist of an on-access scanner for files, together with an execution monitor. Both are similar to their server-based cousins. The workstation and server software uses exactly the same virus signature and execution rules database. Using the same database helps ensure that updates will be effective across the whole network at once.

It is not necessary to copy the software to each workstation; the TSRs can actually be loaded from the file server during the workstation login. *Intel* provides a special LOGIN.COM supplement to help with this, which moves LOGIN.EXE to the top of memory, therefore allowing the runtime-loaded TSRs to be loaded at the bottom of memory. This prevents memory from being fragmented and lost.

The workstation scanner is actually more capable than the server scanner, as it can look inside files compressed using PKLITE and LZEXE. Additionally, unlike the server software, the workstation software will attempt to clean an infected file and to replace damaged boot sectors. The workstation execution monitor can also alter the degree of stricture it applies to program activity.

It should also be noted that the DOS scanner is dated six months later than the NLM: 7 December 1993, as opposed to 29 June 1993 for PSCAN311, the scanner for *NetWare*.

Mobile and Home Users’ Protection

Utilities are included to copy the workstation protection software from the file server to the workstation. This feature enables users to have the protection software loaded when they are not logged in, and to make an installation floppy which may be used to protect mobile or home computers. The license agreement specifically allows for this: it is a very important factor in any virus protection scheme to include all machines which may at some point be connected to the network or which might generate files which will be transferred there.

When a transitory machine finally reconnects to the network, the VPDOCK program checks that the local and server signature and rules databases are in synchronisation, and uploads the results of any virus scans or detections to the centralised database. This is a particularly useful and well-thought-out feature.

Downloading New Virus Patterns

Included in the package is the VPDOWN program, which provides automatic updates to the software via the nearest *Intel* BBS. It is necessary only to supply this program with the relevant telephone number, and the port to which almost any *Hayes*-compatible modem is connected: the program will automatically download the latest software updates and signature files, and update itself.

This feature worked absolutely flawlessly, and is a great way of helping to ensure that signature files are kept up to date. The VPDOWN program can be made part of the administrator login, and will limit downloads to once a month. The transfer was totally automatic and updated both the database files and the LPROTECT.NLM. The server software automatically picked up the newer databases and informed me of the version number, despite the fact that the old database was still present on the disk.

This is a very good feature, and one which could simplify the job of an overstretched network manager. One criticism is that it failed to warn me that the NLM itself had been updated, and that I needed to unload the current one in order to pick up the latest version. This seems to be a rather careless omission.

Virus Detection Results

One of the greatest surprises I had when testing the product was the large differences in performance between the workstation product and the NLM. Overall, the DOS product fared better in the detection tests, scoring 87.2% in the In the Wild test-set, 96.9% in the Standard test-set, and 80% in the Polymorphic test-set. The NLM achieved 78.9%, 94.3% and 1.7% respectively in the same tests. The polymorphic test results are abysmal, especially given the DOS results on the same files. Why the difference? The lacklustre performance of the server-based scanner is inexcusable. No matter how feature-packed an anti-virus NLM may be, the most important attribute is its ability to detect viruses; *Virus Protect* fails this most crucial test.

Conclusions

This product really does pose some difficult questions. It has some clever touches which I had not seen before: these include the TODO.TXT file generated at install time, the provision for protection for laptop and home users, the inclusion of a test virus (which is actually not a virus at all, but will trigger the pattern scanner and the execution monitor), the automatic and free downloading of new signature patterns, and the low overhead of the scanner on the file server [*Full timings will be published in a subsequent comparative review. Ed.*].

Against this must be balanced the insufficient technical documentation, the inconsistent user interfaces and, worst of all, the poor detection results. This last is a very serious consideration, as the product includes no file checksumming ability. If one were to rely solely on *LANDesk*, it would have to detect 100% of the In the Wild test-set, and preferably the Standard test-set as well. In a server-based environment, infections can propagate quickly: the bottom line is that *Virus Protect's* detection rate is not good enough.

