December 1992 ISSN 0956-9979

VIRUS BULLETIN

THE AUTHORITATIVE INTERNATIONAL PUBLICATION
ON COMPUTER VIRUS PREVENTION,
RECOGNITION AND REMOVAL

Editor: **Edward Wilding** Technical Editor: **Fridrik Skulason** Executive Editor: **Richard Ford**

CONTENTS

EDITORIAL

Terminate-but-Stay-Resident

The new year will see a new editor at the helm of *Virus Bulletin.* My successor, Richard Ford, will inject enormous enthusiasm, energy and new ideas into the journal, ensuring that it remains fit and healthy to report the challenges which will face computer users in the foreseeable future.

Forty-two months have passed since the first edition of *VB* was published. On the day it first rolled from the press in July 1989 it had just *one* subscriber. That first edition reported a manageable fourteen PC viruses and epitomised a time when each new specimen was treated with perverse reverence by a motley crew of fledgling researchers.

In those pioneering, innocent days, certain people held wholly irrational blood convictions. I distinctly remember a belligerent telephone call from an irate member of the public, ‘How dare you publish a magazine about computer viruses when you know full well that no such things exist!’

Virus-specific software, one ‘expert’ assured me, would continue for ever and ever, Amen. Whether he is still saying this, with the number of viruses approaching (or exceeding) 3,000, I do not know. He certainly had not predicted the arrival of self-modifying encryption. ‘Impossible’ said another specialist when the first such specimen, the 1260 virus, appeared two years ago. Certain ‘experts’ arrogantly pronounced that they had predicted such ‘polymorphism’ from the outset, claims which were *patently* untrue.

I have variously been assured that a virus undetectable by a scanner will *never* be developed (Dr Alan Solomon, Washington D.C., June 1992), and that a virus which does not recognise itself on files but which at the same time does not multiply infect them is ‘impossible’ (Dr Jan Hruska, Oxford, November 1992). The second assertion has already been shown to be optimistic (see pages 10-11) while the first is *highly* questionable - as shown in this month’s edition of *VB*, the fundamental methods to evade virus-specific detection are *already* developed, but have simply not [yet] been concatenated.

Indeed, it has been the constant failure of the ‘experts’ to get things right which has been the cause of my continuing astonishment. I remember the terrifying prospect of a ‘Novell virus’ which transpired to be a humble sample of Jerusalem. Then there was the Datacrime II virus which triggered ‘before October 12th of any year’ - how on earth could it spread? A paper by Dr Peter Tippett, which gained short-lived respectability, *completely* missed the point, while our own momentary lapse in vigilance resulted in the publishing of a hex pattern found in COMMAND.COM! As a result of these and countless other *faux pas* I never describe *anybody* these days as an ‘expert’ except in a derogatory context.

The Datacrime fiasco provided an early insight into the dangers of pontificating. It was due to trigger on October 12th, 1989. The world held its breath... and nothing happened, *anywhere*. Perhaps, as Peter Norton had declared, viruses were indeed a ‘myth’ (an excruciatingly embarrassing statement which *Symantec Corporation*, current owner of the *Peter Norton Group*, has subsequently quietly forgotten). The Michelangelo virus, three years later, proved to be less of a damp squib but caused more red faces amongst the pundits, one of whom had predicted that five million computers were afflicted worldwide!

There was *nothing* mythical, however, about certain product developers, who guarded their interests zealously and occasionally resorted to bullying (and even more distasteful tactics) to prevent the publication of poor product reviews. In all such cases, including threats of injunction, litigation and worse, *VB* stuck to its guns and, I believe, our readership is genuinely the wiser for it.

The journal has certainly made its enemies. It has crossed swords with the more unscrupulous product manufacturers, it has caused a number of quack doctors, snake oil salesmen and charlatans to froth at the mouth. Mr Washburn (developer of the aforementioned 1260 virus) has described anyone associated with *VB* as ‘criminal’ while even respected ‘insiders’ have cold-shouldered the journal when it has reported unpalatable truths. *Virus Bulletin* has been described as a ‘rag’, its editor as a ‘rogue’. So be it.

The biggest relief is that amongst so many loud and empty vessels is that there are still a handful of talented and knowledgeable people in this industry whose motivations are driven by more than the accumulation of wealth. I am not in the business of sycophantic back-slapping - the best people know who they are, but a special plaudit must go to Fridrik Skulason for his stoic and supportive work as technical editor, to the editorial board, to the contributors and to the many hundreds of computer users I have had the pleasure to meet in the course of my work on *VB.*

I leave knowing that the journal is in safe hands and that it will continue to report the facts accurately, bravely and without prejudice, at all times aiming to assist its readership tackle an increasingly complex and burdensome problem.

A happy Christmas and a prosperous new year to friends and foes alike.

Edward Wilding
Editor

TECHNICAL NOTES

Proto-T - Grist to the Rumour Mill

The discovery in the wild of a new virus which is not detected by well known anti-virus software products is a genuine cause for alarm. However, discerning fact from fiction is not always easy, and false alarms and rumours are a frequent occurrence. One report which turned out to be completely unfounded concerned the Proto-T virus.

The same report has been faxed to *Virus Bulletin* by readers several times concerning a virus known as Proto-T. The virus claims 'to hide in the RAM of VGA cards, hard disks, and possibly in modem buffers' and users are warned that 'there is no known defence against this virus, save formatting your hard/floppy disks.'

The rumours gained credence when a Norwegian computer magazine reported the virus to be in the wild in Norway where 'it had caused extensive damage'.

The virus which turned out to be the cause of all this panic is a 695 byte parasitic virus which infects COM files. Nowhere within the virus is there any code which refers to disk buffers or hiding in video memory. As for causing extensive damage, as claimed, the virus does have a somewhat unusual trigger. After 16:00, the virus triggers and randomly *reads* data from the hard disk - there is no destructive trigger.

The discrepancy between the virus as described by rumours and the actual code is large, to say the least, and demonstrates the danger of believing unsubstantiated reports from the rumour-mongers.

False Positive Problem For *Sophos*

Sophos Ltd, the producer of *Sweep*, has had several problems reported with the November edition of its DOS virus scanner. Version 2.43 of *Sweep* identifies certain clean files as infected with one of five different viruses: Mutant-1680, The Mutation Engine, Lehigh 2 Trojan, Power Pump and Uruguay 3 [*For a detailed report on the Uruguay 3 virus see p.12. Ed.*].

Problems of this kind are likely to become increasingly common as anti-virus software companies attempt to detect highly polymorphic viruses (the Uruguay 3 virus, for instance, has over $2x10^{144}$ mutations*).

Files giving false alarms include executables compiled with certain versions of *Borland C++* and OEM versions of DOS, thus causing major inconvenience to affected sites.

Dr Peter Lammer, Managing Director of *Sophos*, commented 'We have already taken steps to deal with the false positive problem in the November update of *Sweep.* As part of our impending BS5750 registration we are also introducing enhancements to our quality system, which will help avoid problems like this in the future.'

A contributory factor to the problems encountered was the introduction of a more powerful scanning engine within the software. False positive testing within the company is being overhauled and the number of beta-test sites is being expanded to pre-empt such problems in the future. However, it is arguably preferable to delay the detection of a rare, laboratory virus until a reliable detection mechanism is discovered than to race ahead in an attempt to climb up virus detection league tables.

A Veiled Threat

It is clear that the author of the Commander Bomber virus [*See page 10. Ed.*] is well aware of the threat his latest creation poses to the anti-virus industry. From examination of the code it is evident that the author is simply playing an elaborate game of 'cat and mouse' with the developers of scanning software.

The implications of combining the Mutation Engine (or any other form of polymorphism) with the techniques used in Commander Bomber are far-reaching. Scanners which 'top and tail' files scanned will now be forced to step through the entire file.

Add the Mutation Engine to this poisonous cocktail and the resulting virus, while not *impossible* to detect, could well prove *impractical* to detect due to the sheer volume of processing required to undertake the search. In addition to degrading speed, there is also a larger risk of false alarms unless the search algorithm is chosen with great care.

All of the above would be speculation were it not for the fact that within Commander Bomber there is a bypassed routine which contains the ASCII text 'DAME' (an obvious reference to the Dark Avenger Mutation Engine). The author appears to be inviting the inquisitive hacker to experiment with this combination, and leaves this threat hanging like the Sword of Damocles above users' heads.

*For those unfamiliar with the notation, the number $2x10^{144}$ can also be written:

2,000,000,000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000.

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 26th November 1992. 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 preferably a dedicated scanner which contains a user-updatable pattern library.

