ISSN 0956-9979 OCTOBER 1994

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **To be the best.** This month's edition brings the biggest *Virus Bulletin* comparative review yet of NLM anti-virus software (see pp.13-20). Ten products were tested, and the results may cause something of a surprise…

• **Conference spotlight.** *VB 94* recently took place in Jersey - turn to page 6 for an in-depth report on what happened when.

• **Half way there?** One_Half is a multi-partite virus which uses some of the techniques developed by the Dark Avenger in Commander_Bomber. As if this were not enough, it can also encrypt vital parts of the fixed disk. A detailed analysis is given on page 9.

CONTENTS

EDITORIAL

Live and Let Die

“All this, of course, is based on the axiom that Virus Exchange Bulletin Boards are a Bad Thing”

The *Internet* seems to have become the symbol of the ‘computing revolution’ which we are told is going to change our lives forever. As companies scramble to join the connectivity goldrush, new *Internet* service providers are springing up like mushrooms. Such rapid growth inevitably breeds its own set of problems - use of bandwidth, fights over domain names, and, of course, security: with more users than ever, the *Internet* is impossible to police.

The latest incident picked up by the press concerns the availability of virus code from a user’s account on netcom.com. Although details are deliberately kept vague, suffice it to say that an individual decided to make certain undesirable files available via anonymous ftp from his account. The service provider, *Netcom*, was contacted by a member of the anti-virus community, and refused to take action to prevent this behaviour on the grounds that the user was not breaking any US law.

There is obviously little one can do to prevent the growth of such sites. It is tempting to think that it is the role of the *Internet* service provider to clamp down on any unsavoury activities carried out on its system, but if no law has been breached, it is in a cleft stick. If no action is taken, the company is criticised for ‘supporting’ a virus exchange bulletin board; if the company shuts the account down, it will be seen as attempting to prevent free speech.

All this, of course, is based on the axiom that Virus Exchange Bulletin Boards are a Bad Thing, even if not strictly illegal. While nobody involved in the prevention of viruses wants such sites to exist, there is little evidence of *direct* damage from VXBBS sites. This makes it difficult to legislate against the boards, especially if the appropriate weasel words are added to the login screen.

It should now be clear to anyone in the industry that the prohibition of publicly-accessible virus code is doomed to failure: such sites are here to stay. Even if it is illegal to distribute viruses in certain countries, it will probably always be legal somewhere else - and on a global network, there are no international boundaries. The export restrictions imposed on encryption software by the United States are an excellent example of local legislation being rendered ineffectual by increased connectivity: several different encryption packages are widely available on the *Internet*, and can be downloaded by anyone connected to the network. Attempting to stem this flow of information is rather like attempting to hold back the tide.

Historically, policing of the *Internet* has never really been deemed necessary: the users of the network have to a very large extent maintained order by public pressure. This self-regulation is most often referred to as ‘netiquette’ - the unwritten laws of what is and is not allowed on the *Internet*. Anybody wishing to see this self-policing in action need only post a blatant advertisement to every *Internet* newsgroup. Before embarking on such a study, however, the reader should be aware that the ensuing rush of Email will consist almost entirely of criticism (colloquially known as ‘flames’), and almost no positive response.

Turning the concept of netiquette against the virus authors and distributors will unfortunately not be an easy task. The entire *Internet* culture is one of live and let live - if you keep viruses on your own site, but don’t make wide, unsolicited posts of the material, you are likely to be pretty much left alone by the majority of the community. However, this is primarily an education problem: most *Internet* users do not realise the threat posed by widespread distribution of virus code.

The way forward would therefore seem to be a two-pronged attack. Firstly, whenever the subject of virus exchange is brought up, users should make the antisocial nature of the activity known, until it is seen as a fundamental breach of netiquette. If such a situation could be reached, those wishing to trade viruses over the *Internet* would have a much more difficult time. Secondly, users (especially corporate users) should make it clear to their *Internet* service provider that if it allows virus exchange to take place from any account it runs, business will be taken elsewhere. No service provider, no matter how big or small, will ignore that.

NEWS

Virus Total Reaches 5000

Although the number of known viruses is no longer doubling every nine months, the rate of growth is still high, with approximately 200 new viruses appearing every month. Indeed, some researchers now believe that the total of known viruses has broken the 5000 barrier.

Commenting on the continued rise in virus numbers, *VB* technical editor Fridrik Skulason says, 'After analysing the virus collections brought back home from the *VB* conference in Jersey, I believe that the total number of known viruses is now around 5000. Those viruses belong to approximately 1550 different families, which means that for each new virus written from scratch, there are two others created by modifying an existing one.' This illustrates one of the problems caused by the widespread distribution of virus code: it is very easy to create new variants of an existing virus, simply by patching the binary file. For each of these new variants, a new detection algorithm needs to be developed, adding to the size of scanners.

Although the passing of this milestone is no cause for celebration, it is not as disastrous as it sounds: the number of viruses actually encountered 'in the wild' is growing much more slowly. Despite the 5000 different *IBM PC* viruses now known, the vast majority of all virus outbreaks are still caused by just a handful of extremely common samples ∎

CPAV to Continue

Following *Central Point's* acquisition by *Symantec Corporation* (which produces *Norton Anti-Virus*), decisions have been taken as to the future of *Central Point Anti-Virus* (*CPAV*) in all its manifestations. Product managers for *Norton Anti-Virus* (*NAV*) and *CPAV* go on record as saying that both will continue as separately-developed and marketed products.

Symantec views *CPAV* as a viable product in its own right, and has decided neither to discontinue it, nor to merge it with *Norton AntiVirus*, as many feared would be the case after *Symantec* acquired *Central Point*.

Therese Padilla, product manager for *NAV*, said: '*Symantec* will continue to sell and market both the *Norton AntiVirus* and *CPAV* well into the future. In addition, our development teams are working together on virus techology and working towards a common architecture. Joe Wells is working closely with his counterparts in the *Central Point* division.'

Illustrating the company's commitment to the future of *CPAV* is its latest release, version 2.5, which recently came on to the market. Both of *Symantec's* anti-virus products, *Norton AntiVirus* and *Central Point AntiVirus*, are included in this month's comparative review (see pp.13-20) ∎

Virus Prevalence Table - August 1994

Virus	Incidents	(%) Reports
Form	21	33.9%
AntiEXE.A	6	9.7%
V-Sign	6	9.7%
Stoned	4	6.5%
JackRipper	3	4.8%
Parity Boot	3	4.8%
Keypress	2	3.2%
Monkey	2	3.2%
Tequila	2	3.2%
Viresc	2	3.2%
AntiCMOS	1	1.6%
Cannabis	1	1.6%
EXE_Bug.A	1	1.6%
Italian	1	1.6%
Joshi	1	1.6%
Loren	1	1.6%
Michelangelo	1	1.6%
NoInt	1	1.6%
Nomenklatura	1	1.6%
Spanish Telecom	1	1.6%
Yankee	1	1.6%
Total	62	100%

Total Anonymity

During the closing session of the *Virus Bulletin* conference, one delegate questioned the anonymity of reports made to *New Scotland Yard's Computer Crime Unit*.

However when asked about the policy of the unit, Detective Sergeant Simon Janes was quick to quell such rumours: 'Companies who report a virus attack to the *Computer Crime Unit* are victims of crime, and any information they wish to make available to the Unit is treated with complete confidentiality. The only other body the name of the company is passed on to is the local police service in the area of the complainant. Even in the event of a prosecution of a virus writer, no company would be forced to provide evidence in court.' Janes went on to stress the need for companies to report virus incidents to the unit.

Members of the public should remember that if their computer has been modified by a computer virus, the *CCU* needs an official complaint in order to press charges against a virus writer, should his identity be discovered. Anyone from the UK whose computer has suffered such an attack can contact the *CCU* by telephone on 0171 230 1177 ∎

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 19 September 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

Barrotes.849 — **CR:** Probably an early variant of this family, as it is less complex than some of the others.

```
Barrotes.849        3D00 4B74 03E9 BA01 5053 5152 1E06 5657 2E89 1647 012E 8C1E
```

Bobo.1363 — **CR:** A polymorphic virus which has been found 'in the wild'. No simple search pattern is possible, but scanners allowing for variable-length wildcards should be able to handle it easily.

Burger — **CN:** Two new insignificant variants of this virus, 560.AT and 560.AU, detected with the Burger pattern.

Cascade — **CR:** Three new variants have appeared recently. Two of them, 1701.V and 1704.X, are detected with the Cascade (1) pattern, but the third requires a new search string.

```
Cascade.1701.W      018B 3600 0131 3600 018D BF4D 01BE 8206 313D 3135 474E 75F8
```

Danish_Tiny — There are three new members of this family. Two of them are very similar, 284 and 286 bytes long, but the third is a variant of a virus originally reported under the name Tiny.310.

```
284/286             803D E974 07B4 4FEB DBEB 5690 B800 57CD 2152 518B 5501 8994
310.B               C65B 01B9 1000 D1E9 7301 4E89 F7AD 31D8 ABE2 FA5E 595B 58C3
```

Flagyll — **CR:** These viruses are similar to members of the Lockjaw family: the major difference is that one group overwrites infected files, and the other creates a companion file. The two new viruses are very similar to 318- and 371-byte variants reported earlier, and are possibly created from the same source.

```
Flagyll.316         9C06 1E50 5352 3D00 4B75 03E8 0B00 5A5B 581F 079D 2EFF 2E3C
Flagyll.369         9C06 1E50 5352 3D00 4B75 03E8 0B00 5A5B 581F 079D 2EFF 2E71
```

Genesis — **CN:** A family of four small viruses.

```
Genesis.217         8D96 F701 B800 43CD 2151 33C9 B801 43CD 21B8 023D 8D96 F701
Genesis.226         4E44 74E7 B800 438D 9604 02CD 2151 52B8 0143 5033 C9CD 21B8
Genesis.238         4E44 74E6 B800 438D 9610 02CD 2151 52B8 0143 5033 C9CD 21B8
Genesis.295         4E44 74EB B800 438D 9649 02CD 2172 DF51 52B8 0143 5033 C9CD
```

Green_Caterpillar.1575.I — **CER:** Detected with the Green_Caterpillar (1575) pattern.

Hello — **CN:** An encrypted virus, 600 bytes long.

```
Hello.600           8D76 1E90 E802 00EB 108A 9652 02B9 3402 8BFE AC32 C2AA E2FA
```

HLLO — **EN:** New members of this group of primitive overwriting viruses are 4032.B, 4505, 5760 and Mission.

HLLP — **EN:** The name of parasitic viruses written in a High-Level-Language has been changed from 'HLL.*' to 'HLLP.*'. As with the HLLO viruses, no search patterns will be provided, because of the high risk of false positives. New viruses in this group are 5602.A, 5602.B and 5938.

Intruder.1331 — **CN:** Detected with the Intruder pattern.

IVP — **CEN, CN, ER:** The latest IVP-generated viruses are: 803 (CEN), Angry_Samoans (668, ER), Becky (482, CN), Darlene (632, CEN), DNA (701, CEN) and Roseanne (714, CEN). As in the case of PS-MPC and VCL-generated viruses, no search patterns will be provided.

Jerusalem.Sunday — **CER:** Two minor variants, M and N, detected with the Sunday pattern.

KAOS-B — **CN:** A very minor variant, detected with the KAOS4 pattern.