LANDesk Virus Protect

Detection Results:

NLM Scanner		
Standard Test-Set [1]	216/229	94.3%
In the Wild Test-Set [2]	86/109	78.9%
Polymorphic Test-Set [3]	8/425	1.7%
DOS Scanner		
Standard Test-Set [1]	222/229	96.9%
In the Wild Test-Set [2]	95/109	87.2%
Polymorphic Test-Set [3]	339/425	71.4%

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: *Intel LANDesk Virus Protect, v2.0*

Manufacturer: *Intel Corporation*, 5200 NE Elam Parkway, Hillsborough, Oregon 97124, USA.
Tel. +1 503 629 7354, Fax +1 503 629 7580

Distributor (UK): *Intel Corporation*, Piper's Way, Swindon Wiltshire SN8 2BS.
Tel. +44 (0)793 696000, Fax +44(0)793 444447

Price: Single server version £699, 4-server version £2099, 20-server version £6999. All include monthly updates.

Hardware Used: Server - 33 MHz 486, EISA bus, 32-bit caching disk controller, *NetWare 3.11*, 16 MB RAM. Client - 33MHz 486, 200MB IDE driver, 16 MB 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-T1, 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 425 genuine samples of: Coffeeshop (375), Uruguay.4 (50).

PRODUCT REVIEW 2

PC Defender

Dr Keith Jackson

PC Defender is different from other 'standard' anti-virus products: it comprises a plug-in card for the PC, along with various programs to be executed from disk. The plug-in card consists of an EPROM, which contains software, and a PAL and Octal Buffer, which provide access to the PC bus.

The plug-in card, described in the accompanying documentation as a 'BOOTMonitor', claims to provide comprehensive protection against boot sector viruses by executing its on-card software before DOS commences execution. This software looks for boot sector viruses before they get the chance to circumvent DOS, and refuses to let the DOS boot sequence proceed if anything suspicious is found. There is no processor on this plug-in card, so execution of software contained in the EPROM relies on the PC delivering control to that software at the beginning of the boot sequence.

Documentation

The documentation which comes with *PC Defender* is a single A5 booklet, 72 pages long. This contains an index which must have taken at least 30 seconds to prepare: it is so terse as to be basically useless. The content of the manual is minimal, but it does explain the basic functions of each component of *PC Defender.*

The explanation of the default factory settings for the jumpers on the plug-in card is incorrect in the manual, and some of the screens shown in the documentation are not identical to the visible screens. Neither of these facts will help the inexperienced user. [*AMI is currently working on a revised manual, which it plans to ship in 6 weeks. Ed.*]

Installation

Installation of the plug-in card is very easy - just find an empty slot and plug it in. Installation of the software proved more difficult: this is provided only on high-density disks (3.5-inch, 1.44 Mbyte, and 5.25-inch, 1.2 Mbyte). The computer which I had intended to use for testing had both 3.5-inch and 5.25-inch floppy disk drives, but being an XT, both were low-density drives (most XTs cannot work with high-density floppy disk drives).

I was pleased to see that product installation can take place from any subdirectory, so I split its files into two groups, and copied each group to a low-density 3.5-inch floppy disk. I then copied the files into a subdirectory on the fixed disk, and installed from that subdirectory. For at least a year I have been railing against various anti-virus products not providing low-density floppy disks: *PC Defender* is another in a growing list of problematic products.

During installation, the product copied all the necessary files across (taking over 2 minutes to do so), then modified the PC setup files CONFIG.SYS and WIN.INI. After saving unmodified copies of both these files, the installation program offers to install the memory-resident anti-virus monitor program. This would have been acceptable, even useful, had it not tried to access drive C (a floppy disk drive), when I had previously installed the product's software on drive D. My XT test computer has three floppy disk drives: drive C is a floppy disk drive - the first hard disk is drive D.

Plug-in Card

The stated aim of the card is to check for boot sector viruses before DOS executes, and to prevent DOS from booting if problems are detected. I tested this claim by removing the hardcard from the test computer (in case the card failed to carry out its stated task with 100% veracity), and attempting to boot from floppy disks with infected boot sectors.