Type Codes

C = Infects COM files **E** = Infects EXE files **D** = Infects DOS Boot Sector (logical sector 0 on disk)

M = Infects Master Boot Sector (Track 0, Head 0, Sector 1) **N** = Not memory-resident

R = Memory-resident after infection **P** = Companion virus **L** = Link virus

Known Viruses

5 Lo - ER: 1025 bytes. Awaiting analysis.

```
5 Lo           5256 571E 0680 FC4C 7418 80FC 4B74 1307 1F5F 5E5A 595B 582E
```

_17690 - EN: This virus is also known as Worm-17690, but that name should be avoided, as it is by definition not a 'worm'-type program, but rather a primitive overwiting infector, written in C. The use of a search string to detect the virus is questionable, and the following pattern should be used with care.

```
_17690         2A2E 2A00 2573 2573 5C00 4558 4500 2563 2563 2563 2563 0025
```

Anna - CN: One of the viruses from the British ARCV group. Activates in December and displays a message.

```
Anna           E8FF B440 8B9C 3504 B9E6 028D 940E 01CD 21E8 D6FF E8BE FFC3
```

ARCV-Scythe - CR: A 1208 byte virus, written by the same group or individual as is responsible for Ice-9, Kiss, Scroll and Reaper (Reaper was previously called Apache, but the author of the virus has subsequently changed its name).

```
Scythe         BE?? ??B9 5102 BF?? ??FC AD?? ???? ABE2 F9
```

Deicide-B - CN: A 666 byte overwriting virus based on the original Deicide virus.

```
Deicide-B      3C00 7502 FEC0 FEC0 3C03 7517 B802 00B9 5000 FA99 CD26 FBB4
```

Deicide II-Commentator - CN: Two viruses from the author of the Deicide viruses, 2378 and 257 bytes long. The viruses are not likely to spread as they are extremely obvious, regularly displaying silly screen messages.

```
Commentator    B440 BA00 01B9 EF09 CD21 B457 B001 5A59 CD21 B43E CD21 8B1E
Commentator 2  B440 BA00 01B9 4A09 CD21 B457 B001 5A59 CD21 B43E CD21 8B1E
```

Dutch Tiny-117 - CN: A small virus, 117 bytes, which does nothing but replicate.

```
Dutch-117      93B4 3FCD 2180 3C4D 7428 B002 E82B 0097 B175 B440 CD21 B000
```

Gotcha-Legalize - CER: A 1781 byte member of the Gotcha family - detected with the Gotcha-D search pattern.

Ha! - CER: This 1456 byte virus is probably of Polish origin. It may display the word 'ha!' in large letters on the screen.

```
Ha!            BB0D 00B9 5A05 2E2E 8037 ??43 E2F9
```

Jerusalem II-1663 - CER: This virus is obviously based on the Jerusalem virus, but differs considerably in one important aspect - it appends its code to COM files, instead of prepending it. As a result it might best be classified as a member of a different family, but currently there is no agreement as to its classification. The virus is 1663 bytes long.

```
Jerusalem II   2638 05E0 F98B D783 C203 061F 0E07 B800 4B2E 8C1E 4200 9C2E
```

Keypress-Chaos - CER: This 1236 byte virus seems to be based on the Keypress virus (also known as Turku), but has been modified.

```
Keypress-Chaos 5351 521E 0656 0E1F C706 F005 0000 833E F205 0275 3B33 C0CD
```

Kode 4 - CN: 399 bytes. Awaiting analysis.

```
Kode 4         803D E975 0D8B 4D01 2D8F 013B C175 03EB 6390 33C9 33D2 B800
```

Leprosy-8101 - CEN: An overwriting virus which is unlikely to spread.

```
  Leprosy-8101      59B8 0100 EB00 5F5E 5DC3 558B ECA1 E61B 051E 008B D033 C9B0
```

Leprosy-Silver Dollar 736 - CEN: Yet another variant of this primitive overwriting virus. It is 736 bytes long and is detected with the Silver Dollar pattern.

Leprosy-Seneca - CEN: An encrypted, overwriting 493 byte virus, which seems to be derived from the Leprosy virus. It appears to be written in C.

```
  Seneca            9090 90E8 0300 EB2B 90BB 3301 B9BD 0190 8A27 80F4 FF88 2743
```

Little Brother-321 - P: 321 and 349 byte long companion viruses which are related to the variants reported earlier.

```
  Little Bro-321    9C52 5350 1E06 3D00 4B75 03E8 0B00 071F 585B 5A9D 2EFF 2E41
  Little Bro-349    9C06 1E50 5352 3D00 4B75 03E8 0B00 5A5B 581F 079D 2EFF 2E5D
```

Multi - CER: A 2560 byte virus which occasionally ‘drops’ one of two small viruses which are related to the ‘Russian Tiny’ virus.

```
  Multi             A308 0033 FFB8 00B1 8EC0 2689 0526 3905 740D B800 B98E C026
```

Ondra - EN: An overwriting virus, written in compiled BASIC or some other high level language. Unremarkable and very unlikely to spread.

```
  Ondra             004E 0089 EC5D C208 0003 2A2E 2A01 5C0B 2626 6F6E 6472 612E
```

Proto-T - CR: This 695 byte virus was described as being remarkable in several ways, but in fact it is singularly uninteresting. It activates after 4 pm - reading random sectors from the hard disk. Civil War II is a 580 byte related virus.

```
  Civil War II      80FC A075 05B8 0100 9DCF 1E06 5756 5053 5152 3D00 4B75 0D2E
  Proto-T           80FC A075 05B8 0100 9DCF 1E06 5756 5053 5152 80FC 4075 0583
```

PS-MPC Abracas - CEN: This 546 byte virus may multiply infect files. It is detected with the standard PS-MPC pattern.

PS-MPC McWhale - CEN: An 1125 byte virus, generated by the PS-MPC program. When it activates it scrolls a message containing slander about John McAfee across the screen. The virus can be detected with the standard PS-MPC pattern.

PS-MPC Mimic - EN: This 2832 byte virus is detected by the PS-MPC pattern. Awaiting analysis.

PS-MPC Z-10 - EN: A 702 byte virus, generated by the PS-MPC toolkit, and detected in the same way as other PS-MPC viruses.

SillyC-207, _207 - CN: A simple, 207 byte virus, which does nothing but replicate.

```
  SillyC-207        3D00 0059 7503 5FEB 3D33 C0AE 75DE E2DE 2BFA B801 4233 D2CD
```

Trivial-37 - CN: A primitive overwriting virus.

```
  Triv-37           B44E BA1F 01CD 2172 14B4 3D40 BA9E 00CD 21B7 4093 BA00 01B1
```

Trivial-42B - CN: A tiny and unremarkable overwriting virus.

```
  Triv-42           40BA 0001 93B1 2ACD 21B4 3ECD 21B4 4F
```

Todor - CER: This 1993 byte Bulgarian virus uses variable encryption and cannot be detected with a simple pattern.

VCL - CN: Two new VCL - generated viruses have been reported this month. The viruses, both called Diarrhoea, are 1222 and 933 bytes long. They are detected with the VCL-1 and VCL-2 patterns.

VCL-Heevahava: This 514 byte virus is generated by the VCL program, but unlike most of the other VCL viruses it is not encrypted. It contains the text ‘Only heeva-hava’s get stuck with THE HEEVAHAVA virus!’ It is a companion virus, similar to the ‘Pearl Harbour’ sample included with the VCL package.

```
  VCL-Heevahava     51E8 7400 59E2 F9BA 6400 06B8 4000 8EC0 2689 1613 0007 B905
```

Vienna-600 - CN: Yet another Vienna variant - what more is there to say?

```
  Vienna-600        8E1E 2C00 8B76 8F8B 7E8B AC3C 3B74 093C 0074 03AA EBF4 33F6
```

Witch - ER: An encrypted 1140 byte virus which uses a debugger trap which modifies the stack pointer in the decryption loop, probably in order to prevent single-stepping through the code.

```
  Witch             60B9 2403 BA?? ??FA 81EC 4806 4444 5833 C283 C27F 50E2 F5FB
```

Yankee-XPEH 5648 and **5808** - CER: Two Russian variants, 5648 and 5808 bytes long. They are related to the other members of the XPEH group. Awaiting analysis.

```
  XPEH-5648         C602 E2F3 C3BE D109 03F3 B955 002E 8B97 2F00 E8E2 FFE9 2EFF
  XPEH-5808         C602 E2F3 C3BE 9609 03F3 B94F 002E 8B97 2F00 E8E2 FFE9 3BFF
```

INSIGHT

Mark Hamilton

The Scanner To End All Scanners?

If you take the Metro-North Commuter train from Grand Central Station in Manhattan, you will arrive some forty minutes later in Hawthorne, a sleepy hamlet set in rural upper New York State. Hawthorne itself is unremarkable except that it houses the *Thomas J. Watson Research Laboratory*, one of the four *IBM* research facilities that are dotted around Westchester County. This is where *IBM* conducts its research into computer viruses and where I met Steve White, who manages this aspect of the research at the *High Integrity Computing Laboratory*.