Kohn_6 — **CN:** Two similar, encrypted viruses, 633 and 638 bytes long.

```
Kohn_6.633          81C7 7802 8B4C 2C8B 072B C189 0743 4303 4C2E 3BDF 7EF1 EB09
Kohn_6.638          7D02 8B4C 3190 8B07 2BC1 8907 4343 034C 3390 3BDF 7EF0 EB0A
```

Komp — **P:** Just like the Kohn_6 virus above, this virus is by Köhntark, the author of the KAOS4 virus.

```
Komp              80BF 0B01 2E75 F8BE 0701 438D BF0B 01B9 0400 FCF3 A4BA 2901
```

Lockjaw — **P:** There are three new additions to this family:

```
Lockjaw.493       9C06 1E50 5352 3D00 4B75 03E8 0E00 5A5B 581F 079D 2EFF 2EED
Lockjaw.507       9C06 1E50 5352 3D00 4B75 03E8 0E00 5A5B 581F 079D 2EFF 2EFB
Lockjaw.894       9C06 1E50 5352 3D00 4B75 03E8 0E00 5A5B 581F 079D 2EFF 2E7E
```

Natas.4746 — **CERM:** Almost identical to the earlier version, but two bytes longer, and apparently not in the wild.

November_17th.768.D — **CER:** Detected with the November_17th pattern. Another new variant, 864 bytes long, is detected with the pattern published for the 855-byte variant.

PS-MPC — **CEN, CER:** As always, there are several new PS-MC viruses: 569.D (CER), Anarchist (524, CEN), G2.615 (CEN), Guten_Tag (665, CEN), Joana.1075 (CEN), Solution (599, CEN) and Skeleton.601.

Semtex — **CR:** Two new variants, one 515 bytes long and detected with the Semtex pattern, and one of 686 bytes.

```
Semtex.686        B850 0733 FFB9 803E F3A4 5E5F 1F07 5A59 5B58 9D2E FF2E C602
```

Small_Comp — **PP:** A family of small, resident companion viruses. Five known variants, ranging from 88-101 bytes long.

```
Small_comp.88     80FC 4B75 3A60 061E BF5C 0157 8BF2 0E07 ACAA 0AC0 75FA B456
Small_comp.92     80FC 4B75 3E60 061E 52BF 6001 578B F20E 07AC AA0A C075 FAB4
Small_comp.100    80FC 4B75 4656 5351 5706 501E 52BF 6801 578B F20E 07AC AA0A
Small_comp.101.A  80FC 4B75 4756 5351 5706 501E 52BF 6901 578B F20E 07AC AA0A
Small_comp.101.B  80FC 4B75 4790 5351 5706 501E 9090 9090 9090 9090 9090 9090
```

Sterculius — **CR:** A family of small, uninteresting viruses, most of which contain the word 'STERCULIUS'.

```
Sterculius.240    BE84 008E D8A5 A5BF E001 83C7 60FA 897C FC89 44FE FB0E 1F0E
Sterculius.266    BE84 008E D8A5 A5BF E001 83C7 61FA 897C FC89 44FE FB0E 1F0E
Sterculius.273    BE84 008E D8A5 A5BF E001 83C7 68FA 897C FC89 44FE FB0E 1F0E
Sterculius.280    BE84 008E D8A5 A5BF E001 83C7 6B90 FA89 7CFC 8944 FEFB 0E1F
Sterculius.428    BE84 008E D8A5 A5BF 7002 FA89 7CFC 8944 FEFB 1F07 2E83 7E7A
Sterculius.440    BE84 008E D8A5 A5BF 7502 FA89 7CFC 8944 FEFB 1F07 2E83 7E7F
```

SVC.1689.F — **CER:** Detected with the SVC pattern.

Tai-Pan — **ER:** This 438-byte virus, also known as 'Whisper', is one of the few which became a 'real' problem recently. Both names are derived from a text string in the virus: '[Whisper presenterar Tai-Pan]'.

```
Tai-Pan           1658 0306 B401 A3AD 00A1 B201 A3AB 0016 582D 1000 8EC0 8ED8
```

Timid.300 — **CN:** Detected with the Timid.305 pattern.

Tiny-family.137 — **CN:** Detected with the Tiny-family pattern.

Tony.203 — **CN:** Detected with the Tony pattern.

Trivial — **CN:** A bunch of small, overwriting viruses, not remarkable in any way and very unlikely to spread.

```
Trivial.23        D6CD 21B8 013D BA9E 00CD 2193 B440 EBEF
Trivial.24        CD21 B43C BA9E 00CD 21B7 4087 D193 EBF7
Trivial.27.D      9E00 CD21 93B4 4087 CACD 21C3 2A2E 2A00
Trivial.29.C      3CCD 2193 B440 5AB1 1DCD 21C3 2A2E 2A00
Trivial.29.D      CD21 87C3 B440 83C2 62CD 21C3 2A2E 632A
Trivial.31.C      212A 2E2A 0087 CAB7 4093 CD21 B44F EBE3
Trivial.32.B      B440 BA00 01B1 20CD 21C3 2A2E 434F 4D00
Trivial.36.A      8BD8 B440 B124 BA00 01CD 21CD 202A 2E63
Trivial.36.B      0001 B440 CD21 B43E CD21 CD20 2A2E 2A00
Trivial.36.C      C262 B440 CD21 B43E CD21 CD20 2A2E 2A00
Trivial.39.C      B440 CD21 B43E CD21 CD20 2A2E 434F 4D00
Trivial.43.D      2193 B440 B12B 9090 BA00 01CD 21B4 3ECD
Trivial.44.E      B440 B12C 9090 BA00 01CD 21B4 3ECD 21B4
Trivial.66        CD21 93B4 40B1 42BA 0001 CD21 B43E CD21
Trivial.89        B43F B901 008D 1659 01CD 21B4 4F80 3E59
Trivial.178       EB1A 90B4 3BBA AA01 CD21 720E EBE7 B44F
Trivial.342       CD21 720F 93B4 40BA 0001 B956 01CD 21B4
Trivial.Ansibomb  721B B802 3DBA 9E00 CD21 93B4 40BA 0001
Trivial.Infernal  CD21 BA8A 01B4 4ECD 2173 09BA 9001 B44E
Trivial.Vootie.B  CD21 B44F EBD4 5DB8 004C CD21 2A2E 2A00
```

VCL — **CN, PN:** New VCL-generated viruses this month are: 433 (companion), 663 (overwriting), 2805, Dominator (925), Genocide (839), Mindless.423.C and Olympic.1442.

Vienna.Ambalama — **CN:** A 493-byte variant with the text '(C) BLACK STAR Inc., 1991. USA,BOX 13263,Ambalama'.

```
Vienna.Ambalama   03C1 8905 B440 8BFA 2BD1 B9ED 01CD 2173 02EB 17B8 0042 33C9
```

CONFERENCE REPORT

VB 94: The Return to Jersey

The small island of Jersey, just off the French coast, was the setting for the fourth annual *Virus Bulletin* conference. This gave a sense of déjà-vu to many delegates and speakers: the *Hôtel de France* in St Helier hosted the first ever *VB* event, in 1991. Participation was up on last year, with over 200 people from near and distant shores representing both the technical and the corporate sides of the anti-virus world.

Conference Overview

Every year, particular themes tend to surface again and again. At *VB 94*, the first, and most often reiterated point, was that computers do not spread viruses; people do. Virtually every speaker pleaded for more user education and awareness - without these, it was argued, there can be little hope of winning the war against viruses.

During and after sessions, much discussion concentrated on the role users could or should play in attempting to discourage virus writers from continuing their pastime. Many industry luminaries said that it was time corporates sent a clear message of 'we don't like what you're doing'.

In the Beginning…

The Wednesday before the start of the conference proper saw an informative and enjoyable discourse on viruses in general, presented by Dr Jan Hruska of *Sophos Plc* and Dr Steve White of the *IBM TJ Watson Research Center*. This double act is rapidly becoming a conference institution, and provides an excellent way for delegates to catch up with the current state of play before the conference begins.

Thursday morning saw delegates creeping into the main auditorium for the opening address, still fuzzy from a Wednesday evening which ended in the 'wee small hours'. *VB* editor Richard Ford, however, soon woke everybody up with his factual and rather depressing assertions that viruses will continue to proliferate, and that virus source code will be more readily available - thanks in no small part to actions such as those of Mark Ludwig and his infamous CD-ROM. Ford's opening address set the tone for the conference: the past year has seen ever more complex virus code, and increasingly bold actions by both virus authors and distributors. 1995 looks likely to provide much more of the same.

Alan Solomon of *S&S International* then took centre stage, regaling the audience with his experiences with virus writers. Delegates learned about the *ARCV* (*Association of Really Cruel Viruses*) and the people behind it, and of Solomon's view that, though such people may exercise their freedom to write viruses, we as users should exercise exactly that same right to try to stop them.

Mechanics and Management

After Solomon's talk, the conference separated into technical and corporate streams. Kicking off the technical stream was Paul Ducklin, of the South African *CSIR* (*Council for Scientific and Industrial Research*). Ducklin, an energetic and entertaining speaker, firmly believes that in many ways the effort to educate made by both the corporates and media has missed its target. He cited the misunderstanding still surrounding viruses such as Stoned as an example of the problem. The virus can be detected and cleaned with standard DOS commands. Why, then, does it still cause so much trouble? His conclusion, reiterated many times over the next two days, was that users must become more aware, and that education must also be directed towards virus authors themselves. It is not enough just to have a well-informed technical support department.

The technical stream continued with a live (and lively!) demonstration of a Virus Exchange Bulletin Board in the USA, by Jeremy Gumbley of *Symbolic*, who accessed a VXBBS to show delegates how easy it is to obtain viruses. Such action is not possible in Italy, where Gumbley lives, as virus transmission is illegal. This is not the case in most of the rest of the world, and Gumbley posed the question of how best to address the issue. During the presentation, Gumbley left a tongue-in-cheek note to the board's SysOp. Interestingly, the account used has since been closed…

While these technical issues were being addressed, Edward Wilding, *VB's* founding editor, now turned hi-tech 'super-sleuth', was directing a presentation to the corporate stream, discussing how best to detect and prevent illegal computing activities. His ultimate recommendation was for the implementation of legal guidelines to assist those encountering the use of computers in criminal and civil cases.

The Big Blues Brothers? Dr Steve White (second from left) and friends model the new look for *IBM's* men in suits.

Winn Schwartau's talk concerned Information Warfare. The growth of the information superhighway, in his opinion, has led to commensurately increased risks, with computers being both the weapons and the targets of those weapons. Schwartau argued for education and protection, a stance which reflects the concerns of many security personnel: with added connectivity comes added risk. Many now feel that the expansion of the *Internet* has been 'too far, too fast'.

Virus, Virus Everywhere

After a hard-earned (and welcomed) lunch break, the conference continued in two sessions, one chaired by Fridrik Skulason, the other by Rod Parkin. Skulason's technical stream opened with an unsettling vision from Vesselin Bontchev: future trends in virus writing.

The 4000+ viruses which exist at present grow by 3-5 daily, stated Bontchev. This poses problems for software developers, who must keep abreast of the epidemic as well as developing such techniques as heuristic and generic detection. Virus authoring packages, virus mutators, and viruses designed to target particular anti-virus products are other problem areas, as are false positives, which Bontchev views to be as problematic as real viruses. The next speaker, Mikko Hyppönen (*Data Fellows*) spoke on retroviruses, the viruses which target anti-virus products. Fortunately, his conclusion was that developers can take many precautions to ensure that their products do not become targets.

An active and vigorous open forum closed the technical stream, with many valid points raised. Bontchev put forward the view that scanners, with the ability to detect only known viruses, are inherently weak. Any scanner can be *made* to look good, asserted a delegate, if the 'right' test-set is used. On the subject of VXBBSs, the worrying scenario of viruses not in the wild being downloaded and released was raised.

Security Measures

The corporate stream, meanwhile, heard talks on principles of computer security (Martin Smith, *Kroll Associates)*, the *DTI* code of practice (Mike Jones, *DTI*), and key controls used to detect viruses (Linda Saxton, *Midland Bank*).