I do not have a large collection of boot sector viruses, and was initially only able to test those samples available on 3.5-inch disk (I do not possess a PC which can boot from a 5.25-inch floppy disk drive). This left just four boot sector viruses: Brain, Italian, Monkey and Quox. The card detected only the Brain and Italian viruses, refusing to let *MS-DOS* boot from a floppy disk infected with either of these viruses. When an attempt was made to boot from floppy disks infected by Quox or Monkey, the card did not intervene, and the *MS-DOS* boot sequence commenced as normal.

> *"the card was capable of warning of 'virus-like behaviour' for a random selection of twenty-five different boot sector viruses"*

This poor result left me wondering precisely what the card did, and I therefore tested the product on a machine which had a hard drive installed. The test results were very different: the two previously-missed viruses produced a red warning message which stated that the code exhibited virus-like behaviour. I can only conclude that the card is looking for the presence of a hard drive, and operating differently if one is detected. This needs to be explained in the manual - had my curiosity not overcome me, I could easily have concluded that the card's performance was abysmal. Reviewing anti-virus products is hard enough without leaving such pitfalls for the unwary.

Subsequent testing at the *Virus Bulletin* office showed that the card was capable of warning of 'virus-like behaviour' for a random selection of twenty-five different boot sector viruses. It should be noted that for the majority of these samples, the virus was detected not by name, but by its actions - an impressive result. My one concern is how well the product copes with boot add-ons, like Boot Manager, which carry out rather unusual procedures at boot time. Unfortunately, I had no such software available to me while testing *PC Defender*.

I tried to determine what the *PC Defender* plug-in card would do when various versions of DOS were tested, but it did not balk at any of the versions which were tried. How does *PC Defender* detect boot sector viruses? The documentation claims that it uses 'intelligent algorithms' so that it can detect unknown boot sector viruses. My tests show that the product has some generic boot sector virus detection capabilities, but the manual is worse than useless on this issue. This needs to be improved.

Even with all components of *PC Defender* installed, files could be copied from any boot sector-infected disk at will; i.e. checks for boot sector infection seem only to be made when the PC is booted. Although a boot sector virus infection can only propagate at boot time, spotting such an infected disk at all times would seem to be a good tactic.

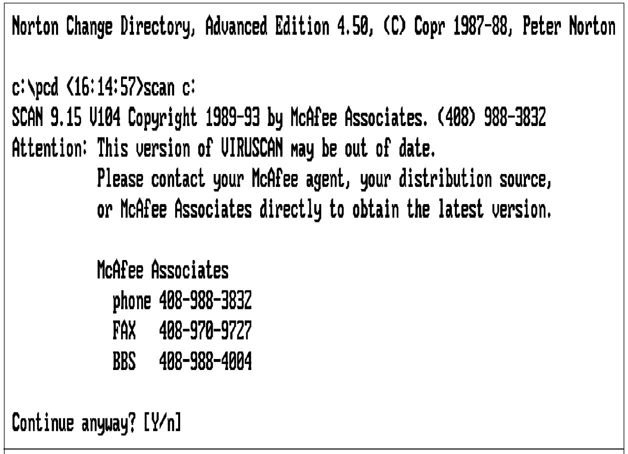
Scanning

The software which comes with *PC Defender* consists of a scanner with a menu-driven front-end program, a memory-resident anti-virus monitor program, software to immunise files, and software to 'clean' (remove) virus infections. Both the scanner and the cleaning program appear to be identical to those offered by *McAfee Associates* (the manual even acknowledges this point).

I am not in favour of immunising files; only the original manufacturer can perform this task reliably. Likewise, 'cleaning' infected files is very much inferior to simple replacement with a known clean copy of the original file. Note that 'cleaning' a file is impossible if the immunisation route has been followed.

I started testing the scanner on the XT computer where the *PC Defender* plug-in card was installed, but soon grew weary of the long times required to scan this old hard disk. I thuerefore installed *PC Defender* on a 486/33, without the plug-in card (because all of the slots in this computer were occupied). Installation proceeded as normal, despite the absence of the plug-in card. The only problem encountered was *Windows'* refusal to execute: it produced a warning message stating, 'unrecognizable disk software installed on this computer … you should run a virus-detection program to make sure there is no virus on your computers'.