Traditionally *IBM* has tended to hide its light under a bushel - a truism that extends throughout the company. However, with regard to the company's anti-virus effort this attitude may be about to change.

Uncharted Waters

'Last Monday [2nd November], we announced *IBM AntiVirus/DOS* and *IBM AntiVirus/2* in both the United States and in Holland', White told me. 'These products completely replace our previous scanner [*VirScan*] which was made available on a very much ad-hoc basis', he continued. 'Unlike all other *IBM* products, these are developed and marketed by *IBM Research* - we're breaking new ground here.'

He believes that these two products, one for DOS and *Windows,* and the other for *OS/2*, will prove successful in an already cut-throat marketplace. They have been designed as 'install and forget' products which are not meant to be run as applications like so many of their competitors. This does not mean to say that they do not have pretty CUA (Common User Access) keyboard-and-mouse driven front-ends - they all do. The difference here is that the user should never *need* to use them.

'In addition to scanning files for viruses and virus-like behaviour, our anti-virus products can also disinfect in memory, so infections are less likely to spread', White said. He went on to explain that since less than ten percent of all known viruses have ever been discovered on machines 'in the real world', *IBM's* product has been designed to identify and disinfect the one hundred or so viruses that are at large: 'We can detect a much larger number than that, but since they aren't at large, we haven't written any disinfection capabilities for those', he explained.

Tucked away in sleepy Hawthorne is the T. J. Watson Research Center - a hive of anti-virus research.

White then posed the key question. 'If we write a program that won't catch every single virus, are we doomed?'. He doesn't think so. 'We know from research we've carried out that there is an epidemic threshold below which viruses go extinct. It doesn't take much effort to make viruses extinct!'

Affordable Protection

How much is this scanner? In the United States it is the astonishing price of $29.95. Simply on cost alone this therefore makes *IBM's AntiVirus* program one of the most competitive on the market. There is, however, an even more compelling reason for predicting its success.

'Nobody ever got fired for buying *IBM*' is a familiar maxim, and it is easy to see how trusting one's data to *IBM* is an appealing idea. *IBM's* image in the marketplace, and its sheer size, also help inspire confidence - a vital commodity when selling anti-virus software.

In addition to this, *AntiVirus* is the product of over *five years* of research and development. The cynical observer may even go so far as to suggest that *IBM's VirScan* was simply a massive beta test of the principles which underlie the new software. Regardless, *IBM's* expertise is undisputed, and there is certainly no need to worry about whether the company will still be there tomorrow - *IBM* is unlikely to be on *Symantec's* list of prospective take-over options!

IBM means business, and this scanner will have a tremendous impact on the industry.

Automated Signature Extraction

IBM has invested considerable resources into investigating and monitoring viruses at its *High Integrity Computing*

Laboratory and has manpower and equipment that is matchless in this industry.

‘We don’t disassemble every single virus these days - that is too costly. We are only interested in those that pose a significant threat, those we *do* examine closely are those that we need to be able to disinfect’, White explained. ‘We have written a suite of programs that examine infected files and extract a signature for the infecting virus. These signatures are then run across a huge library of clean programs and any false positive signatures are eliminated.’

White’s team constantly hones the auto-extraction software to such an extent that most of their signatures are extracted automatically. ‘Don’t forget that we’ve got over a quarter of a million PCs within *IBM* - that’s a huge beta test site!’

No Windows Here...

In an electronically sealed laboratory across the corridor from White’s office, banks of PCs sit whirring away, twenty-four hours a day, seven days a week. Some are extracting identification signatures from the latest batch of new viruses while others are busy testing other companies’ anti-virus products whose manuals line the shelves.

These PCs are interconnected by means of a self-contained local area network to a secure server. Apart from a telephone and a door, there is no connection to the outside world - there are not even any windows [*but presumably plenty of OS/2. Ed*]. ‘Each PC has its own boot diskette and when it is rebooted, the PC automatically and completely disinfects itself and loads a new hard disk image from the server’, commented White.

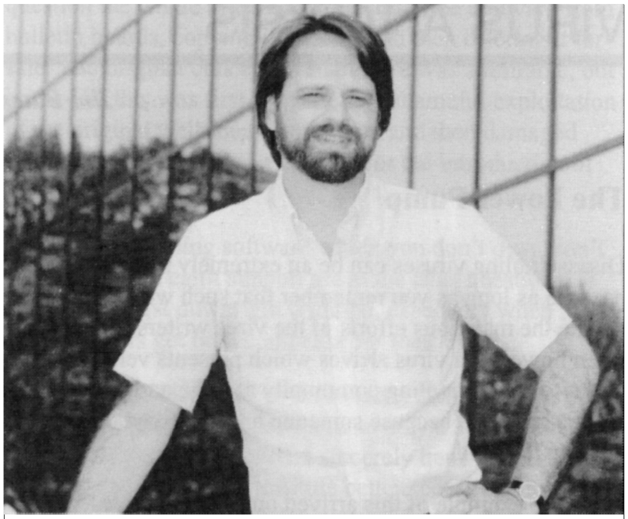
White, originally from California, began his career as an electronics design engineer working in chip design and later became fascinated with computer viruses.

Another who made the vocational switch is Jeffrey Kephart, though in his case, the transition seems more logical. Kephart is an epidemiologist who researched the spread of biological viruses and he now brings his training and experience to bear in the field of computer viruses.

Biological Parallels

Kephart believes that computer viruses spread in the same way as human ones and once you produce the conditions that reduce the spread of infection, the virus will die out naturally: ‘The critical point is called the epidemic threshold. Reduce virus numbers below that point and they die out naturally; above that point a virus thrives and spreads.’

By using anti-virus viruses, Kephart believes that new outbreaks could quickly be reduced to below the epidemic threshold. He showed me video-taped simulations of viruses being tackled first by conventional means and then by the use of anti-virus viruses. The results were impressive and he may well have a point. However, the idea of an anti-virus virus is controversial, and a massive amount of research would need to be done before releasing one in the wild. These reasons effectively preclude anti-virus viruses for the foreseeable future.



White: ‘Don’t forget that we’ve got over a quarter of a million PCs within *IBM* - that’s a huge beta test site!’

Kephart and White have jointly authored a number of research papers including *Measuring Computer Virus Prevalence* in which they reassure us that computer virus incidents currently occur at the rate of one incident per thousand PCs per quarter and that these are caused by less than 9% of all known computer viruses.

The Way Ahead

As to the future? As communications links and networks become increasingly sophisticated ‘we all will need to be aware of active mail agents and build safeguards against their misuse’, White believes. Meanwhile, he and his team will continue to monitor virus attacks, develop detectors and antidotes and educate those inside and outside *IBM* - all from the seclusion of their lab which, until now, has been hidden from the public gaze.

[*The products will be available initially within the United States and The Netherlands. IBM UK will evaluate the software before deciding whether or not to sell and support it. VB hopes to have IBM UK’s decision in time for the review of these products in next month’s edition. Ed.*]

VIRUS ANALYSIS 1

Jim Bates

The Power Pump

Disassembling viruses can be an extremely rewarding pastime as long as you remember that such work helps to negate the malicious efforts of the virus writers. Every so often however, a virus arrives which presents very little threat to the computing community at large and yet must still be analysed because someone has contrived to get it out into the wild.

A classic example of this arrived on my desk recently and analysis of the code and the circumstances of its distribution raise some disturbing questions. The virus is called (from a message within it) ‘Power Pump v1.2 Virus - Silent But Dead.’ and qualifies as probably the most maladroit chunk of code in this whole sorry saga so far. Quite apart from being so riddled with bugs that it hangs more often than it runs, the whole concept represents a ludicrous attempt to produce self-replicating code using the companion virus idea in a novel way.

If the average virus writer is postulated as an adolescent teenager, I’d guess that the author of Power Pump must be a mentally deficient 10 year old working after school using an out of date copy of *The Beginners Book of Computing* and an old copy of Turbo ‘C’ with several pages missing from the manual.

Origin Of The Species

The code was sent to me by a user who bought some disks from a UK shareware vendor (*Transend Services Ltd* in Keighley) and noted some strange effects on his machine when he ran one of the programs. The program in question was an interrogative database called IQTEST which purported to determine the user’s Intelligence Quotient in response to the answers to around 150 questions.

This is reminiscent of the AIDS Disk incident in which an interactive database asked questions concerning the user’s sexual/drug habits and from the answers calculated a risk factor. In that case the program was accompanied by a damaging Trojan within the installation routine and money was sought to provide a cure - with Power Pump, the programs are accompanied by a clumsy companion virus which is in two parts quite distinct from the actual program files. The parallel of an apparently attractive program, freely available but with an accompanying ‘nasty’ provides plenty of food for thought.