Smith placed responsibility for computer security squarely in the laps of users; a problem with people, not machines. His concluding thought was that 'awareness and training are the food and drink of security'. The following two speakers covered similar ground, illustrating key controls in computer security and virus protection. Ms Saxton summed up the afternoon's assertions in one succinct statement: 'For the future,' she declared, 'better technology may offer partial solutions - but people will decide our fate.'

The Next Instalment

Friday started somewhat later than the first day of the conference - after the late-night gala dinner, most people were pleased to have an extra hour's sleep!

The conference wasn't all play, play, play. Speakers and delegates settle down to some serious arguments about the future of the industry.

The day opened with a stimulating talk in both streams: David Ferbrache spoke on viruses on platforms other than the *IBM PC*; a subject about which, when compared with the PC arena, relatively little is known. However, as Ferbrache said, the first known computer virus in the wild, Elk Cloner, was written not for the PC but the *Apple II*. Threats are inherent in most operating systems: the *Amiga*, the *Atari*, the *Mac* and the *Acorn Archimedes* all have their own viruses. The multi-platform virus, which can be transmitted through different systems, is also a problem facing researchers and developers. Ferbrache's premise is that many techniques seen on the PC can be expected to spread to other platforms, and that invaluable lessons can be learned from such viruses.

Running parallel to this discourse, delegates at the technical stream were participating in one of the most interesting presentations of the conference. The talk, titled 'The generic virus writer', was presented by Sara Gordon, from *Indiana State University*. Gordon has spent many years researching the motivation behind those who write and distribute viruses, and has gathered large amounts of data on the subject, including comments from the Dark Avenger.

She outlined the results of a survey which she had made of virus writers. People from various age groups were polled, with case studies carried out in each area to try to determine common factors. Respondents were overwhelmingly male, the only female respondents being the girlfriend of a virus writer, and a female VXBBS SysOp.

Her conclusions were that, for the most part, virus writers conformed to the ethical norms for their age group. The exception to this generalisation was the adult virus author, stereotypically a loner, concerned with power issues and the injustice of society. Such a person, even if not an expert programmer himself, seems to expend considerable energy encouraging other, usually younger, people to write viruses.

Apart from the adult virus writer, Gordon believes that there is no 'homogeneous group to which the virus writer conforms', and that there are too many observable differences to

Rumours now abound that Alan Solomon and Jan Hruska are in fact twins, accidentally separated at birth. Here they prepare to establish which scanner has the highest hit rate.

be able to categorise them generically. In her opinion, most people become involved in this underworld through simple boredom and peer pressure, and although she conceded that legal means can and should, be used as part of any solution, her view was that enforcement of jurisdiction would prove in many cases virtually impossible - far better, she said, to give young people alternatives to antisocial actions (something which may be easier said than done).

Next, the *ITSEC* certification of anti-virus software, with its goals and achievements to date, was described by Chris Baxter. This is a UK government initiative with an ultimate aim of support and organisation by industry. It plans to evaluate products as a service rather than just software; i.e. as well as testing the effectiveness of the software, the company will be evaluated for its ability to maintain its standard. Areas to assess might include:

- whether the company is tracking the threat closely
- whether the threat is adequately understood
- whether the company responds effectively to the threat.

Tackling the Threat

The afternoon's corporate sessions opened with a presentation from Joe Norman, of *SGS Thomson*, on whether vendors are meeting users' needs. Norman's premise seems already to be becoming 'received wisdom': namely, that server-based anti-virus protection is at least as important as workstation-based measures.

Another highlight of the afternoon was Joe Wells' talk on viruses 'in the wild'. Wells, from *Symantec*, is in contact with many vendors and researchers, and maintains a list of which viruses have been found on users' machines. The result of this work is the 'Wildlist', which allows a user to identify which virus he has, even if the product used to detect it does not use a standard naming convention. One of the spin-offs from Wells' work is that the naming of common viruses is gradually being standardised across competing products.

Automatic extraction of computer virus signatures was the basis for Jeff Kephart's presentation. He and colleagues at *IBM* have developed a statistical method for automatic extraction of 'good' signatures from a virus. His premise is that any automatic task can be done as well by a computer as a person - but that a computer does it more quickly. This method raises the possibility that a computer encountering a previously-unknown virus could develop an 'antibody' to that virus without human intervention, cutting response time to a new virus to hours rather than days or weeks.

The Social Scene

As usual, many delegates came with partners, and the organisers ensured their entertainment while the rest of us worked: on Thursday, a sightseeing tour of Jersey was deemed 'most enjoyable and educational'. Delegates also managed to find 'time to play': Wednesday's informal dinner turned into a festive occasion - not surprising, as people greeted each other in person often for the first time since the last *VB* conference.

The gala dinner on Thursday was enjoyed by all: the theme was the *Blues Brothers*, and the hotel added to the fun by providing the ladies with feather boas; the men with fedoras and dark glasses - a relaxed start which set the tone for the evening. Entertainment throughout dinner was provided by a jazz band and a roving caricaturist [*who did **not** excel at flattering likenesses. Ed*].

Later, guests were regaled by Edwin Heath, the world's foremost hypnotist. Feelings were mixed as to the phenomenon's authenticity, but most agreed that those 'under hypnosis' were affected to some degree. Sadly, all pictures of those hypnotised mysteriously disappeared from the *Virus Bulletin* office before publication…

Thanks and Thoughts

The organisation team, not yet recovered from the exertions of *VB 94*, is already hard at work planning next year's event. Thanks are due to all those who helped with the conference, in particular Karen Richardson, Victoria Lammer, and Rosalyn Rega (*Expotel International Travel*): without their efforts, chaos would undoubtedly have reigned.

Special thanks go to Petra Duffield, the mastermind behind the *Virus Bulletin Conference*, who always ensures that things run flawlessly. Thanks also to the speakers, who gave their expertise and time, and finally, to the delegates themselves, who are the reason for the existence of the conference.

Discussions have already been taking place as to the venue and content of *VB 95*: if readers have suggestions, please contact *VB*; new ideas are always welcome. As for next year - keep your ears to the ground and your fingers at the keyboard; it won't be long before we let you know!

VIRUS ANALYSIS 1

One_Half: The Lieutenant Commander?

Eugene Kaspersky
Kami Associates

Two years ago, a virus appeared which amazed researchers with its infection algorithm. Regular *VB* readers will remember Commander_Bomber (see *VB*, December 1992), which caused numerous problems for researchers by inserting its code into a random location within an infected file. Control does not pass from the beginning of an infected file directly to the main virus body: several blocks of polymorphic code pass control from one part to another, before the main body of the virus is executed.

This means that the standard method of calculating a virus' offset in a file cannot be used, and many anti-virus scanners still do not detect the virus correctly (at least, not when they are run in their default modes).

Now a new virus has appeared - a polymorphic, multipartite sample 'à la Commander Bomber'. Like that virus, One_Half (the name comes from its internal text string, 'Dis is one half') writes polymorphic code into random positions in the file. These 'spots' of code not only pass control to the main virus code, but also contain a loop which decrypts the main body of the virus.

Commander_Bomber is not encrypted, and can be found simply by scanning the whole file. The One_Half virus, on the other hand, is, and cannot be detected using a simple hexadecimal search string. Moreover, the decryption routine is broken up into several pieces, making decryption tricky.

Execution of Infected File

When an infected file is executed, control passes to the decryption code. The decryption loop contains ten blocks of code which are placed at random locations throughout the host file: the first five initialise registers for the decryption loop; the rest decrypts the virus body. Each block contains only one function, on completion of which there is an immediate near JMP to the next block. The last block passes control to the virus' installation routine.

The virus' first action is to issue an 'Are you there?' call (Int 21h with AX=4B53h). If a copy of the virus is already memory-resident, the value 454Bh is returned in AX. If the call is answered, the memory image of the host file is repaired and control passed to it.

If the virus is not already memory-resident, it tunnels the Int 13h vector and reads the MBS to check for the virus' presence, comparing the word at offset 0025h with value 00D3h. If this condition is met, the virus skips the infection routine and returns to the host program. A similar test is made for the value of 072Eh at offset 0180h in the MBS. This part of the boot sector does not contain viral code, and I see no reason for the virus not to infect such disks, unless to prevent conflict with another program. Another possibility is that it might have been used by the virus author to keep his own computer clean during development of the virus.

Next, the virus checks disk parameters, using function Int 13h, AH=08h, and saves the original MBS (and its own unencrypted complete code) in the last eight sectors of track 0. If the disk has been partitioned in the usual way, these are the sectors before drive C's DOS Boot Sector. It then copies 29h bytes of code (which read the virus code from the infected sectors and pass control to the virus) into the original MBS, and writes the MBS back to disk.

"the hooked Int 13h performs two functions ... the first is the trigger routine, the other, the stealth mechanism code"

After hard drive infection, One_Half modifies the Memory Control Blocks (MCBs) in the standard manner, disguises itself as a copy of COMMAND.COM (by copying the 'COMMAND' string into the MCB 'program name' field), and hooks Int 21h. This routine is somewhat unreliable - the virus did not become resident on my test computer during normal operation, functioning correctly only when executed under the control of a debugger.

Finally, the virus restores the infected host program, and passes control to it. If the file is in EXE format, the virus reads the file header and corrects the words to which the Relocation Table points, in addition to returning the decryption blocks to their original form.

The last part of this process is necessary due to the fact that, on infection, the virus overwrites randomly-selected bytes of the host program and may corrupt bytes containing information on the Relocation Table.

Loading from the Hard Drive

When the machine is booted from an infected MBS, the virus' header decreases the size of system memory (offset 0000:0413), copies the virus body into the memory area thus reserved, and passes control to the copy.

The installation routine hooks Int 13h and Int 1Ch, then reads the original MBS and passes control to it. Several other multi-partite viruses use Int 1Ch in a similar manner:

the code checks the Int 21h handler address; if changed (as it will be when DOS is loaded), it saves its current value and points the new Int 21h vector to the virus code.

The hooked Int 13h performs two functions; the first is the trigger routine, the other, the stealth mechanism code. On accessing the infected MBS through READ and WRITE functions (Int 13h, AH=02h,03h), the virus redirects the call to return either the uninfected MBS or a buffer full of zeros.

File Infection

One_Half intercepts a long list of Int 21h functions: the file infection routine is called from the Int 21h handler. On calls to FINDNEXT and FINDFIRST functions (AH=11h, 12h, 4Eh, 4Fh), the virus calls a semi-stealth routine which 'decreases' the apparent file length. On opening, renaming or execution of a file (AX=3D??h, 4B00h, 56??h), the infection routine is called. If a file is created (AH=3Ch, 5Bh), the virus stores its name and infects it when closed.

Before infection, the virus checks the file name, only infecting files with a COM or an EXE extension. Then it checks for the strings SCAN, CLEAN, FINDVIRU, GUARD, NOD, VSAFE, MSAV: if any of these is found, the file will not be infected. The virus looks particularly carefully for the CHKDSK utility and disables the semi-stealth routine during execution of that program, preventing CHKDSK from raising an alarm over lost disk space.

One_Half then checks the file's date and time stamp, which is returned in two registers, CX and DX. The CX register, contains the date stamp (year, month and day); the DX register, the time stamp (hour, minute, seconds). One_Half divides the DX register (time stamp) by 30, and if the result equals the seconds stamp, that file will not be infected. Oddly, one time in 30 the virus does not mark infected files, so it is likely that some files will be multiply-infected.

If the date/time stamp allows infection, the virus executes its polymorphic routine. This selects several random offsets in the file, copies the code from the offsets, overwrites that code with parts of the decryption loop, and encrypts and saves the virus body at the file end. The virus code is at a constant offset from the file end, so a scanner can detect the virus by checking the end of the file, rather than the file header - a useful weak point. Unfortunately, the code is encrypted with a randomly-selected key, and a special routine must be written to 'x-ray' it and catch the virus.