Oh, what a wonderful irony that an anti-virus program can cause such a well known piece of software as *Microsoft Windows* to issue this message! The culprit turned out to be the memory-resident anti-virus monitor program, although just why *Windows* was complaining about how this software had changed one or more of the *MS-DOS* interrupt vectors is unclear. The manual is silent on this point.



```
Norton Change Directory, Advanced Edition 4.50, (C) Copr 1987-88, Peter Norton

c:\pcd (16:14:57)scan c:
SCAN 9.15 V104 Copyright 1989-93 by McAfee Associates. (408) 988-3832
Attention: This version of VIRUSCAN may be out of date.
           Please contact your McAfee agent, your distribution source,
           or McAfee Associates directly to obtain the latest version.

           McAfee Associates
             phone 408-988-3832
             FAX   408-970-9727
             BBS   408-988-4004

Continue anyway? [Y/n]
```

When run from the command line the bundled version of *McAfee SCAN* warns the user that it is out of date. No such warning is displayed when it is run via the GUI. This is unforgivable.

By default, *PC Defender* scans only files with COM and EXE extensions, though facilities are provided for users to add other extensions if desired. It is a shame that this facility does not seem to work: no matter how hard I tried, *PC Defender* steadfastly refused to scan all files.

Scanner Speed and Accuracy

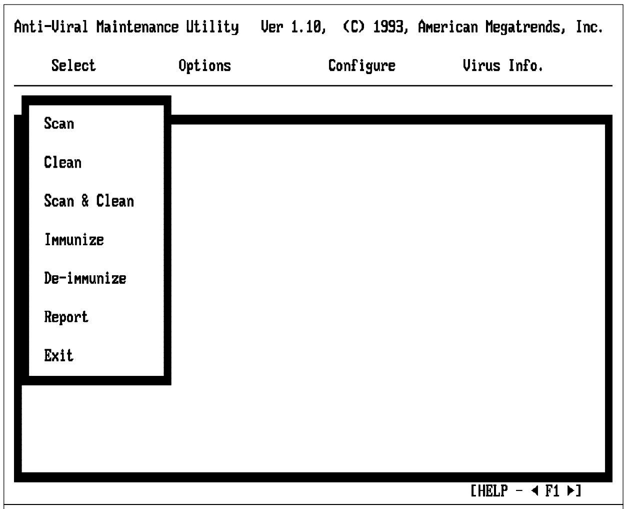
I tested scanning speed on the hard disk of my test PC (299 executable files spread across 11.8 Mbytes), a test which took 1 minute 4 seconds to complete. In comparison, *Dr Solomon's AVTK* scanned the same disk in 20 seconds, and *Sophos' Sweep* took 23 seconds for a quick scan (1 minute 14 seconds for a complete scan).

A scan carried out with the *McAfee* scanner executed directly from the DOS command line produced no degradation in scanning speed, but did cause the scanner to issue a warning that it was out of date and should be upgraded. The front-end menu software provided with *PC Defender* did not show this warning when it executed the *McAfee* scanner - an unforgivable omission. [*AMI claims that VB was shipped an older version of the product, and will look into how this occured. Any user who received an outdated copy of McAfee SCAN can claim a free update via any AMI office. Ed.*]

The scanner detected 229 out of the 239 parasitic test viruses, and 7 of the 9 boot sector test samples. This corresponds to an overall detection rate of 95.2%. All 1024 Mutation Engine samples were detected correctly.

Memory-resident Protection

The memory-resident monitoring utility provided with *PC Defender* comes with its own setup program, permitting ready alteration of options which enable scanning during copying, monitoring of memory-resident programs, scanning on file execution, boot sector writing, device formatting prevention and general write-protection. A configuration file is left by this program in the root of drive C (remember that this caused a problem with the XT test computer: see above).



```
Anti-Viral Maintenance Utility  Ver 1.10,  (C) 1993, American Megatrends, Inc.

    Select        Options          Configure       Virus Info.

  Scan
  Clean
  Scan & Clean
  Immunize
  De-immunize
  Report
  Exit

                                                    [HELP - ◄ F1 ►]
```

PC Defender's strength is its plug-in card. The remainder of the software is based around a behaviour-blocking approach.