Operation

I understand that the files were distributed as freeware in a self-extracting archive called UNPACK.EXE. This is quite harmless and when executed, the files are unpacked and written to disk. One of the files is called READ.ME and contains the following text:

```
IQTEST version 1.01 01/01/92
IQTEST is an intelligence quotient test and
analysis

contents  ------------------------------------------

POWER.EXE for use by IQTEST only
IQTEST.EXE asks you 160 questions & prints an
analysis
IQTEST.COM determines computer configuration
IQ.DOC text file containing analysis
IQTEST.DOC text file with 160 questions
READ.ME this file

IQTEST is freeware. Have fun!
```

All of these files are dated 1/1/92. The actual program and its associated data files are harmless, but note that there is a file called IQTEST.COM as well as IQTEST.EXE. When a COM and EXE file of the same name exist in the same directory, DOS will execute the COM file rather than the EXE file - this is how companion viruses work. With this virus however, there is a slight variation on the theme - the COM file is the companion, but it is not the virus. It is a batch file which has been compiled with Douglas Boling’s excellent program BAT2EXEC (v1.5), which converts batch files into true executable code in order to improve their speed of execution.

New Additions

The compiled batch file executes the POWER.EXE program which actually does the infection. Once POWER.EXE has run, it returns control via the batch file to the EXE program of the same name. In the original case, this meant that typing IQTEST ran first the IQTEST.COM file, then the POWER.EXE file and finally the IQTEST.EXE file which starts the interactive database.

POWER.EXE checks its location on the disk and if it isn’t in either C:\ (the root directory) or C:\DOS, it copies itself there and deletes the original. It then tests to see whether write permission is available for one of the four drives A:, B:, C: or D: (at random) and if so it searches the directory structure of the disk for an EXE file which does not have a corresponding COM file in existence alongside it (again at random). When a suitable file is found it copies the compiled batch file IQTEST.COM to the selected directory under the same name as the target program file but with a

COM extension. This new file is then hidden (by setting the HIDDEN attribute) and if the target was a floppy disk, the POWER.EXE program is also copied to the root directory. This file is not hidden, presumably under the impression that the average user will not notice the new addition to the family! At least, that’s the plan...

> *“Leaving such vital information in a file... is like leaving a bomb somewhere with its wiring diagram attached to the outside!”*

Helpful Hints

I try to remain reasonably optimistic about the virus problem although I do find the sheer volume of new viruses a little depressing. Every so often however, something occurs to lighten my day a little and the arrival of Power Pump was one such occasion.

The POWER.EXE program was written in ‘C’ and this could have meant that I needed to set up a fairly complex set of software tools to undertake the disassembly process. Not in this case though - the writer thoughtfully left all the symbolic debug information in the file during compilation and this made me smile and gently shake my head. Leaving such vital information in a file which is bound to be investigated, is like leaving a bomb somewhere with its wiring diagram attached to the outside!

The mistake of leaving the debug information in the file was nothing compared to the horrendous series of bugs that I found in the program when I disassembled it. These are far too numerous to mention specifically and it is sufficient to note that there is no trigger routine within the virus. If POWER.EXE is run on its own, it will execute any EXE program at random (because of a duplicated buffer area associated with its file search routine).

Conclusions

In this instance, as with the AIDS disk, there are doubts about whether the two programs (IQTEST.EXE and POWER.EXE) were written by the same man. Certainly the READ.ME file indicates that the original distribution was done deliberately with knowledge of the virus. However, IQTEST.EXE does not contain debug information and is in a completely different format to the POWER.EXE file.

This is not the first time that virus code has been distributed directly by a shareware vendor and it obviously calls into question the whole system of collecting free software from bulletin boards, copying it to disks and then offering it for sale. The original concept of shareware was admirable, but in the UK this was first subverted by shameful exploitation of the original well-meaning authors and then damaged further by being used as a channel for the introduction of virus and Trojan code.

The ethics of selling software which you don’t own, don’t understand and can’t support are at the least questionable and at the worst downright criminal, especially when most of the authors get no recompense for their efforts.

I have not been in touch with *Transend Services Ltd.*, but I understand that the company claims only a small number of these disks were distributed. I sincerely hope that this is true and I would advise it to institute better security procedures as soon as possible. Only they know where they got this program from (there is no author identification in any of the programs although the indications are that they were American) and they might consider letting the rest of us know who duped them so easily.

POWER PUMP

Aliases: None known.

Type: Companion - creates hidden COM files with same name as target EXE file. EXE file remains unaffected.

Infection: EXE files at random but with no accompanying COM file.

Recognition:

File POWER.EXE file - look for the words ‘Power Pump v1.2 Virus - Silent But Dead.’ at offset 12452 in the COM file.

System This virus does not become resident.

Hex Pattern (for companion COM file)

```
504F 5745 5200 0D20 2531 2025
3220 2533 2025 340D 008D B627
```

Intercepts: None. This virus is non-resident.

Trigger: None.

Removal: Locate and delete companion COM files. Also delete the POWER.EXE file.

VIRUS ANALYSIS 2

Commander Bomber

A favourite pastime of virus researchers has always been to try to second-guess the virus writers by predicting just what future developments might entail. Such discussions are seldom published because this might feed ideas to the virus writing community.

Many researchers have always believed that a successful virus would always contain some self-recognition capability on disk to prevent multiply infecting the same file. However, the latest virus to arrive on my desk for disassembly has destroyed part of this cherished assumption in a way that was speculated upon over two years ago, and does it with the most complex code yet seen in a virus.

There is no immediate cause for concern, since this virus has not been reported at large, does not use code encryption and is easily recognisable with even a simple file examination utility. However, the first sample brought to the West (apparently from Bulgaria) reportedly caused certain vendors to decide to suppress all information (even amongst known genuine researchers) 'because it was so dangerous'. It is debatable whether such a decision was genuinely in the public interest - the virus represents more of a threat to product manufacturers' methods than it does to users.

The virus has been called Commander Bomber, after an internal text message which reads 'COMMANDER BOMBER WAS HERE'. It is reported as the work of the so-called 'Dark Avenger', and although there are some similarities of style, my own feeling is that this code is beyond his limited capabilities.

Commander Bomber is a resident virus that infects memory image files between 5120 and 61183 bytes in length which are invoked by a LOAD-and-EXECUTE system call. Infected files grow by exactly 4096 bytes although the actual virus code is only 2496 bytes long. The reason for the discrepancy is that the virus needs space for data used in reconstitution and repair of host files.

Code-In-The-Hole

Where other parasitic viruses prepend, overwrite or append their code to the host file, this virus inserts its code at a randomly chosen position between 32 bytes from the beginning and 4064 bytes from the end of the host file. At this random position, 4096 bytes of host code are removed and added to the end of the file. The virus code, together with its attendant data areas, is then inserted into this 'hole'.

This on its own would create only minor problems if the virus code was then executed by a simple jump from the beginning of the file, but this is where the devious and malicious mind of the writer has expended most of its considerable effort.

The virus contains several routines which generate random code. This 'junk code' does nothing except bring processing (eventually) to the virus code proper. The effect is to produce a file which does not start with a jump or call, but has code which seems 'normal' inserted at random spots throughout the file. The integrity of the code is maintained and the range of op-codes generated is almost complete - even memory modifying instructions are included.

"the first sample...reportedly caused certain vendors to decide to suppress all information 'because it was so dangerous' "

To any cursory inspection, the 'junk code' appears quite normal and contains conditional and unconditional jumps and calls exactly like proper code. There is no attempt to maintain the value of any registers with the single exception that the condition of the stack is monitored. The random generation routines include occasional checks for processor type so that processor specific op-codes are not generated incorrectly. While the code is sophisticated, there are several bugs which may cause system malfunction when an infected file is executed.

Installation

When the virus code is executed it first checks the DOS version and exits directly to the host code if the version is earlier than 3.*xx*. The code then searches the environment to locate the path and name of the program being executed. Once this has been obtained, a familiar 'Are you there?' call routine is executed which calls INT 21h with a value of 424Fh in the AX register. If the virus is resident, the call returns a value of 4D42h in AX (these values represent the ASCII letters BO and MB), and processing is passed to the host code.

If the virus is not resident, a secondary rebuilding routine is processed which repairs various sections of the virus pre-processing code before re-writing the infected program back to the disk. Due regard is taken of the existing Time and Date stamp of the file, but any Read Only setting in the file attributes will be removed.

The virus code (2596 bytes) is then relocated to the normal COM offset in memory and the intercept routine is hooked into the system at INT 21h using the normal system services. The next phase of installation calls the infected file and executes it as a child process before finally exiting via the system TSR (Terminate and Stay Resident) function.