The Trigger Routines

There are two trigger routines: the first is complex, and many attempts to execute it failed. When this routine is called, the virus analyses the size of the DOS primary partition or the extended partition, if present, and encrypts part of the latter with an XOR instruction and a randomly-selected key. The virus decrypts partition sectors 'on-the-fly' before writing or after reading. The partition is available under an infected system, and lost after virus removal. I can just hear the telephone calls to anti-virus vendors: 'Your software disinfected the virus, but we lost all data on the hard drive!'

The second routine is called when the virus is installed in system memory. The virus checks a generation count and tests the system timer value: if these conditions are 'good', the virus displays the message:

```
Dis is one half.
Press any key to continue...
```

and awaits a keystroke. It also contains the internal string 'Did you leave the room ?'

Conclusions

One_Half poses many problems to the developers of anti-virus software. The most pressing of these is the difficulty in removing the virus from infected disks: the usual simple-minded approach of replacing the disk's Master Boot Sector is not enough. This makes it worthy of further attention, and extreme care should be taken when removing it from an infected disk. Any predictions for the next new threat?

One_Half

Aliases: Free Love.

Type: Memory-resident, multi-partite, polymorphic. 3544 bytes long.

Infection: COM and EXE files, MBS of hard drive.

Self-recognition on Disk:
Checks the word at offset 0025h for the value 00D3h.

Self-recognition in Files:
Checks the file date and time stamp.

Self-recognition in Memory:
Via 'Are you there?' call. INT 21h, AX=4B53h returns 454Bh in AX.

Hex Pattern: No search pattern possible in files.
One_Half-infected MBS:

```
33DB FABC 007C 8ED3 FB8E DB83
2E13 0404 B106 CD12 D3E0 BA80
```

One_Half resident in memory:

```
9CFB 80FC 1174 0580 FC12 752F
EB?? 5306 50B4 2FE8 7FFC 58E8
```

Intercepts: Int 13h for stealth and trigger routine, Int 1Ch for installation on loading from infected MBS, Int 21h for infection.

Trigger: Encrypts sectors of the hard drive, displays message.

Removal: Can be difficult, due to encryption of sectors in the DOS partition.

VIRUS ANALYSIS 2

AntiEXE.A - Missing the Target?

Derek Karpinski
Andersen Consulting

On my return from the rigours of the *VB* conference in Jersey, I was confronted with three diskettes infected with a multitude of boot sector viruses - these included Quox, NYB, and NewZealand.I. There was one other virus, which at first appeared to be several: AntiEXE.A, Newbug, D3, CMOS4… any other suggestions for aliases?

Problems are compounded when a virus has so many different names: would anti-virus vendors and researchers please agree on a standard naming convention? It is undoubtedly difficult when a virus appears in many places more or less at once, but such variations serve only to add to confusion in the 'real world'! For now, however, here is the story of one of those diskettes, a virus I call AntiEXE.A.

This boot sector virus has basic stealth capabilities, and infects the Master Boot Sector (MBS) of hard drives. It has been reported in the wild in the UK and in the People's Republic of China. Easy to find and remove, this creature has one interesting feature, which is that the payload specifically targets EXE files meeting certain criteria. The really interesting part is that I do not know which EXE files these are - but more of that later. For now, let us dissect this little beast and see how it ticks.

Action on Booting

The BIOS loads the infected boot sector into memory at 0000:7C00h. The virus examines the interrupt vector table and modifies the Int D3h vector, a handler which is normally unused (apart from by the ROM BASIC interpreter), to point to the original Int 13h diskette Device Service Routine (DSR).

It then builds a stack for its own use, and in time-honoured tradition, subtracts 1 Kilobyte from the amount of memory which will become available to DOS after booting. It installs a replacement Int 13h DSR to do its work, copies itself to the area of memory which will be hidden from DOS, and continues execution from this hidden memory. As an aside, it is interesting to speculate why Stoned, the first virus to infect the MBS, subtracted 2 Kilobytes from memory in a similar fashion. It does not need this amount of memory - perhaps it was just an attempt to arouse less suspicion by using round numbers.

Once the virus has become resident, the original boot sector is read into memory at 0000:7C00h. On a hard drive, this will have been stored at Track 0, Sector 13, Head 0. If booting from a hard drive which is already infected, control passes to the original boot sector, and the boot process continues as usual. Otherwise, the virus reads the MBS from the hard drive to a buffer in hidden memory and checks for the presence of its code. If infection has already taken place, control is passed to the original floppy boot sector, and the machine appears to boot as normal.

As stated, the virus will copy the Master Boot Sector of an uninfected hard disk, and replace it with its own code, copying the partition table data from the original MBS. Control will then pass to the original floppy boot sector, with the outcome that booting from an infected disk or diskette takes a moment longer. However, users are unlikely to notice any extra time spent during the boot process - the success of the Form virus proves this is the case!

Action when Resident

The virus intercepts the Int 13h diskette Device Service Routine. Strangely, the first action of the new DSR is to check for function code F9h, which, if present, it ignores. This may represent an attempt by the virus to avoid detection by, or to subvert, an anti-virus program. If so, either that anti-virus program or the theory being employed by the virus is flawed.

> *"the virus neglects to check whether the area is being used, so loss and corruption of data on floppies may result"*

The virus will normally have the first hook into Int 13h, unless something very strange is going on. If a program added a subsequent hook into Int 13h without using interrupt tunnelling, which can be used to insert a handler into an interrupt handler chain, the program's Int 13h would usually be called first - although this is not always the case. If the call is not for function 2 (Read Disk Sectors), the virus takes no further action. All other function codes result in a call to the original Int 13h routine, which is pointed to by the previously set up Int D3h vector.

On every read, there is a 3 in 256 chance that the payload will be activated: this is based on the current state of the least significant word of the BIOS RAM data area maintained by the system timer at 0000:046Ch. All reads to Track 0, Sector 1, Head 0 call an infection routine which is designed to stealth the original read.

When active, AntiEXE infects diskettes in both the A: and B: drives. The virus performs a series of calculations based on the diskette's Bios Parameter Block, stored in the boot sector, to determine where on this diskette to store the original boot sector. The virus neglects to check whether the area is being used, so loss and corruption of data may result.

The Bios Parameter Block from the target diskette is copied to the viral image and the infection process is completed by writing the virus to the boot sector of the diskette. The next time someone boots from that diskette, the virus will have a chance to spread.

The DSR also checks to see whether the sector being read contains the virus code. If so, the original boot sector is returned instead, successfully spoofing several types of anti-virus or disk-editing software - but the stealthing will not prevent a write to the infected area.

Checking for this virus, and removing it, is therefore best done from a clean boot environment, although this is not strictly necessary. Most anti-virus software vendors should be able to envisage several techniques to do this. I do not advocate this without absolutely precise identification; I have had too many days spoilt by people saying, 'But I disinfected it…'. In my view, the only safe way is for restoration to take place in a known clean environment, from known clean backups or system disks. I do feel disinfectors have a place, however - they help protect against that human tendency to lose source code and master disks. Just check, and check again if you use one.

The Payload

As stated above, the payload is executed 3 in 256 times on any normal read issued via Int 13h. The beginning of every sector read in the call is examined to see whether it contains an EXE header for a file 200,768 bytes long with 3895 relocation items. If so, the image of the EXE file header read is corrupted, meaning that attempts to load the EXE file will fail, and attempts to copy that file will result in corruption.

To try to determine which EXE file was being targeted by this virus, I adopted the 'brute force and ignorance' approach, searching a large body of executable files for matching headers. It failed. If anybody with a large collection of EXE files would like to search them for this file header, I would be pleased to supply a program to do this - it would be nice to know what the author had in mind.

It has been suggested that the target of this virus is a Russian anti-virus program; however, I have not been able to confirm that. If so, the curious way the virus handles function call F9h to Int 13h would be explained. Additionally, if this proves to be true, then AntiEXE.A has spread around the world in an indiscriminate fashion, in search of its target, and will presumably continue to do so.

Detection and Removal

The virus makes itself noticeable through the one-kilobyte memory loss which occurs during booting, in addition to the fact that it intercepts Int 13H, reflecting it to Int D3H.

Removal should follow a cold boot from a known clean system diskette. Although the SYS command will remove the virus from floppies, the original boot sector will remain on the diskette; therefore, the better option would be to reformat it.

Users of DOS 5.0 or later may remove the virus from a hard drive with the FDISK /MBR command (where available). Otherwise, they should copy the original MBS, stored at Track 0, Sector 13, Head 0 back to its correct location at Track 0, Sector 1, Head 0, using a sector editor.

Conclusion

AntiEXE.A is a fairly run-of-the-mill boot sector virus which is spreading in the wild. It is not quite as badly written as some, but is nevertheless easily detected and removed. There is absolutely no excuse for any anti-virus software not to be able to detect it.

The only slightly worrying thing about this virus is that it would be a trivial exercise to modify it to damage commonly found executables. Doubtless researchers merely need to sit back and wait for this new development to happen. AntiWindows, anybody?

AntiEXE.A

Aliases: D3, Newbug, CMOS4.

Type: Memory-resident boot sector virus with stealth capabilities.

Infection: Master Boot Sector of hard drive, boot sector of floppy disk.

Self-recognition in Memory:

None.

Self-recognition on Disk:

Checks for the hex pattern E901h 144Dh at offset 00h of boot sector.

Hex Pattern:

```
33FF 8EDF C416 4C00 8916 4C03
8C06 4E03 FA8E D7BE 007C 8BE6
```

Intercepts: Int 13h. All Read calls (function 2) have a 3 in 256 chance of triggering the payload. All read calls to Track 0, Sector 1, Head 0 are redirected to the infection routine, causing diskettes in drive A or drive B to become infected.

Trigger: Corrupts image of certain EXE files when read.

Removal: Under clean system conditions, use the FDISK /MBR command. For further details, see text.

COMPARATIVE REVIEW

NLMs in Depth

Jonathan Burchell

This month's edition presents a comparative review of ten anti-virus products designed to run on *Novell* servers, providing centralised virus protection from the server. Once loaded, they become an extension of *NetWare*, and are therefore referred to as NLMs (*NetWare* Loadable Modules). As such, they are able to act independently of workstations on the network.

NLMs are a feature of *NetWare 3.x* or above, requiring a minimum of a 386-based server. *Novell* has officially pronounced *NetWare 2.x* dead, which may not be much comfort for those still using this version: although some vendors provide network-non-specific protection, few anti-virus products are designed to support *NetWare 2.x* directly.

NLMs can provide virus detection through two mechanisms: scanning of files for known virus code, and maintaining a list of checksums for files on the server and watching for unexpected alteration to the files. Only scanning will stop an infected executable being placed on the server; checksumming merely detects changes to 'already clean' files.

In view of the absolute necessity of being able to declare files clean before relying solely on a checksumming approach, this review has tested only virus scanning abilities. This is not meant to imply that checksumming is not a valuable and indeed important technique in preventing viral infection - in fact, for a high security environment, it would probably be mandatory to consider using both server-based detection and checksumming.

To Server-base…

Scanning using server-based software rather than a workstation-based scanner has a number of advantages. The single largest benefit is that the NLM runs under *NetWare*, and is thus protected from the activities of stealth viruses.

Additionally, file activity can be intercepted by the NLM, enabling it to scan every file before it is passed to or accepted from a workstation. This should detect infections as soon as they are presented to the file server. Real-time detection is extremely important on networked systems - an infected LOGIN.EXE could spread to every workstation on a network between 9:00 and 9:05 am.

Alongside real-time virus scanning, the other major benefit of server-based anti-virus software is convenience. The software is less limited in the amount of memory it can take up, and can be configured to run automatically with no interaction required from the system administrator (in one case, even updating the virus database by modem!).