I found it annoying to be asked repeatedly during a boot sequence whether a particular piece of software was permitted to become memory-resident, and even more so to be asked by the memory-resident monitor to confirm every file deletion. Although these features of the memory-resident monitor can be disabled (which makes them in effect pointless), neither can be tailored to be more apt for a particular circumstance. The only way to avoid such warnings to to immunise the files - something I do not wish to do.

The overhead which was imposed on PC operation by the memory-resident monitor was measured by copying a large number of small files, both with and without the software present. A set of 42 executable files (1.77 Mbytes) could be copied from one subdirectory to another, with all memory-resident features enabled, in 19.8 seconds. When the scanning during copy feature was disabled, this time fell to 16.5 seconds, a measured overhead of just 16%.

However, when the memory-resident program was disabled (but still present in memory), the time *increased*, to 18.8 seconds. The time to perform the test copy with the memory-resident monitor removed was 16.3 seconds. Something strange is happening here. I was very careful to ensure that the above quoted results are repeatable, but I have no explanation for the anomalous timings.

The detection of viruses by the memory-resident monitor proved to be very poor. During testing, only 64 of the 239 parasitic test viruses were detected: a detection rate of merely 27%. *AMI* explains this result by stating that the philosophy begind the product is that of generic detection. If this is the case it should be explained in the documentation. Notwithstanding, it is no excuse for the poor virus-specific results..

The executable file of the memory-resident anti-virus software contains a list of 41 viruses, and, allowing for some problems with virus naming, all the virus samples detected were on this list. Quite frankly, this pitiful detection rate makes the anti-virus monitor of doubtful usage.

Conclusions

I have mixed feelings about *PC Defender*. I dislike some of the claims and statements made in its documentation. For instance, the statement 'most viruses are meant to cause as much damage as possible to your computer' is simply untrue. Some people would call this sensationalism. In fact, the percentage of viruses which actively cause damage is small, and problems caused by a virus infection come more from poor programming by the virus writer, and unintended side-effects than from deliberate malice. The *PC Defender* documentation also advises users always to 'immunise virus-free executable files'. This is poor advice: stick with an uninfected copy of the original executable file.

These moans are offset by the fact that the card seems to work well: although lacking in virus-specific measures (it identified only a handful of the boot sector viruses by name), the generic detection is excellent. As a protection against boot sector viruses, the product seems very reliable. The parasitic virus detection and prevention is weaker. The out-of-date *McAfee SCAN* and the poor performance of the TSR do not provide the cover needed by a large corporate.

To the best of my knowledge, *PC Defender* is *AMI's* first foray into the world of anti-virus software. As such, the product has some features which show a great deal of promise, but it must be improved if it is to survive in a cut-throat marketplace.

Technical Details

Product: *PC Defender*

Developer: *American Megatrends Inc.*, 6145F Northbelt Parkway, Norcross, GA 30071, USA. Tel. +1 (800) 828 9264 (a US freephone number uncontactable from outside the USA).
BBS +1 (404) 246 8780/1/2/3

Availability: An ISA or EISA computer with one vacant 8-bit or 16-bit expansion slot, 16 Kbytes of ROM space, and a hard disk drive with at least 3 Mbytes of available space, running under DOS v3.1 or higher. Operation under *Windows v3.1* is supported.

Version evaluated: 1.10

Serial number: PCD 001051

Price: £69.00 (quarterly upgrades posted to BBS, ROM upgrades available at nominal cost)

Hardware used: 1. An *ITT XTRA* (an XT clone) with a 4.77 MHz 8086 processor, 640 Kbytes of RAM, one 3.5-inch (720 Kbyte) floppy disk drive, one 5.25-inch (360 Kbyte) floppy disk drive, and a 32 Mbyte hard disk (a plug-in hardcard), running under *MS-DOS v3.30.* 2. A 33 MHz 486 clone with 4 Mbytes of RAM, one 3.5-inch (1.4 Mbyte) floppy disk drive, one 5.25-inch (1.2 Mbyte) floppy disk drive, and a 120 Mbyte hard disk, running under *MS-DOS v5.00.*

Viruses used for testing purposes: This suite of 158 unique viruses (according to the virus naming convention employed by *VB*), spread across 247 individual virus samples, is the current standard test-set. A specific test is also made against 1024 viruses generated by the Mutation Engine (which are particularly difficult to detect with certainty).