Commander Bomber uses a clever trick to avoid multiply infecting files. Whenever a file infected with Commander Bomber is run, the virus saves the memory image of the file to disk before disinfecting the file in memory and passing control to the host program. Therefore, if a file is doubly infected with the virus (as is briefly the case when running an infected file with the virus resident), the host is rebuilt as it was when it was first infected - i.e. a program which is infected *once*. When control is passed over to this host program, the second infection of Commander Bomber saves the memory image of the file (i.e. with only one infection) to disk. Therefore even though the virus cannot recognise itself in files it will not multiply infect them. [*Simple! Ed.*]

Operation

The complexity of this virus code precludes a detailed description of its operation but the general operation of the intercept code works as follows:

In order to load and execute a program, the name of the program must be passed to system function 4B00h of INT 21h. The virus intercepts this function call and stores it while locating and opening the target file. The first word of the file is examined to see whether it is either the ‘MZ’ or ‘ZM’ header which signifies a segmented EXE file. Files containing this header are *not* infected and execute normally. This process does not test the filename extension.

However, the virus does check to avoid infecting files with a name of COMMAND (with or without an extension). The length of the file is also checked, and only files between 5120 and 61183 bytes long are infected. The virus does not check that the target file is already infected: *all* files of an acceptable length are infected as a matter of course. Thus the virus does not need to recognise its own existence within an infected file (see above) and this particular Achilles heel has been protected.

Once the file has been infected and re-written to disk, the original function request is allowed to continue.

The Implications

Plain pattern recognition scanners which scan only the beginning and end of files (to increase scanning speed) will need to be modified to allow them to complete an exhaustive scan of files.

The other so-called ‘smart’ scanners which use processing flow analysis to locate the virus code may find that their analytical capabilities will need considerable enhancement to cope with the range of ‘junk code’ which this virus generates. The absence of a self-recognition signature within the code is particularly alarming. The hallowed precept that a virus must always recognise itself on file if it is to avoid multiply infecting its host has been shown to be false. This discovery, if further developed, will contribute significantly to the difficulties faced by scanner developers who have traditionally relied on this self-recognition file signature as a part of the detection process.

Fortunately, good generic integrity checking software will have no problem in noticing that a file has changed.

COMMANDER BOMBER

Aliases: Bomber.

Type: Parasitic, inserting virus.

Infection: COM and memory image files invoked by the LOAD and EXECUTE function, between 5120 and 61183 bytes in length. COMMAND.COM is exempted.

Recognition:

File The text message ‘COMMANDER BOMBER WAS HERE’ is plainly visible within an infected file and the file will be 4096 bytes longer than it should be.

System An ‘Are you there?’ call which returns 4D42h in AX if INT 21h is called with a value of 424Fh in AX.

Hex Pattern

```
E852 FFD1 E096 2EFF 9400 04EB
BE2E 0460 066F 0685 06A3 06E0
```

Intercepts: INT 21h for infection and detection of ‘Are you there?’ call.

Trigger: There is no trigger routine, but infected files may fail because of bugs in the code generation routines.

Removal: Specific and generic file disinfection is not possible. Under clean system conditions, identify and replace infected files.

VIRUS ANALYSIS 3

James Beckett

Uruguay 3 - A Slippery Eel

The Uruguay 3 virus is one of a series of viruses that seem to have been written in a progressive way, with new ideas being added in turn to a basic model. Within each of the viruses is a short piece of text:

```
'Uruguay-#3' Virus Programmed in Montevideo
(URUGUAY) by F3161. 06/92.
This is a research virus - DO NOT DISTRIBUTE
```

Correlating the text in the viruses, they appear to have been finished at a rate of about one a month.

The world of anti-virus research is constantly plagued by the question of whether viruses should be written to test ideas and theories, to pre-empt the use 'in anger' of the ideas by a less benign virus author. Most researchers seem to be against the idea, with a few proponents staunchly touting the banner of Freedom of Speech or claiming that such research can be kept under sufficiently tight control.

Unfortunately, this stand provides the virus authors with a convenient cover for their activities and a way of disclaiming responsibility for their creations. The message in this virus claims it is a research virus, but why should any bona fide researcher be shy of admitting his real identity?

Analysis

As its name suggests, there are two viruses before this one in the Uruguay series. Neither of these seems to replicate under test conditions.

The message above is stored encrypted within the virus, and is only displayed once in 100 times, with random musical accompaniment. Again, one wonders whether this seems the likely behaviour of a true researcher.

In this virus, several techniques are employed which put it in the class of the more advanced viruses seen to date. That is relative, of course: most viruses are very simple and very stupid; here, we have a COM/EXE infecting, resident virus employing code encryption with random key decryption, code obfuscation and interrupt tunnelling.

The code obfuscation is intended to confound recognition by simple hex-pattern based scanners. The decryption routine consists of only five or six instructions, but they can each take several forms and are interspersed with runs of other random instructions.

The Decryption Routine

The actual en/decryption routines used in most viruses are extremely simple, involving no more than ADDs, SUBtracts, or most commonly XORs; there is no cryptographic problem to solve in dissecting them. The reason that the encryption has such a nuisance value is because the pattern-searching of a virus checker must be confined to the decryption routine, and with only a few bytes of code to implement, the decryption routines can take many different forms without an excessive expenditure of programming. Additionally, irrelevant instructions can be interspersed between them without unduly increasing the overall size of the virus.

The obvious and usual way to load a machine register with a value is a MOV instruction - by definition loading the register with the number given. By assuming that the registers AX, BX and CX are zero on entry to the virus code, the author has utilised, at random, an ADD, OR, and XOR instruction to confuse the code.

The five or six instructions that comprise the decryption routine can each appear in several forms, giving approximately ten thousand permutations using different sets of registers. This is augmented by adding a series of up to 33 bytes of random instructions between each useful one, such that the operation of the routine is not affected. Each run is composed from a list of 73 filler instructions, giving some $3.1x10^{20}$ possible permutations. Multiplying the lot together gives a grand total of $1.97x10^{144}$ different combinations (for the non-mathematicians, that is a number composed of 2 followed by 144 zeroes.).

Running a program within DEBUG can present a slightly different environment. Most programs make no assumptions about the values of any registers, so their contents make no difference. This virus rather lazily depends somewhat on the initial state of the registers, though in this case they should make no difference to the execution of the code or its analysis; is this an oversight or an attempt at confusion? Any debugger which does not adequately simulate a genuine environment runs a risk of causing the code to run incorrectly, and it would even be possible for a virus to take advantage of this to trip up anyone using such tools and deliberately hinder disassembly.

Interrupt Tunnelling

A resident program looking for 'virus-like' activity traps the software interrupt vectors to spot the calling of functions which the designers consider indicative of a virus' operation. A tunnelling virus subverts such TSRs by locating the original vector and making its system calls directly to the system, bypassing the monitor.

There are several ways of tunnelling; for some vectors there is an undocumented DOS or BIOS call to find the original vector directly, and some viruses use this method. A knowledgeable anti-virus program author could of course trap this vector and return its own vector address to the virus, although this may disrupt other programs which have a legitimate need for the information.

Uruguay 3, like a few other viruses, has adopted another method, ironically enough using the 80x86 processor's internal debugging system.

When confronted with a program which is behaving in an unpredictable or inexplicable way it is often useful to be able to trace through assembly code instruction by instruction. The processor facilitates this by checking a special flag at the end of executing each instruction, and if the flag is set, it automatically makes a call through a software interrupt. This is intended to be trapped by the debugger which halts program execution to display information about the state of registers and flags within the machine.

A method of tunnelling called 'interrupt stripping' uses trace traps to follow the chain of interrupt handlers until it finds the original DOS handler in segment 0070h. Calls made direct to this location will bypass all the other resident programs in the system, and allow the virus unimpeded access to the basic facilities of the computer.

After going resident, Uruguay 3 tunnels both INT 21h (the DOS function despatcher) and INT 13h (the BIOS disk access interrupt), and when activated by a program execution request, temporarily interposes its own handlers or bypasses the chain of installed handlers. It even writes in a JMP FAR to itself into the start of the INT 21h routine itself, so that **nothing** can call the DOS INT21h services directly. Any TSRs which are hooked to these interrupts will remain blissfully unaware of these foul deeds.

Infection Method

Uruguay 3 infects on execution of a DOS file (COM and EXE format). It also intercepts file open (read-only) and returns an error if any attempt is made to open a COM or EXE file, neatly side-stepping the question of stealth. Any virus scanner run while Uruguay 3 is resident will produce countless errors when trying to access files, or abort on the first file it attempts to access.

This interception occurs at the level of the stripped interrupt, so no program will prevent it if the virus has been allowed to install itself.