Finally, *Novell* file servers support multiple file namespaces and files (*OS/2* and *UNIX* etc.). Software running on the server is able to inspect files in all of these namespaces.

… Or not to Server-base

If the benefits of server-based scanning are obvious, some of the drawbacks are less so. The biggest single problem for server-based scanners is the fact that there are various trade-offs to be made between detection performance and file server load, which is seen by users as an overall decrease in server response time.

Problems exist in both real-time and background scanning. In real-time, files really need to be inspected on the fly (that is, as they are read from or written to the server). There is only so much file buffering that the scanner can afford to do, both from a performance and a resource point of view, which means that, unlike a workstation scanner, the NLM may not see the complete file. This could lead to problems in the accuracy of the detection algorithm, particularly for polymorphic detection.

Additionally, every sector/byte transferred must be checked by the scanner, so an inefficient algorithm will impact greatly on server performance. Background scanners can monitor server load and back off if the server is being heavily used, thus minimising the impact on the workstation users. They do, however, have the ability to see the entire file and sometimes use different detection algorithms from the real-time scanner. Consequently, many products show an improved detection ratio on background scanning.

NLMs may require the updating of various parts of the *NetWare* operating system. If *NetWare 3.12* or *4.0* is installed, there will be little problem. *NetWare 3.10* systems will in general need upgrading to *3.11*; *3.11* systems often need upgrades to specific modules (e.g. CLIB).

Novell documented file I/O interception only in *NetWare 3.12* and later; CLIBs in earlier releases do not contain the documented hooks. Some vendors made the undocumented calls and so can work with almost any version of CLIB, whilst others stick to the *Novell*-documented method, requiring the user to have a later version of CLIB on the system. Many of the products undertake sophisticated server-to-server and server-to-workstation networking, which may also require specific patches.

NLM anti-virus products are plumbed into *NetWare* at a fairly fundamental level. In general, only one task at a time (or several closely co-operating tasks) is allowed to hook such calls, meaning that only anti-virus products from a single vendor can run on the server at any one time. This is a shame, as it gives no chance to 'mix and match' products for better detection, cross-referencing or richer features.

Server-based software uses variable file server memory and resources; as can be seen from the results table, this varies from a few hundred kilobytes to nearly a megabyte. A server may need more memory in order to load, or to gain adequate performance from some of these products.

Full of Fine Features?

Unfortunately, ease of use and configuration is not synonymous with an accurate and secure product. To some extent this simply reflects where the vendor has invested time and effort - either building an extremely good detector, or developing a snazzy user interface. It is up to the reader to decide what trade-off (if any) to allow between ease of use and security.

When selecting a product, apart from its ability to find viruses, the most important areas to consider are administration, scanning options, workstation integration, logging and reporting, and alerts. Each of these aspects is considered in detail below, and listed in the features table on page 19.

• Administration

Once installed, server software needs to be configured and administered. Different vendors have taken different approaches to this problem.

Some products provide control and configuration via standard *NetWare* menuing on the console; using these programs requires *NetWare* console access, either physical (going to the file console) or logical (loading RCONSOLE on a workstation). In general, the user interface and facilities of these programs is limited to the 80x25 text format of the console screen, so there will be no pretty drawings of the network in high resolution graphics.

Others provide programs to run on the workstation which allow complete configuration of the server software. The workstation control programs may be supplied in DOS and/or *MS Windows* format. The latter tends to take full advantage of the GUI environment, sometimes even succeeding in easing the task of administration! One product, the *S&S AVTK*, provides no configuration and setup tools whatsoever, despite the fact that the product itself is highly configurable. The configuration is carried out via a text file which is interpreted by the NLM on start-up.

In an organisation with multiple servers, it can be important to share configuration information (and virus databases) automatically between servers. Again, vendors provide varying support here, from none at all to the ability to group several servers into a single logical administrative domain.

• Scanning Options

Server protection is normally divided into real-time and background scanning options. It is important for both to be able to specify the location and types of files to be checked, and what to do on detection. Most products allow for detected files to be made inaccessible to normal users, deleted, or moved to a quarantine directory.

Real-time scanning would normally be expected to include options to scan incoming and/or outgoing files and selection of file type (all files, or only executables).

	Baseline	Central Point	IBM	InocuLAN	Intel	Net-Prot	Norman	NAV	McAfee	S&S Toolkit	Sophos Sweep
Time 1	236	249	239	227	246	246	245	245	259	217	230
Time 2	240	242	251	222	249	252	225	243	260	222	231
Time 3	239	233	242	224	256	255	227	237	256	209	246
Average	238.3	244.7	244.0	224.0	250.0	251.0	232.3	242.0	259.3	216.0	235.0
%Overhead	N/A	2%	2%	*See Notes	5%	5%	*See Notes	1%	9%	*See Notes	*See Notes
Detection											
Polymorphic Test-set	600	481	500	502	27	462	480	350	80	583	575
Standard Test-set	229	190	227	206	219	214	225	224	223	229	229
In the Wild Test-set	109	78	105	102	92	94	108	68	98	109	109

Table 1: Loading and Detection Results during a background scan. The upper part of the table shows the time above taken to copy a set of test files to the server. Note that the loading results seem illogical. This would appear to be due to changes in the way in which *NetWare* was caching file accesses, and was surprisingly reproducible. Results for real time scanning are given in Table 2.

Background scanning should include options to perform an immediate or a scheduled scan. A great deal of variation exists between the options offered by the product, ranging from only an immediate scan to multiple scans at differing times with sophistication such as 'the first Sunday of the month' being allowed.

• Workstation Integration

Workstation integration takes several forms, and may include the ability to allow only virus-checked stations to log in. Server software will check that the vendor's workstation software is loaded and active on the workstation, or force the workstation to load checking software at login time. Additional facilities may allow collation of status and alert messages from the workstation software in the server logging/reporting database, and the ability to force workstation logout if the system is compromised (no workstation software loaded) or a virus detected as being copied to or from the workstation.

• Logging and Reporting

Scanning and maintaining servers can create a terrific amount of information. Most NLM-based solutions offer the ability to report and log status and detection information into a central log file (which may additionally contain results from workstation and other servers in the same domain). The ability to extract reports from this log file varies greatly between manufacturers, as does the extent to which they are sufficiently documented to allow third party report generators to be written.

• Alerts

When a virus or configuration problem is detected, the user will probably send alert messages both to users and to administrators. The products vary in their ability to configure who gets what message, and how the message is sent. They all support simple *NetWare* message-sending in real time, but some allow for messages to be delayed until a user/administrator logs in. The most sophisticated alert mechanism (*InocuLAN*) also allows for Email, electronic pagers and even fax messages to be sent.

Test Procedures

As mentioned above, this review concentrates on trying to measure the detection ability for each product in both real-time and background scanning modes, together with the amount of server memory consumed by the product. We have also tried to measure the impact on server performance of both real-time and background modes for each product.

No workstation components were loaded unless they were an essential part of achieving a comparison, and products were used 'straight from the box', with settings altered only where absolutely necessary.

A dedicated two-node and one-server thin Ethernet network was built for the tests. The server consisted of a 486DX/25Mhz, 256K local cache and 10MB motherboard memory, an *Adaptec 1542* SCSI Controller, together with a *Fujitsu M2614S* 180 MB drive, and a *DLink DE-220* 16-bit network card and native driver. This gave a *Novell* speed rating of 686. The server was loaded with a 50-user copy of *NetWare 3.11*, which was then patched with CLIB 3.12f, together with the latest version of the IPX and SPX fixes.

One of the two nodes was used solely to load the server; this consisted of a 486DX/25Mhz machine with parallel port adapter. The *Novell*-supplied DO_FILE utility (explained below) was run continuously on this machine. The other node was used to load, administer and test the NLMs. The hardware consisted of a 486DX2/66 MHz (256k cache) PCI based machine, with 32MB of motherboard memory and a 1.6GB local hard disk controlled from the on-board *NCR* SCSI controller. The test machine was loaded with *MSDOS v6.2* and *Windows 3.10*; memory management was provided by *QEMM v7.04*. No disk cache software was loaded.

DO_FILE is supplied by *Novell* as part of its NLM certification tools. It simulates the file-accessing activity of several network users and helps create a typical end-user network environment by continually opening a file, writing to and reading from that file, closing it, deleting it and starting again. Running DO_FILE showed an almost constant 10% loading of the server as measured by the monitor program on the server. However, DO_FILE does not report the throughput it is achieving in a usable manner, so whilst using this provides a realistic user environment, I could not judge to what extent the 'dummy users' were being affected by the NLM under test. I could, however, judge the effect on the test user.

Memory Usage

The amount of memory required by each product was obtained by starting the scanner in immediate scan mode and then using the *Novell* monitor program to view the memory usage of each NLM. As stated, earlier products were used straight from the box, but fine tuning may be possible (for instance not loading *Mac* namespace support on a server with no *Mac* files). The server had a lot of free memory, so products may have been able to allocate a desired maximum of memory to themselves rather than a working minimum.

File I/O Overhead of Real-time Scanning

In an attempt to measure the overhead of real-time scanning, the same procedure was repeated for each product. With the server loaded by the node running DO_FILE, the time taken to copy a test-set of executable files from the test workstation to the server was measured. This was repeated three times with no NLMs loaded, and an average was taken to establish a baseline performance figure. The NLM under investigation was then loaded and the tests repeated.

I tried to ensure that the scanner was set to check only executable files, so that the results were not affected by the scanner having to check the output of DO_FILE constantly

	Baseline	Central Point	IBM	InocuLAN	Intel	Net-Prot	Norman	NAV	McAfee	S&S Toolkit	Sophos Sweep
Time 1	236	240	306	839	828	319	816	518	297	545	2726
Time 2	240	254	420	849	826	336	818	551	288	511	1200
Time 3	239	240	387	858	831	334	828	561	298	511	1210
Average	238.3	244.7	371.0	849.0	828.0	329.7	821.0	543.0	294.3	522.0	*See text
%Overhead	N/A	2%	55%	256%	248%	38%	244%	128%	23%	119%	*See text
Detection											
Polymorphic Test-set	600	462	500	352	27	433	480	350	72	0	575
Standard Test-set	229	137	208	219	219	212	225	224	198	224	229
In the Wild Test-set	109	52	102	98	92	90	108	61	95	98	109

Table 2: Loading and Detection Results during Real-time Scanning. Speed tests are given in the upper part of the table, detection results in the lower. In several cases, products achieve worse detection results in this test than in the background scan. In the case of *Sophos' Sweep* and *S&S International's AVTK for NetWare*, real-time file scanning is provided by a TSR which is loaded on the workstation.

(though every time it created/opened a file, the scanner would have had to confirm that it was not an executable - this should be a minor perturbation).

The test directory consisted of 1024 variously-sized COM files and 47 EXE files, a total of 1071 unique files representing 12,484,818 bytes of data. However, life is much more complicated than this, as the figures need more careful consideration before conclusions are drawn.

The additional complexity arises from the fact that three NLMs (*CPAV*, *IBM AV for NetWare* and *McAfee Netshield*) postpone scanning of files until periods of server inactivity. Thus, it can be a considerable time before a file gets checked. This can present a real problem: for each of these three products, it was possible to pass a virus to the server and copy it to another workstation when the product was loaded and scanning in 'incoming files only' mode. This is a major hole in security, and means that the NLM must be configured to scan both incoming and outgoing files.

Products which move real-time scanning from 'as the file is being accessed' to a later time will score better on overhead than their counterparts; however, a time of reckoning eventually comes when the outstanding list has to be processed (causing a drop in performance unmeasured by our tests) or is sufficiently long that security could be compromised. This warrants further investigation.