For a complete listing of viruses in the test-set used, see *Virus Bulletin*, February 1994, p 23.

BOOK REVIEW

Computer Viruses, Artificial Life and Evolution

Computer Viruses, Artificial Life and Evolution is the latest book from Mark Ludwig, Tucson-based virus writer. Ludwig's earlier book on the subject, *The Little Black Book of Computer Viruses*, evoked the following response from reviewer Richard Jacobs:

> *The Little Black Book of Computer Viruses* is an irresponsible and potentially harmful publication ... Coming from a country where gun control is virtually non-existent, this book might be regarded as relatively innocuous - a fact which will be of little comfort to afflicted computer users.

Containing numerous examples of virus source code, and four complete working viruses, the book caused a wave of protest at its launch. Ludwig's latest book claims to be *The Little Black Book II*, the next in the series. Will it become anything like as infamous as its elder brother?

Justifications

The book begins with a highly pretentious preface, which is followed by an attempt by Ludwig to justify his actions. These arguments look increasingly thinly worn and almost apologetic; daren't he just publish and be damned?

The remainder of the book sets out to help provide some insight into the question of whether computer viruses are alive and can evolve. This subject has been brusquely dismissed by most researchers in the field, but the author actually raises some interesting questions and challenges many of the preconceptions about computers and life.

These rosy words aside, the author of the book appears to have answered these questions in his own mind before committing pen to paper (or, in this computer age, digit to keyboard). Ludwig argues that viruses, although not actually alive, exhibit many of the features necessary for life to be present. So far, so good. However, he then proceeds to twist the most tenuous pieces of evidence to suit his own aims. For example, Ludwig on the subject of viral adaptability (*The Little Black Book II*, pp.43-44):

> Computer viruses have also shown a phenomenal ability to adapt to changes in programming techniques and environments. For example, it is amazing that the Jerusalem virus is still capable of infecting a wide variety of executable files and function properly five years after it was released. Most of the programs it infects today were not even written when it was first released.

This is no argument for adaptability: if a lump hammer which is normally used to smash Brazil nuts is one day brought to bear upon a walnut, is it amazing that it still works? Has the hammer 'adapted'? No - it is just good at breaking things, and it does not know or care what they are. It is a function of how it was created. The same is true of the Jerusalem virus - it was designed to operate in a certain type of environment, on a certain type of program. The fact that it still functions five years later is a testimony to the *MS-DOS* backwards compatibility, rather than any evolution or adaptation on the part of the virus.

The majority of the remainder of the book is written in a similar pseudo-scientific style, with suitably hand-waving descriptions of chaos, evolution and real mutations in viruses. Once again, Ludwig raises interesting points, but completely fails to justify them. This is a great shame, as the subject matter of the book is rich enough to warrant genuine, unbiased treatment.

Virus Code

The virus code supplied with the book is in two appendices. One is directly related to one of the chapters, and contains source code for the 'Darwinian Genetic Mutation Engine' (DGME). This program uses an altered version of the Trident Polymorphic Engine (TPE) to create a virus which can 'evolve' to avoid detection by virus scanners. Although this sounds like a sticky problem for developers, in practice Ludwig's code is not that complex, and viruses utilising the DGME should pose no more problem than the MtE. The book also includes a sample virus which uses the routine.

The second chunk of virus code is source code listings for the winners of Ludwig's 'First International Virus Writing Contest'. Once again, the code is relatively simple, and provides no more of a threat to computer security than any virus exchange BBS.

Having read the book from cover to cover, it seems that the virus code included within it is there simply to generate hype - it could be completely removed from the book without losing any of Ludwig's points. Similarly, the bright yellow Warning banner and text on the back cover appears to have a similar intent. The virus code presents no real challenge to anti-virus software vendors, and the whole feel of these sections is simply that of a marketing exercise.