Files are marked to prevent re-infection by rounding the size to the nearest twenty-three bytes. The virus will therefore not infect about 4% of all suitable files.

Further Versions

Uruguay 5 (dated 07/92) concentrates on the decryption code with minor modifications to overall operation. Multiple decryption instructions have been added, so that instead of, say, a single XOR in the loop, one can find up to 5 instructions, XORs, ADDs and SUBtracts in any order. This increases the number of permutations to something over $5.64x10^{246}$.

Uruguay 6 (dated 08/92) uses dynamic in-memory decryption and re-encryption of program code. At the start of many of these subroutines a procedure is called which decrypts the code following, using a one-byte key and two-byte end address stored immediately after the CALL. This procedure increments the program counter on the stack to return to the address after these three bytes, and sets up the stack to call the re-encryption routine automatically when the decrypted code finishes. This can cause some problems in analysis, but there is nothing that cannot be solved by iterated disassembly or the use of a good debugger.

Presumably this edifying research is continuing - what will Uruguay 7 entail?

URUGUAY 3

Aliases:	None known.
Type:	Resident Parasitic.
Infection:	COM files which have a length not divisible by 23.
Recognition:	
Files	File size is a multiple of 23.
System	Calls INT21h with AX=3032h, DX=1234h. Returns 5678h in AX if resident.
Hex Pattern	No simple search pattern is possible.
Intercepts:	INT21h for infection, blocking of scanners, and 'Are you there?' call.
Trigger:	Displays warning message and plays random tones on the internal speaker.
Removal:	Under clean system conditions, identify and replace infected files.

NETWORKING

Igor Grebert
McAfee Associates, USA

Anti-Virus NLMs

In recent years, as the technology to connect individual PCs has become faster and more reliable, large numbers of networking products have become available. Among those, *Novell* and its server based networking platform is undoubtedly the most commonly used.

Within the past few months at least five companies producing anti-virus software have announced or released *Novell* specific products, most of which are NLMs. NLM stands for *NetWare* Loadable Module. NLMs are extensions of the *Novell NetWare 3.xx* operating system, and in a sense, they are to *NetWare* what TSRs are to DOS.

To understand better what to expect from such products, let me give a brief description of the networking environment in which they operate.

NetWare

For performance reasons, a *Novell* network requires a PC to be dedicated as a file server. It is the platform on which the *NetWare* operating system is loaded, and from which all aspects of networking are controlled. *NetWare* cooperates with software executed on the workstations to provide shared resources across the network.

NetWare provides several layers of security. Access to the system is controlled by passwords and account restrictions, while access to files and directories is controlled by privileges and attributes. When correctly set, accounts and privileges provide a much better safeguard against computer viruses than attributes in the DOS file system. The Read-only or System attributes in DOS are merely warning flags and are very easily bypassed. However, File Rights under *NetWare* cannot be bypassed with such ease.

NetWare can execute more than one program simultaneously. While one process is sending a file to a workstation, another one can check the validity of a password, while another performs a backup. It is the multi-tasking aspect of *NetWare* which allows the file server to provide multi-user access to all resources. It is a non-preemptive multi-tasking operating system, which means that each process is given full control when first run. The program is expected to return control regularly to the operating system, so that other programs may perform their tasks.

When used correctly, the security provided with *NetWare* offers a good way to limit the potential for viral infections. Alone, however, it is not enough to secure a network completely. Supervisor privileges have to be used from time to time, and some DOS applications do not allow the file attributes to be set in the most secure way, leaving loopholes in the file protection.

Types of Scanning

There are two main types of server-based scanning offered by anti-virus NLMs for *NetWare*.

The first one, called ‘direct’ or ‘on-demand’ scanning, allows scanning of all volumes directly available from the file server. The virus scan must be performed on a regular basis, for example once a day, or before a backup. Very similar to running a daily scan of a drive on a workstation, this method allows detection of viral infections only *after* they occur. Its major disadvantage is that it always leaves a period of time between two scans during which the virus may spread between workstations. Also, this method does not allow the system manager to determine how the virus was introduced onto the network. Therefore, when an infection is detected, all workstations must be scanned, and the network must typically be brought down to allow effective removal of the virus.

> *“DOS viruses cannot fool the NLM as they could DOS virus scanners”*

The second method, called ‘real-time’ or ‘on-access’ scanning, is the equivalent of memory resident scanning on a workstation. Much more difficult to achieve technically, this type of scanning offers real time virus detection.

Although *Novell* made it easy to call most *NetWare* internal functions from an NLM, there is no provision for simple internal hooks as in DOS. The DOS interrupt structure allows developers to easily get control of any part of operating system. This ease of access to sensitive areas within DOS has contributed to the number of viruses written for DOS. However, innovative programming techniques can make such hooks viable under *NetWare*, and NLMs using them are able to monitor any file access. The best type of protection controls all incoming files. If a virus is copied on a network drive, or if a workstation is infected, and the virus tries to infect files on the network drives, the NLM will detect it, and will usually prevent the infection operation from being completed.

Advantages of NLMs

A distinct advantage of using server-based scanners is that they overcome some of the existing weaknesses in workstation-based products. Not all network activities are monitored by TSRs, and in some cases available memory is too scarce to run monitoring programs on the workstations.

If one station on the network is not running a monitoring program, it represents a potential risk to the network drives. NLMs that monitor all file activities therefore provide proactive network security.

When a virus attempts to infect a file on the network drive, the NLM can stop the operation before changes are written to disk, and send messages to key users. This provides the system administrator with an early warning system to track infected workstations which may not have a monitoring program running. Clearly this extra information is of great help when dealing with a virus outbreak.

Being executed on the file server itself, NLMs have more access and control than a workstation application would have. It is important to examine what action the NLM can take after a virus is detected. Typically, the NLM should offer the ability to delete an infected file, to move it to a special directory (for further analysis), to log which user or workstation was the source of the infection, and to inform a list of users.

Also, because the machine on which the NLMs are executed is controlled by the *Novell* proprietary operating system, DOS viruses cannot subvert the NLM as they could DOS virus scanners. Stealth viruses are thereby stripped of their camouflage, leaving the way clear for scanners to detect them.

The centralized operation is very convenient for the system administrator. Only one program needs to be updated with new virus patterns. If real-time scanning is not available in all NLMs, most of them allow scheduled scans. This provides the Supervisor with a way to monitor virus infection on the system.

Conclusion

When well implemented, NLM virus scanning provides the network administrator with a simple, convenient, and centralized way of protecting network drives from infection.

Early detection is the key to minimising the cost of a virus incident. On-access scanning is therefore an important feature to look for in anti-virus NLMs. Ultimately, however, whether you run an anti-virus package from a workstation or from a file server, remember that the most critical issue is the ability of the product to detect viruses.

NLM SURVEY

Richard Ford
Virus Bulletin

Barrie Layfield
Information System Networks plc

Virus Protection Under *NetWare*

This survey examines those scanners designed for use on *Novell* networks. All these products are NLMs, and offer centralised virus detection and reporting for systems running *Novell NetWare 3.1x*. NLM virus protection has many advantages over DOS-based scanning, and unless otherwise stated, all packages provide scheduled background scanning and real-time file scanning. For details of the hardware configuration used and the virus test set please see the *Technical Details* section at the end of the article.

Intel's LANProtect

LANProtect arrived on both 3.5-inch and 5.25-inch disks, of which only the 5.25-inch disk was permanently write-protected. The documentation supplied is a slim booklet, which gives an overview of the different programs which make up *LANProtect*.

The Supervisor is required to login from a workstation and execute the installation program from one of the disks supplied. The only criticism of this procedure is that at no time is the Supervisor warned to reboot the workstation from a clean system disk. While any network manager worth his salt is only too aware of the need to do this, a warning should be present in the manual.

Once installed, *LANProtect* is controlled from the system console. The screen offers a simple menu-driven interface which allows the software to be configured. The product is easy to use and the function of each menu is clear.

When a manual scan is in progress the server displays the name of the current file being scanned, and any viruses found within it. When running a prescheduled scan this information is not displayed - instead, a report is generated. This report can be either viewed from within *LANProtect*, sent to a file, or sent to the printer queue.

LANProtect missed six of the viruses in the test set. This is a poor result which needs to be improved.

Workstations can also be scanned using *LANProtect*. This is done by executing the DOS program LPSCAN, which uses the same virus signatures as the NLM scanner. In addition to scanning for viruses, LPSCAN also has some disinfection capabilities. In tests, it successfully disinfected a number of programs. LPSCAN can be password protected so that it is only run by those authorised to use it. It should be noted, however, that because this program is a DOS executable, it is susceptible to stealth viruses which are already resident on workstations.

The last package in the suite is PCSCAN, a memory resident program which checks executed files for viral infection. It identified all but six of the viruses in the test set before execution and successfully prevented these infected programs from being run.