Finally, there are special conditions to be considered with *S&S's* and *Sophos'* products which are mentioned in the mini-reviews. Both these products (under normal operation) load the workstation, not the server, and therefore large file accesses from one workstation will not affect the speed of overall network operation.

Detection Tests

Real-time detection tests were measured by copying the virus test-set to the server. The server NLM was set into real-time scan mode and left at the default detection quality (some products can perform either quick or full scans), and the scanner was set to move any infected files into a quarantine directory. Regular readers should note that the Polymorphic Test-set has been expanded to 600 samples: its current contents are Cruncher (25), MtE.Coffeeshop (250), MtE.Groove (250) and Uruguay.4 (75). The remaining test-sets are unchanged (see *VB*, September 1994, p.22 for details of the viruses used).

To measure detection performance of background/immediate scans, the real-time scan option of the NLM under test was disabled, the virus test-set copied to the server, and an immediate scan carried out.

The file I/O overhead of background scanning was measured in exactly the same way as real-time scanning, except that the NLM under test was configured to do an immediate scan of the server, with real-time checking disabled.

Some NLMs allow the amount which they will load the server to be controlled and so would actually give better/worse figures than those presented, which were for the default settings. Some appear to improve performance when compared to the baseline with no NLM loaded. This is a spurious result, and probably due to one of two factors: either the NLM is slowing down DO_FILE and thus the server gets less file I/O from DO_FILE, or it causes *NetWare* (deliberately, or as a side effect) to cache more of DO_FILE's work. A new test procedure to investigate this type of effect more thoroughly is currently being devised.

I also noticed that, if the immediate scan was processing a file which the workstation wanted to access, a 'sharing' error could result. I avoided this problem by waiting until the scanner was in a large test directory (4096 executables) before copying over the test-set. In the real world, there may be some specific issues here: only one product (*IBM Anti-Virus*) made any attempt to deal with this problem.

CPAV for NetWare v2.5

This package consists of three distinct modules; CPAVNET, ALERT and Central Setup. The installation and initial configuration procedure is extremely slick. CPAVNET (which comes in both DOS and *MS Windows* versions) provides the administration and configuration of the server-based detector. As an example of user interface design it is a truly excellent product, and rich in features. CPAVNET allows servers to be configured into logical domains; multiple domains can be administered in a single session.

Central Setup allows individual workstations to be remotely administered by altering the login script for the workstation. It can also ensure that the user is running the specified protection TSRs, updating them if necessary, as well as checking specific configuration issues on the workstation. ALERT receives messages from many of the *Central Point* products and then carries out predefined alert actions. Alert messages are forwarded to users by any combination of Electronic pager, *NetWare* broadcasts, Email or SNMP messages. The package includes a full copy of the *CPAV* for DOS and *Windows* (and also *CPAV* for the *Mac*).

The product achieves extremely low overheads when operating by postponing scanning until the server is not busy. This has a major security impact: with the system set to scan incoming files only, it is possible to copy an infected file to and from the server before the virus is detected. This is a serious flaw, and needs to be addressed.

IBM AV for NetWare Version 1.06

In terms of ease of use and facilities, *IBM Anti-Virus for NetWare* has some definite 'big blue' features. The NLM is entirely configured and controlled via the server console - in fact, via command line options to the NLM. However, the console program allows these to be typed in (literally) and presented to the NLM without unloading it, so there is no menuing; the user merely types the desired commands at the prompt line. The actual range of features available on the NLM places it in the middle of the pack: it has more features than the simple ones, but lacks the sophisticated grouping and centralised administration tools of *Central Point*, *Norton* and *InocuLAN*.

The NLM keeps a list of files waiting to be checked. This can be large: during the real-time I/O overhead test it reached over 1000 items. In tests, with only incoming file scanning enabled, this allowed a virus to be copied on and off the server before the product detected it. Another shortcoming is that the list of infected files is only 250 entries long. Once an infection has been detected, it is entered into the list to await further manual processing. Exceeding this number of infected files causes the data to be lost - this obviously needs fixing.

In use, *IBM Anti-Virus* has a slightly strange feel. I think it appropriate to wait for the full review before commenting further; however, the detection ratios, whilst not the best, show great promise for a 'release one' product.

InocuLAN 3.0

Cheyenne Software, which produces this product, has a long history of producing *NetWare* products - the very successful server backup package *ArcServe* to name just one. The pedigree certainly shows in this product.

The NLM is fully featured, providing almost every configuration option imaginable. The product includes a full set of DOS and *Mac* workstation software and administration tools. It can be administered from the console, DOS or *MS Windows*. The DOS administration tools follow the familiar *Novell* character base menuing system, and will be immediately understood by anyone who has used this kind of tool. The windows GUI is a joy to use and appears to provide the same level of sophistication in terms of server control and domain organisation as products like *CPAV* and *NAV*.

I will wait until we publish a full review of *InocuLAN* before saying much more, but detection results show that it is going to be up there with the leaders. An improvement in detection ratios may well allow this product to be considered as the 'best there is': it has the configuration sophistication of *Central Point* and *Norton*, combined with the potential of good detection results.

Intel Virus Protect Version 2.1

This product has some good features. Installation is relatively straightforward and once installed, the server code can be administered from either a DOS or *Windows* set of tools. The two sets of tools are not one-for-one replacements of each other, but either will work satisfactorily. Separate DOS and *Mac* workstation software is supplied as standard, and includes a stand-alone scanner, and utilities to help ensure that stations logging on to the network have the desired and correct version of the protection TSRs loaded.

Specific support is included for home and nomadic users. The licence allows for home copies of the workstation software, and when a transitory machine reconnects to the network, the VPDOCK program ensures that the local and server signature databases are synchronised, and uploads the results of any scans or incidents from the workstation to the centralised database. Updating the signature and rules database is eased by *Virus Protect's* ability to download new signature updates automatically via modem.

Overall, administration and configuration tools are good, and support for home and mobile users is a definite plus: this, however, is insufficient to make it a viable alternative after the disastrous polymorphic scores are noted. *Intel* claims that these are due to a bug, but the overall detection rate is simply too low to make this an attractive choice.

Net-Prot: F-Prot for NetWare v1.25

Net-Prot is the server-based addition to *F-Prot*, from *Command Software* - a well-known and respected package for DOS, which is included as standard with the NLM.

Net-Prot fits into the class of simple server products, lacking many of the features of its competitors. It is not possible to configure groups of servers into logical domains, so the product is probably most suited to single-server networks. Given its lack of sophisticated messaging and reporting facilities, this is probably desirable in any case. The detection ratios of *Net-Prot*, although good, are not good enough to warrant its use as the only means of network protection. This is counteracted to some extent by the inclusion of *F-Prot*, which provides excellent workstation protection. Oddly, there is no integration between *Net-Prot* and *F-Prot*, and viruses detected by the workstation software do not alert the software running on the network.

Documentation is also extremely disappointing: the printed manual is only a few A5 pages long, containing little of use. On the plus side, *Net-Prot* offered by far the lowest overhead when scanning files in real-time: the products which display a lower overhead in Table 2 do so by storing up 'real-time' checking for later - a procedure which has several security implications. This benefit may well make up for the lacklustre detection results, and make the product worth a second look on a heavily-loaded server.

Norman Firebreak v3.42

Firebreak is the NLM component from *Norman Data Defense Systems*. It is a very simple product, and does not contain many of the 'bells and whistles' provided by its competitors. Configuration is carried out only from the server console and options are limited: for instance, it is possible to scan only those files which match a preset list of extensions. No options are given to alter the extension list or even to control which volumes/directories are to be included or excluded from the scan.

Firebreak offers only real-time and immediate mode scanning; it is not possible to schedule a scan by time or frequency. Multiple servers cannot be dealt with as a single entity. Logging is provided, but facilities to view and report on the log file are not. Virus alerts are limited to *NetWare* messages; however, they are highly configurable, allowing the actual incident message to be completely customised.

At the moment, *Firebreak* lacks so many features that it is difficult to place it. The features which are included are in general well executed, and its detection results are quite respectable. Definitely a product to keep an eye on.

Norton Anti-Virus for NetWare v1.0

Norton Anti-Virus for NetWare has a truly remarkable user interface, allowing for extremely neat centralised administration of a group of servers. However, the *Norton* package includes only server code and workstation administration tools: no workstation-based scanners or protective TSRs are provided. This will obviously add to cost.

As well as server-based scanning, the *Norton* package is able to store a checksum for scanned files which theoretically can detect subsequent alteration to a file. Little data is provided about how this works or how secure it is, but placing this code in an NLM does protect it from stealthing by a virus.

In conclusion, great product, great looks. However, the detection results are poor, mysteriously lower than those printed for the same product in the product's last in-depth review (*Virus Bulletin*, July 94, p.20). A bug, perhaps?

McAfee Netshield 1.5

McAfee products are freely available electronically, although the latest trend by the company has been toward a more formal, boxed version of the software. The copy supplied in this case was the electronic version, which fitted on a single floppy disk and the included barest of 'manuals' (which must be printed out).

The DOS version of *McAfee* (*SCAN*) is a well-known and respected product, in contrast to the NLM version, which has only basic features. For instance, although scheduled scans are supported, only a single scheduled scan can be defined. The installation procedure is somewhat traumatic, involving patching the server (to bring it up to date) and

Table 3: Summary of features included in the Anti-Virus NLMs

	CPAV	IBM AV	InocuLAN	Intel	Net-Prot	Norman	Norton	McAfee	S&S	Sophos
NetWare tested and approved to support	3.11, 3.12, SFT III, 4.01, OS/2 4.01	None	Certified by Novell for v3.x and v4.x	3.11, 3.12, 4.01/binder emulation	None	None	3.11, 3.12, 4.0x	3.11, 3.12, SFT III, 4.0x	3.11, 3.12	3.11, 3.12, SFT III, 4.0x
NetWare versions supported	3.11, 3.12, 4.x, SFTIII	3.11, 3.12, 4.0x	3.11, 3.12, 4.0x	3.11, 3.12 SFTIII, 4.01	3.1, 4.0x	3.11, 3.12, 4.0x	3.11, 3.12, 4.0x	3.x, SFTIII, 4.01	3.11, 3.12, 4.0x	3.11, 3.12, SFT III, 4.0x
Specific 4.0 features	Yes	No	No	No	No	No	No	No	No	No
NameSpace support in box	DOS, MAC, & OS/2	DOS & OS/2	DOS & MAC	DOS, MAC & OS/2	DOS	DOS & OS/2	DOS & MAC	DOS	DOS	DOS
Viruses detected	DOS/MAC	DOS	DOS/MAC	DOS/MAC	DOS	DOS	DOS/MAC	DOS	DOS	DOS
REAL-TIME DETECTION									See review	See review
Executables	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Any File	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Specific Inclusions	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Specific Exclusions	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Processing Delayed	Yes	Yes	Yes	No	No	No	No	Yes	No	No
Immediate Scanning	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SCHEDULED SCANNING						Manual only				
Executables	Yes	Yes	Yes	Yes	Yes	None	Yes	Yes	Yes	Yes
Any file	Yes	Yes	Yes	Yes	Yes	None	Yes	Yes	Yes	Yes
Specific Inclusions	Yes	Yes	Yes	Yes	Yes	None	Yes	Yes	No	Yes
Specific Exclusions	Yes	Yes	Yes	Yes	Yes	None	Yes	Yes	No	Yes
Flexible Schedules	Very	Yes	Yes	Yes	Limited	None	Yes	Some	Yes	Yes
Multiple Schedules	Yes	Yes	Yes	No	No	None	Yes	No	Yes	Yes
ADMINISTRATION										
Console configuration	No	Yes	Yes	No	No	Yes	No	No	No	Yes
Console monitor	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DOS configuration	Yes	No	Yes	Yes	Yes	No	No	No	No	No
MS Windows configuration	Yes	No	Yes	Yes	Yes	No	Yes	No	No	No
Grouping of servers	Yes	No	Yes	Yes	No	No	Yes	No	No	No
Cross-server updates	Yes	No	Yes	Yes	No	No	Yes	Signatures	No	No
MESSAGING AND ALERTS										
NetWare messages	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Email	MHS	No	Yes	MHS	No	No	MHS	No	No	No
SNMP	Yes	No	Yes	No	No	No	No	No	No	No
Pager	Yes	No	Yes	No	No	No	Yes	No	No	No
Fax	No	No	Yes	No	No	No	No	No	No	No
REPORTING & LOG FILES										
Display of log file	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes	No
Filtering of log file	Yes	No	Yes	Yes	No	No	Yes	No	No	No
Server-based checksums	No	No	No	No	No	No	Yes	Yes	No	No
Server-based file repair	No	No	Yes	via DOS	No	No	No	No	Yes	No
WKSTN INTEGRATION										
Login checks	Yes	No	Yes	Yes	No	No	No	No	Yes	Yes
Force logout	Yes	No	Yes	Yes	No	No	Yes	No	Yes	Yes
Centralised messaging	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes
Built-in encyclopaedia	Limited	Yes	No	No	No	Yes	Yes	No	Yes	Yes
WKSTN SOFTWARE IN BOX										
Scanner	Yes	No	Yes	Yes	Yes	No	No	No	Yes	Yes
Checksummer	Yes	No	Yes	No	Yes	No	No	No	Yes	Yes
MAC WKSTN SOFTWARE	Yes	No	Yes	Yes	No	No	No	No	No	No
TOTAL MEMORY REQUIRED	891363	868064	967307	399649	166302	353985	667354	720649	677497	974065