Conclusion

Stripped of virus code, the book itself is a rather limp, self-satisfied tour through what is a potentially interesting subject. *The Little Black Book of Computer Viruses* caused a tremendous furore when it was launched. *Computer Viruses, Artificial Life and Evolution* is very much a damp squib in comparison. Avoid.

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

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

Tel. 0235 555139, International Tel. +44 235 555139
Fax 0235 559935, International Fax +44 235 559935

US subscriptions only:

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

Tel. +1 203 431 8720, Fax +1 203 431 8165

END NOTES AND NEWS

Heuristics under attack. According to anti-virus software developer *Computer Security Engineers*, a document entitled Anti-TBAV details how to write code which does not raise an alert in packages employing heuristics, is now available to hackers. The file, which claims to be written by a hacker calling himself ‘Köhntark’, gives examples of programming techniques which will circumvent protection. Although specifically targeted at the *ThunderByte* product, the techniques illustrated can be used to evade any scanner which uses heuristic detection.

The *VB 94* Conference will be held on 8-9 September 1994, at the Hôtel de France, Jersey. Tel. +44 (0)235 531889.

Italy and Poland have announced new measures in the fight against computer crime: in Italy, a new law came into force on 14 January 1994, covering damage to public information systems, abusive entrance into protected systems, etc. Transgressors can expect heavy fines and up to eight years imprisonment. In Poland, the *Business Software Alliance* (*BSA*) has announced a media campaign to run in conjunction with a new law which came into effect on 23 February. This law provides for up to five years in prison, heavy fines, and confiscation of equipment.

A **Live Virus Workshop** will be held by *S&S International* on 16-17 May 1994 at the Ashbridge Management College, Berkhamsted, Herts, UK. Tel. +1 442 877877. Fax +1 442 877882.

Sophos is holding two **Computer Virus Workshops** on 18/19 May and 27/28 July, at the *Sophos* training suite in Abindgon, near Oxford. Cost for one day is £295+VAT, and for both days £545+VAT. For further information, contact Karen Richardson. Tel. +44 (0)235 559933

In Germany, **the Neo-Nazis have gone ‘high tech’**, organising and disseminating their news through their own network of BBSs. It is called the *Thule Network* (after the Nazi vanguard of the ’20s), and is run securely, requiring each member to pass certain tests before being granted access to the system.

Hackers are once again on the rampage in the US: their latest point of attack is voice mail. Tactics involve changing messages left on voice mail. Breaches have been detected before too much damage has been done - however, the vulnerability of voice mail has now been demonstrated.

S&S International’s Dr Solomon’s Audit allows network managers to follow every user’s use of application software, and to detect piracy, maximise disk space and standardise software. It will ship this month in two components: the Management Centre will retail at £495, the Scanner for £5 (up to 20 users) or £1 (up to 1000 users) per workstation.

Central Point Software has launched *MacTools 3.0 for Macintosh with Power PC*. The product is claimed to be the first ‘native’ disk utility program available specifically designed to take advantage of the Power PC technology, and among other disk utilities, includes anti-virus software. Tel. +44 (0)81 848 1414.

The *5^0 Congreso Internacional de Seguiridad en Entornos Informaticos* will be held at Palma de Mallorca from 18-20 May 1994. Further information from *Integral*. Tel. +1 971 77 07 37. Fax +1 971 46 40 13.

STOP PRESS:
McAfee agent hacked by employee of *Data Fellows*. According to newspaper reports from Finland, an ex-employee of *Safeco Oy* hacked into BBS and customer systems using a ‘backdoor’ installed when working for the company. To add more confusion to the case, the employee in question is alleged to have been working for *McAfee’s* competitor *Data Fellows*. Commenting on the intrusion, Managing Director of *Safeco*, Hannu Öhrling said ‘If the penetration is connected to the competition in the anti-virus business, which we know to be much overheated, we condemn it strongly. We would not, however, want to believe that even the hardest competition would lead to illegal actions. This penetration is, however, an obvious crime. Whether or not the employer of this person is behind this is very difficult to find out.’ The full story follows next month.