LANProtect is a carefully thought-out product, and is well written. The disk scanning part of the package is fast, and the pre-scheduled disk scan is sufficiently flexible to suit most users. In addition to this, the incoming and outgoing file scanning is a useful line of defence.

The largest drawback with *LANProtect* is its relatively poor detection rate. However, *Novell* has demonstrated great faith in the product by placing a worldwide internal licence for it - a recommendation in itself.

McAfee's NetShield

McAfee's NetShield is a shareware product. The review copy arrived on a 5.25-inch disk directly from *McAfee Associates*, but the software is also available via Bulletin Board Systems.

There were no installation instructions on the disk, so there was no choice but to leap in at the deep end and copy the NLM onto the file server. However, *NetShield* would not run until another file containing the virus signatures had been copied across.

The lack of documentation *could* be a major drawback with *NetShield*, especially during the installation procedure. However, once installed, the user interface is sufficiently good that this is not the case - running the software is easy due to the simplicity of the control system.

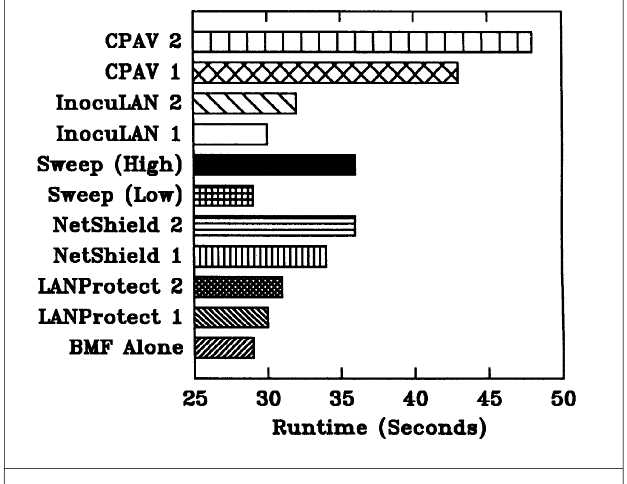
The configuration options provided are reasonably flexible, and are very similar to those offered by the other products. Periodic scanning is allowed, and this can be done either once a day, once a week or once a month. All these functions are controlled from the system console.

Scanning can be undertaken 'on the fly' on incoming and outgoing files. When a virus is detected, a message can be displayed on the workstation of the offending user, in addition to informing designated users. The infected file is then overwritten, deleted, moved to a specified directory or ignored, depending on the software configuration.

McAfee Associates' other anti-virus products have a good reputation for detecting viruses, and it is surprising that *NetShield* did not identify all the viruses in this test-set - it missed six, including a copy of the Syslock virus. This is a puzzling result, as all of these viruses were detected by the current version of *McAfee's Scan* product. *McAfee Associates* claims that it is aware of this problem and that it will be corrected in later releases.

Reports can be generated by *NetShield*, and are written to a log file. If the log file is viewed from within *NetShield* the contents of the file appear in a box which fills the lower portion of the server's screen. The log file contains all the relevant information which a network manager needs to know, giving details of which virus was found, when, and what action was taken. If the infected file was found when being accessed by a particular user, the user's name is also recorded. The options when viewing the log file are somewhat limited; a find function would have been useful, as would the ability to view the data from a specific date.

It is difficult to predict how network managers will receive a *shareware* product designed to protect the integrity of their LAN. Money is not such an overriding concern when protecting a network - the entire cost of the scanner will frequently be less than a day's downtime, and buyers may



Network Load. The time for a standard test program to run is measured during scanning of files. Where appropriate, the labels 1 and 2 refer to scanning with and without scanning of incoming files. For individual product details see text.

(incorrectly) assume that the more you pay, the better a package you get. However, the *McAfee* approach to network protection is direct, and has a clear and easy-to-use front end.

All in all, therefore, *NetShield* provides a no-nonsense approach to the job of virus detection under *NetWare*.

Sophos' Sweep

Of all the NLMs on offer at the moment *Sweep* from *Sophos* is the simplest. The installation procedure consists of copying one file from the installation disk into the system directory of the file server. *Sweep* is then executed by a simple LOAD command.

Sweep has been designed for operation from the system console, and is controlled by a series of easy-to-use menus which allow the various options to be set up.

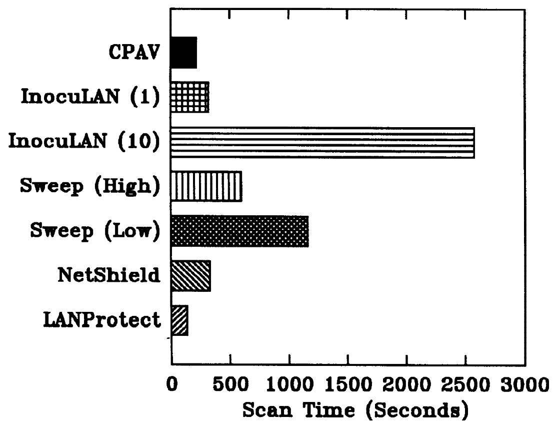
All the other NLMs can optionally check all incoming and outgoing files for viruses. *Sweep* is not capable of doing this, but instead offers continuous background scanning of files in the server. This seems rather like closing the stable door after the horse has bolted, as it will only warn of virus infection *after* that infection has taken place. However, background scanning should detect virus infection quite quickly, so some users may feel that this is sufficient.

Sweep does have one advantage in that it is extremely easy to set up and use - while *Sophos* may not repeat *Amstrad's* offer of 'learn to use it in five minutes or your money back' it really was simple to operate. Scans can be optionally carried out in the background at scheduled times, or run immediately. In detection tests, *Sweep* scored very well, detecting all of the viruses in the test-set.

The *Sweep for NetWare* package comes complete with a site licence for the DOS version of *Sweep*, thus allowing all workstations to be swept. While the DOS version of *Sweep* can be run from the file server, *Sophos* still sensibly recommends a clean boot for complete protection of workstations.

Sophos says that it has omitted on-line file scanning from the current NLM because it can only be done by 'hacking' *NetWare*, and that this feature will be available for *NetWare* v4, which supports the necessary functions properly.

Sweep is entirely capable of protecting a network, and is easy to operate. The principal drawback ('hack' notwithstanding) is its lack of a 'real-time' file scanner.



Comparison of the scan times. *LANProtect* was extremely fast, leaving all the other products behind. The scan times recorded are the time taken to scan all executables on the file server. For individual product details see text.

Cheyenne's InocuLAN

InocuLAN by *Cheyenne Software* contained not only a manual for the Supervisor, but no fewer than *five* manuals explaining how the features offered to other network users function - an excellent idea. The Supervisor's manual is factual, though not very readable. However, it compensates for this with an easy-to-follow checklist for the installation process - another good idea.

Installation is easy: the user simply has to sit back and wait while the appropriate files are copied onto the workstation and the file server. *InocuLAN* is not fully configurable from the system console and so has a DOS-based control program which is executed from a workstation. This is somewhat unnecessary - all the options could just as easily be set from the console.

In addition to this, it detracts from the security of the system, as it requires the Supervisor to log on to control the software. DOS is inherently less secure than *NetWare*, and these NLMs should limit their reliance on DOS to a minimum. If a program *needs* to be reconfigured from a workstation this can be done using the RCONSOLE utility.

The usual option of interception of incoming and outgoing files is offered, along with a scheduled file scan option. The scanner itself, however, did let the product down by missing six viruses from the test-set.

Who would be a product reviewer? Reading software manuals is never fun at the best of times, but reading five of them...

InocuLAN also comes with a number of TSR programs which are designed to help stop viruses from spreading on the workstations, as well as a generic detector. The generic detector, PREVENT, claims to detect virus-like behaviour 'such as unauthorised formatting of the hard disk'. Such claims are often treated with some scepticism, but PREVENT does appear to work: when run against some of the new viruses reported in last month's *Virus Bulletin* it successfully detected all of the viruses when they were executed - an impressive result. It is not clear, however, how PREVENT works as in tests certain straightforward modifications to files went undetected - a puzzling result.

The remaining two TSRs check files for viruses at load time, and the user has the choice of either checking for those viruses deemed to be 'in the wild' or for all the viruses *InocuLAN* is capable of detecting. These programs seem to work as described in the manual, checking all files which are deemed to be executable when they are opened. Unfortunately, once loaded it was not clear how to unload these TSRs without rebooting the computer.

InocuLAN offers good protection for a *Novell* Network, and has a couple of nice touches to boot (such as five copies of the manual for the workstation software). The virus-specific detection ability of the software does need to be improved, but the generic detection offered appears to work well.