extracting zipped files by hand to a workstation before copying the required NLMs to the server. *McAfee* does at least include the required *Novell* patches in the distribution package, so it is not necessary to download them.

All administration is via the console: workstation-based administration utilities do not exist for this product. Given its simplicity and options, this is a satisfactory solution.

The product's overheads are artificially low when carrying out a 'real-time' scan, as the NLM stores up files to be checked. When both incoming and outgoing file scanning were selected, attempting to copy a file which had yet to be approved by the NLM caused the workstation to pause for several seconds while waiting for the server to catch up with its backlog. However, this is not as serious as the possibility of copying a virus onto the server and off to another workstation when scanning of incoming files only is enabled. This loophole is dangerous, and must be fixed.

S&S International's AVTK for NetWare v6.65

The *Anti-Virus Toolkit for NetWare* is an environment from which a comprehensive server protection system can be built. The server part of the package consists of two NLMs: the scanner and a scheduler. Despite being probably one of the most flexible background scanners, there are no configuration and administration tools in the package (not even an installation routine). Configuration must be carried out via a script language.

Strangely, the NLM only provides background/immediate scanning capabilities: real-time on access capabilities are provided via a TSR, loaded at the workstation. This is completely stand-alone and scans target files when they are accessed. Unfortunately, the TSR performs very poorly when dealing with polymorphic viruses, getting the lowest score out of all the products tested.

As well as lacking configuration and administration tools, the NLM component is limited in messaging and reporting functions; there is no concept of grouping servers into administrative domains. The *AVTK for NetWare* contains a full copy of the excellent *S&S* anti-virus package for DOS and *Windows*. The exceptionally high score in the background scanning means that the *AVTK* is a real contender for 'verifying servers as clean'. However the lack of decent on-access scanning makes it a poor choice for all-round network protection.

Sophos' Sweep v2.64

Sweep is the NLM virus scanner from Oxford-based *Sophos*. The package includes the *Sweep NLM*, the DOS workstation version of *Sweep*, *InterCheck* and the *Sophos Utilities*.

Sweep is configured and controlled from the server console; neither DOS nor *Windows* administration utilities are supplied. The configuration options are extensive and allow for fine control over what is scanned when. In fact, about the only thing missing is the ability to group multiple servers into domains; however, I suspect that this requires a remote administration package as part of the solution.

Only scheduled and immediate scanning is fully server-based: real-time protection is provided by *InterCheck*, a workstation component which maintains a list of certified files. If an attempt is made to access an altered or not-yet-certified file, *InterCheck* ships it off to the *Sweep* NLM on the server for verification. This explains the unusually long time taken by the first pass of the overhead scanning tests: here, every file executed had to be shipped to the server first. In the second and third iterations only the signatures had to be calculated and compared to the database.

In the tests carried out, the overhead was so large that it merited a call to the vendor. It transpired that these tests carried out represented a worst-case scenario for the product, and by installing *SmartDrive* on the workstation and repeating the tests (including the baseline), overheads dropped to 241%. Even this result may be misleading, because the startup time for *Windows* with and without *InterCheck* loaded was almost identical - certainly to within a few seconds. This perhaps merits further testing.

The only weakness in the detection results was missing the Cruncher infections: this was because the NLM was unable to look inside Diet-compressed files. It is understood that this issue will be addressed in the November release.

Conclusions

Even though this is the most comprehensive review of NLM-based products ever carried out by *Virus Bulletin*, it is clear from the test results that it only scratches the surface of all the issues involved. A more comprehensive review is planned (see box below), but it is likely to be beyond the scope of the usual monthly edition of the journal.

These facts notwithstanding, the tests have shown three products which excel in particular areas. *Sophos' Sweep for NetWare* scored the highest detection results in the critical on-access file scanning tests. In terms of real-time file scanning overheads, *Net-Prot* was a clear leader, without resorting to postponed 'real-time' scanning. Finally, in the middle ground between the beautiful and the functional lies *InocuLAN*, which had a highly-usable user interface, and an acceptable level of virus detection. If these figures could be improved, it would be a strong contender for first place.

In view of the highly critical nature of network virus protection via NLMs, *Virus Bulletin* is currently seeking companies which would like to assist in preparing an in-depth analysis of the products currently on the market. For further information on the objectives of such a project, please contact the Editor.
Tel. +44 (0)1235 555139, fax +44 (0)1235 559935.

PRODUCT REVIEW

Virex for the PC

Dr Keith Jackson

Virex for the PC was last reviewed by *VB* in October 1990, when its name was the subtly different *Virex-PC*. Then, it was deemed rather indifferent, and I was interested to see how well the developers have kept up with the increasing pace of the last four years. The package can now also be used on a network; however, this review concentrates on the workstation facilities provided.

Documentation

The A5 manual provided with the product is 101 pages long, and quite well written. It is thoroughly indexed, and provides a reasonable if somewhat terse explanation of how to use *Virex*. It also contains what claims to be a 'Glossary of Terms': in fact, this only defines sixteen technical terms (a hopeless underestimate of the number actually used in the manual), and curiously, it appears in its own right as Chapter 3, rather than as an appendix.

The manual is basically a reproduction of on-screen information, with very little added explanation, excepting 33 pages devoted to a detailed discussion of the various installation methods available. This is excellent, but does highlight the somewhat skimpy content of the rest of the manual. One would expect that sort of detail in all areas.

The developers of this software package persist in using the term 'inoculation' to mean that they are calculating a checksum (or list of checksums), and storing critical information about the file. This will confuse many users: the rest of the computer world defines this term as adding code to an executable program in an attempt to detect and prevent virus attacks, a meaning which closely corresponds to its biological sense. Worse still is the fact that the odd usage of this word is not defined in the manual's Glossary.

The manual claims that 'New viruses appear at a rate of about 20 per month'. However, as the first PC viruses date from 1987, such calculations would account for less than 1000 viruses. This does not fit well with *Virex's* claim of knowledge of 2000 viruses, or with claims made by other vendors of around 5000 viruses.

Installation

Virex was provided as a single 3.5-inch, 720 Kbyte floppy disk - a 5.25-inch, 360 Kbyte, floppy disk is available from the vendors on application, if required. Installation is easy to carry out, and there are several closely-related ways in which the product can be installed (that is, installation must always be to the hard disk; see below). The usual choice between a custom or a predetermined installation was offered. During installation, *Virex* offers to create an 'Emergency Disk' which can be used to 'boot your computer, and to disinfect and restore information if viruses disable your hard disk'.

Installation did present several problems. The program thought that the device drivers used to communicate with my CD-ROM drive via the parallel port of the PC were network drivers, a result I find baffling. After asking the user a few questions, the installation program stated that it was 'Low on memory, running more slowly'. This was despite the fact that it had 630 Kbytes of low memory, and about 3 MBytes of extended memory available! Tests on another machine did not exhibit this problem, and sped the installation time up markedly.

My first attempt at installation was not successful: the program refused to complete its task, failing repeatedly with an error message saying 'Read error, Disk may be bad'. This was untrue, and I eventually worked out that the inoculation program was objecting to a large railway timetable file (over 2 MBytes). The developers claim that this may be due to a bad sector, as *Virex* exits immediately when it encounters a read error. However, *Norton Disk Doctor* did not identify any problems on the disk.

During installation, files are copied to the hard disk, and checksums of executable files are calculated for an 'Integrity database' (their phrase). *Virex* takes an inordinate amount of time to copy what only amounts to a few hundred kilobytes of data across to the hard disk.

The product took an amazing length of time to install: using my *Toshiba* portable (a 16 MHz 386), I got bored after 20 minutes, and went off for a cup of coffee. The product produces a beep every time the screen has filled - apparently this is a bug which the vendors have since fixed.

```
f:\virex (15:39:01)d *.com

 Volume in drive F is STACVOL_001
 Directory of  f:\virex\*.com

virex.com        4807  16/08/94  15:11
      4,807 bytes in 1 file(s)            8,192 bytes allocated
  17,645,568 bytes free
                                                   ╡VPCSCAN 2.94 ╞
f:\vi                       VIRUS ALERT!!
     `PCBB_2(mem)G Virus'  found in memory.
 Volu It was found at memory location 5000:CFB5 (DOS Memory)
 Dire                 (Press a key to continue)

sinstall.exe    48964  16/08/94  15:09
vlist.exe       38055  16/08/94  15:08
vpcscan.exe    269498  16/08/94  15:11
vpcshell.exe    57830  16/08/94  15:11
    414,347 bytes in 4 file(s)           425,984 bytes allocated
  17,645,568 bytes free

f:\virex (15:39:07)virex
Scanning 640K of memory (640K real, 0K Phantom) in system for 2001 viruses.
320K Scanned
```

The biggest problem with *Virex* is encountered during installation, when the product finds virus searchstrings left in memory by its own installation routine.

After this has been completed, all the files on the hard disk show the date/time stamp at installation, not at creation. This is annoying: how can I tell which version of a product has been installed if it keeps changing its date/time stamp?

Such a feature is irritating, but not desperately so. Less forgivable was the fact that when the scanner was executed immediately after being installed, it reported that many viruses were present in memory - I gave up after reading ten error messages. It is obvious that the installation program left its list of virus signatures behind in memory, and did not clear up. Although this error was not repeatable, the developers have confirmed that the order of the patterns found is consistent with *Virex* leaving its own scan strings in memory, although it is supposedly designed to prevent this.

Scanning

When using the product's scanner, the user may inspect a drive, a subdirectory, or an individual file. *Virex* inspects a single file when the memory-resident monitor program detects an unknown executable file (see below). Every time the scanner is invoked, it scans memory, self-checks its own executable file, then checks whatever disk and/or files have been specified by the user. Dynamically compressed files (e.g. those compressed by LZEXE or PKLITE) are uncompressed, and the scanner looks inside the compressed image at the actual executable.

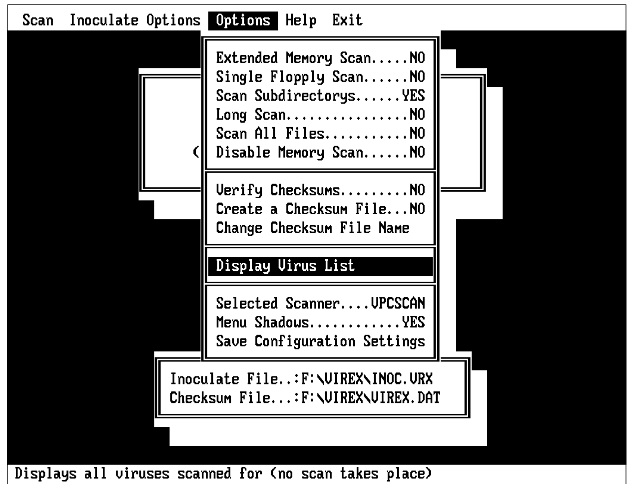
Virex in fast mode took 1 minute 53 seconds to scan my hard disk. With the memory check option disabled, it went down to 1 minute 50 seconds. The product can scan a disk under *Windows*, but does so merely by executing the normal scanner in a DOS box. Using this option, scan time rose to 2 minutes 8 seconds (22% slower than DOS scanning). When carrying out a full scan under DOS, the scan time for the same hard disk rose to 2 minutes 12 seconds.

In comparison, *Dr Solomon's Anti-Virus Toolkit* took 27 seconds to scan the same hard disk; *Sweep* from *Sophos* took 28 seconds in 'Quick' mode, and 1 minute 23 seconds in 'full' mode. No matter how they are interpreted, the above measurements show that *Virex* is not one of the fastest scanners around. Being scrupulously fair, neither does it claim to be.

Accuracy

When last reviewed by *VB*, this product was able to detect just 57 unique viruses. Of these, the product claimed to be capable of repairing 21 - a number small enough for each virus to be described in the documentation individually. The world has moved on since those halcyon days, and the *Virex* manual states that it now has knowledge of 'more than 1700' viruses, a figure which is updated to 2001 viruses by the executable file.

When the scanner was tested against the *VB* test-set in 1990, it detected 77 of the 101 virus test samples. That review concluded that the 77% detection rate could 'perhaps do



Virex is controlled by a series of menus, allowing users to find their way around the scanner quickly and easily.

with being improved somewhat'. The product has indeed moved on, and is now capable of detecting 240 out of the 248 samples contained in the *VB* test-set (see the *Technical Details* section below). This corresponds to a detection rate of 97% - quite acceptable. The only viruses missed were 8888, Suomi, PcVrsDs (2 copies), Pitch, Halley, and Invisible_Man (2 copies). All 500 Mutation Engine-infected test samples were detected correctly.

Memory-resident Monitor

The memory-resident monitor program provided is simple, and occupies merely 528 bytes of RAM. It intervenes only when a program is executed; all other actions carry on as normal. If an attempt is made to execute a program which is not listed in the Integrity Database, whose checksum held in the Integrity Database appears incorrect, or which the scanner thinks the program is infected with a known virus, the memory-resident monitor will intervene and ask the user what action should be taken.

This has the drawback that virus-infected files can be copied at will, and *Virex* will not detect them, making the job of testing its capabilities somewhat difficult. This is balanced by the fact that the overhead imposed by the memory-resident program is very small, as it only intervenes when a program is loaded, and increases the loading time.

If the memory-resident monitor does not find a correct checksum for an executable program, it will simply perform a scan before execution is permitted. Therefore, the virus-specific capability of the TSR can never be better than the intrinsic detection capability of the main scanner.

Testing this memory-resident program was not possible by copying files, so I executed one sample of each virus listed in the *Technical Details* section to see if the memory-resident program prevented their execution. This had to be carried out on a PC with a hard disk, as *Virex* refused to install the memory-resident monitor program when it was unable to access the C: drive.

An uninstalled copy of the product also accessed drive C whenever the memory-resident program was invoked. The test computer locked at this stage. Although this is not how the product is designed to be used, it is still unacceptable that the machine should hang.

During testing, every time the memory-resident monitor failed to prevent a virus-infected file from executing, I was forced to clean-boot the computer - a time-consuming process. Given the memory-resident monitor's mode of operation, it was unsurprising that the list of virus-infected files which were not allowed to execute proved to be identical to the list of infected files the scanner detected.

Each time a file was found which was not present in the Integrity Database (this applied to all the test samples), the memory-resident monitor program produced a box on the screen asking the user to choose whether to add the file to the Integrity Database, abort execution, or merely scan the file before execution. If a file was found infected with a known virus, *Virex* offered options to delete the file, disinfect it, or exit to DOS with no further action taken.

The virus-specific disinfection capability was odd: of 149 virus-infected files executed, 143 were detected as infected, but disinfection was offered for only 14. Rather impressively, the memory-resident monitor program detected that, although it had attempted to disinfect the test file infected with Typo, the attempt had not been successful, and asked the user to confirm deletion of that file.

It is a known fact that not all viruses can be disinfected, but I do not believe that 91% of the viruses listed in the *Technical Details* fall into this category. I am normally exceedingly sceptical, not to say scathing, about disinfection capabilities offered by anti-virus programs - these results serve only to reinforce my long-held belief that disinfection should be approached only with a very long barge pole.

Results were vastly improved when the product attempted to disinfect files which had been 'inoculated' before they were infected. When a change is detected anywhere in a protected file, the file is rescanned, and if no virus is found, the user is offered the chance to update the inoculation records or repair the files. Repair using the inoculation databases was very effective, and in tests, allowed me to append code to target executables, alter the header of the file, and still get *Virex* to carry out a byte-by-byte identical repair. If any changes are made the middle of the file, the TSR detects that the repaired file does not match the checksum of the uninfected file, and prevents the repair process from completing - a vital precaution.

Additional Points

Virex comes with an optional drop-down menu interface which lets a user set options for the scan program interactively. This does not seem to be written by the same authors as the rest of the package, and merits only a four-page explanation in the manual, explaining how to install and execute it. If this program is so new that it is not described in the manual, then the README file ought to contain a decent explanation: it does not. The standard type of drop-down menu structure employed by the program makes it very simple to use, and I found little fault with it. The term 'Single Flopply Scan' used in one of the onscreen drop-down menus caused much amusement (see the screenshot shown on the facing page).

Some of the phrases and messages used by the product are not as clear they might be, and the manual does not have a section which is specifically devoted to explaining all possible error messages. For instance, within the executable image of the scanner, the following message is present: 'Disk out of paper! I kid you not'. The mind boggles…

Conclusions

Virex for the PC is not the ideal choice for the naïve user. Installation is slow and (as described earlier) exhibits several problems, and the scanner's operational speed is certainly not the fastest available. However, it proved quite good at detecting viruses, and the unobtrusive and effective nature of the memory-resident monitor program made it one of the more acceptable examples of this type of program which I have encountered. Unfortunately what has the potential to be a useful weapon in the fight against viruses is marred by the problems with installation and documentation.

Interestingly, my original *VB* review of this product concluded that it was not good at detecting viruses, and that the memory-resident monitor program was very obtrusive. Given that the developers have tackled these problems, it is good to realise that people do read my reviews!

Technical Details

Product: *Virex for the PC*

Vendor: *Datawatch Corporation*, Triangle Software Division, PO Box 13984, Research Triangle Park, NC 27709-3984, USA, Tel. +1 (919) 549-0711, Fax +1 (919) 549-0065, BBS +1 (919)549-0042.

Availability: Any *IBM XT* or above. The operating system must be *PC-DOS*, or *MS-DOS version 3.x* or later. A minimum of 512 Kbytes of RAM is recommended, and a hard disk is required.

Version evaluated: 2.93.

Serial number: None visible.

Price: US$49.95 (US$39.95 direct from *Datawatch.*)

Hardware used: 1. 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*. 2. A *Toshiba 3100SX* laptop portable with a 16MHz 80386SX processor, one 3.5-inch (1.44M) floppy disk drive, and a 40Mbyte hard disk, running under *PC-DOS v6.1*.

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 500 viruses generated by the Mutation Engine (which are particularly difficult to detect with certainty). For a complete listing of all the viruses used in these tests, see *VB*, February 1994, p.23.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

Tel. 01235 555139, International Tel. +44 1235 555139
Fax 01235 559935, International Fax +44 1235 559935
Email virusbtn@vax.ox.ac.uk
CompuServe 100070,1340@compuserve.com

US subscriptions only:

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

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

END NOTES AND NEWS

Global Business Communications, a subsidiary of the American *AT&T Corporation*, has formed an investigative unit to monitor, track and catch hackers 'in the act'. **The unit will profile hacker activity** and initiate 'electronic stakeouts' - the wild west, digital style…

S&S International is again branching out into less virus-specific areas: the company is developing a program, *SmartDesk*, which will operate above *Windows*, **replacing its file and program managers.** It will give users and administrators the ability to configure the desktop, and enable the administrator, from a central console, to reconfigure any machine(s), and monitor network activity. Release is planned for November: further information from *S&S* on +44 (0)1296 318700.

Sea Change Corporation Europe has announced the launch of the *Janus Firewall Server*, purported to be **a completely secure *Internet* server system** which prevents external *Internet* users from accessing information held on an organisation's internal data network. Details from John Coulston, Elvin Turner, or Ruth Johnson of *Sea Change*. Tel. +44 1483 456666, fax +44 (0)1483 456555.

A US federal court has charged six men with computer fraud: the men allegedly hacked into credit reporting service computers, forged purchase orders, and hacked into local companies' voice mail systems. The maximum possible sentence if any of the men are charged on all counts is **50 years in prison and a US$2.25 million fine.**

The proceedings of the fourth annual *Virus Bulletin Conference*, *VB 94*, are now available. Price is £50 + p&p (UK £7; Europe £17; and £25 elsewhere in the world). Contact Victoria Lammer at *Virus Bulletin*: tel. +44 (0)1235 555139, fax +44 (0)1235 559935.

Compsec 94 takes place in **London from 12-14 October 1994.** Further information from Phillipa Orme at *Elsevier Advanced Technology*. Tel. +44 (0)1865 843691, fax +44 (0)1865 843971.

Precise Publishing Ltd has been appointed UK distributor for ***Norman Data Defense Systems'*** anti-virus NLM product, *Firebreak*. Further details can be obtained from Kevin Powys (of *Visionsoft* fame) at *Precise Publishing*. Tel. +44 (0)1384 560527.

A **computer crime ring has been exposed in Scotland:** according to the *Edinburgh Evening News*, the group, containing at least 12 members and based in Lothian, is reported to have netted millions of pounds by breaking into computer systems to redirect bills for using phones and to alter credit card balances.

The latest virus alert from *CYBEC Pty* concerns Tai-Pan, identified as Whisper by some anti-virus programs. It is a memory-resident parasitic EXE file infector with no payload, spreads quickly, gives no obvious signs of infection, and is in the wild in Australasia and Scandinavia. Several vendors have issued updates to detect and in some cases disinfect the virus (see *IBM PC* Virus Updates, p.5).

The 2nd ACM Conference on Computer and Communications Security will be held from 2-4 November 1993 in Fairfax, Virginia, USA. Further details from Ravi Shandu; tel. +1 703 993 1659.

A 20-page brochure about computer viruses has been published under the auspices of *3M* and the *NCSA*. ***How to Avoid Computer Viruses*** uses simple language to explain what a virus is, how viruses spread, how to distinguish a virus from virus-like behaviour, and what to do before and after infection. It is available for US$2 from *3M Virus Brochure*, PO Box 8031, Young-America MN 55551-8031, and through the *NCSA* forum on *CompuServe*, which can be reached by typing GO NCSA at the main *CompuServe* prompt.

The ***European Security Forum Annual Congress*** will take place in Cologne, Germany from 9-11 October 1994.
Tel. +44 (0)171 213 2867, fax +44 (0)171 312 2477.