Central Point Anti-Virus for NetWare

Central Point's product arrived in two boxes, bearing the standard *Novell* badge with the words '*Novell* Network Aware' proudly printed in white on red. The package comes in two parts, a DOS part for the workstations and an NLM part (which requires *NetWare* v3.11) which provides the background scanning and file interception features which the products offer. *Central Point's* product is capable of detecting both MS-DOS viruses and Macintosh viruses.

All of the new *Central Point Anti-Virus* products have the ability to communicate with each other. By using this facility, the company offers a completely centralised anti-virus strategy which will automatically report all virus infections to one central point - no pun intended!

The *Central Point* product is designed to be controlled from a workstation. The configuration and control program which comes with the software is undeniably glossy and is easily used with or without a mouse. However, by making the software fully usable only from a workstation, one of the most important advantages of anti-virus NLMs is removed, since the Supervisor is required to log on to the system. If *Central Point* wants to offer system managers the option of a glossy interface it should be in addition to, and not instead of, a simple display on the system console.

When a virus is detected, the software updates the log file, and can inform designated users of the discovery. This can be done either by a Network Broadcast message, an MHS mail message or by paging the system manager using a modem connected to the server.

The virus detection ability of the software was good, missing only the Father virus from this test-set.

The package also offers generic virus detection, by monitoring the way programs access executable files. As this method is highly prone to false positives the user is not informed of suspected virus behaviour. However, every time a possible virus is found, a note is made in the log file. The generic checker successfully identified the actions of several viruses, but also identified DEBUG as showing virus-like behaviour.

There were a couple of irritating bugs in the product. When operating in a DOS shell under *Windows 3.1,* every time a virus is discovered, the machine returns to the *Central Point* control program. This repeated return to *Windows* eventually managed to confuse the application running in the DOS shell to the extent that it crashed. Also, Network Broadcast Messages occur randomly - when a large number of viruses are discovered the user is not informed of all infections found. These bugs aside, the *Central Point* product does provide a good level of cover for the file server, and offers a complete package for virus protection.

Network Load

Possibly the most important aspect of the software is the question of network load. Virus protection is clearly unworkable if it loads the system so much that it is unusable. In order to determine some measure of the network overhead that the packages impose, a test program was run while scanning was in progress. The test program imposes overhead on the server by requesting a combination of random and sequential reads and writes. Clearly, the greater the overhead imposed by the anti-virus software itself, the longer this test program will take to execute.

A graph of the network overhead is shown on page 16. The network overheads are (products running in fastest modes without in/outgoing file scanning): *LANProtect* 3.4%, *InocuLAN* 3.4%, *NetShield* 17%, *Sweep* 24%, *CPAV* 48%.

Speed

Overall scanning speed is arguably not as important for an NLM as for a DOS based scanner, as the user is unlikely to sit waiting for the NLM scan to finish - one of the principal advantages of server-based protection is the automation of this onerous task. All products were tested over the same set of files, and asked to scan executable files only (86.2 Mbytes). The scan times are shown in Graph 2 (page 17). Products which appear twice in the graph have been run using different internal speed settings.

When comparing like with like (scanners running in ‘Advanced’ modes at full speed) the scan times in minutes and seconds were as follows: *LANProtect* 2:13, *CPAV* 3:38, *InocuLAN* 5:18, *NetShield* 5:29, and *Sweep* 9:58.

One point to note is that a scanner can be made faster by increasing its overheads on the file server. *NetWare* gives full control to any NLM running on the server, and the NLM is, in turn, expected to return control to the operating system to allow other products to run. Therefore, there is a trade-off between network overhead and scanning speed.

Conclusions

Choosing anti-virus software has never been easy, and with NLMs many different factors must be taken into account. The stability of the file server is of utmost importance and because NLMs operate at a low level within the system it is vital that they are well written. From this point of view *Intel’s LANProtect* package seems to offer the best features as it is the only product tested which has been through the *Novell Laboratory’s* testing procedure. It should be stressed, however, that this testing only ensures that the software is ‘well behaved’ and does not cause the system to become unstable.

The virus detection ability of the software is equally important. *Sweep* detected all the viruses in the test-set, with *Central Point* coming a close second, missing only one infection. The biggest surprise of the review was the relatively poor detection performance of *NetShield*. *McAfee Associates Scan* is capable of detecting all the viruses in the test-set, which implies that these false negatives are almost certainly due to teething problems with the new product.

Because of the large number of factors to be taken into account, it is difficult to choose a clear leader out of this range of products - the reader is left to decide for himself which product best suits his needs.

Supplier Details

Product: *LANProtect*
Version Evaluated: 1.5
Cost: Annual price per server £599
Manufacturer: *Intel Corporation (UK),* Pipers Way, Swindon, Wilts. SN3 1RJ. Tel 0793 696000. Fax 0793 430763.

Product: *NetShield*
Version Evaluated: 1.02
Cost: Annual price per server £424
Manufacturer: *McAfee Associates Inc*, 3350 Scott Boulevard, Building 14, Santa Clara, CA 95054-3107, USA.
Tel (1) 408 988 3832. Fax (1) 408 970 9727.

Product: *Sweep*
Version Evaluated: 2.43
Cost: Annual price per server. 1-25 Users £495. 25+ Users £895.
Manufacturer: *Sophos Ltd*, 21 The Quadrant, Abingdon Science Park, Abingdon, OX14 3YS. Tel 0235 559933.
Fax 0235 559935.

Product: *InocuLAN*
Version Evaluated: 1.1
Cost: $495 for up to 20 users. $995 for up to 250 users.
Manufacturer: *Cheyenne Software Inc*, 55 Bryant Avenue, Roslyn, NY 11576. USA. Tel (1) 516 484 5110.
Fax (1) 516 484 5220.

Product: *Central Point Anti-Virus for NetWare*
Version Evaluated: 1.0
Cost: £699 per server, unlimited users.
Manufacturer: *Central Point Software International Ltd*,
3 Furzeground Way, Stockley Park, Uxbridge, Middlesex.
UB11 1DA. Tel 081 848 1414. Fax 081 569 1017.

Technical Details

Hardware Used: 33 Mhz ’386 PC with 300 Mbyte SCSI disk and *Adaptec* SCSI controller, and 8 Mbytes RAM, running *Novell NetWare* v3.11.

Viruses used for testing purposes:

777, 1575, 2100 (2), 4K (2), Anti-Cad (2), Cascade (2), Captain Trips (2), Tequila, Eddie, Eddie 2 (2), Dark Avenger (2), Darth Vader (3), Dir II, Father (2), Flip (2), Hallochen, Invader (2), Jerusalem (2), Keypress (2), Liberty (2), Macho (2), Maltese Amoeba, Mystic, Nomenklatura (2), Nothing, PcVrsDs (2), Penza, Slow (2), SBC (2), Spanish Telecom (2), Spanz, Syslock, V2P6, Vacsina, Vienna (4), Virdem, Warrier, Warrior, Whale, Yankee (2).

PRODUCT REVIEW

Dr Keith Jackson

Iris' Anti-Virus Plus from *Menorah*

Anti-Virus Plus consists of four distinct modules: CURE which scans for virus infections, IMMUNE which acts as a memory resident monitor and checks executable files for virus signatures, EXAMINE which checks that no unwanted changes have been made to the important areas of a hard disk, and PREVENT which looks for 'viral activity'. The *Anti-Virus Plus* package comprises an A4 bound manual, and software stored on a single 3.5 inch floppy disk (which did not arrive write-protected).

Copy-protection

The first paragraph of the manual discusses what to do if the copy of *Anti-Virus Plus* is copy-protected; stripped of verbiage it explains that you have to use the original disks - a backup copy will not work correctly. The manual explains that the copy-protected version of *Anti-Virus Plus* will not work from a floppy drive, and must be installed on a hard disk. This prevents the user ever working from a write protected disk. I am at a loss to understand the circumstances in which the copy-protection version is sold, as the manual simply says 'Copy-protected versions of *Anti-Virus Plus* are sold for use on a specific number of computers'.

The version of *Anti-Virus Plus* supplied for review was not copy-protected, which is not surprising, as *VB* has a policy of not reviewing copy-protected software. The reasons for this decision have been discussed at length in previous issues [*VB, July '90, p.18. Ed.*], and will not be repeated here. However, the manual makes it very clear that *Anti-Virus Plus* is sometimes sold in copy-protected form.

As a simple test, use the MS-DOS command DISKCOPY to make a copy of *Anti-Virus Plus* floppy disk and try to install from this copy. If this will not work then the version you have been sold is copy-protected and should be avoided. Ask for a refund - life is too short to put up with the inanities that can be introduced by such schemes. [*Menorah has informed VB that it will never sell the software in its copy-protected form. Ed.*]

Installation

The installation process recommends running the scanning program (CURE) from floppy disk before installation is performed. This is admirable, but why doesn't it just perform the scan for itself during the installation process?

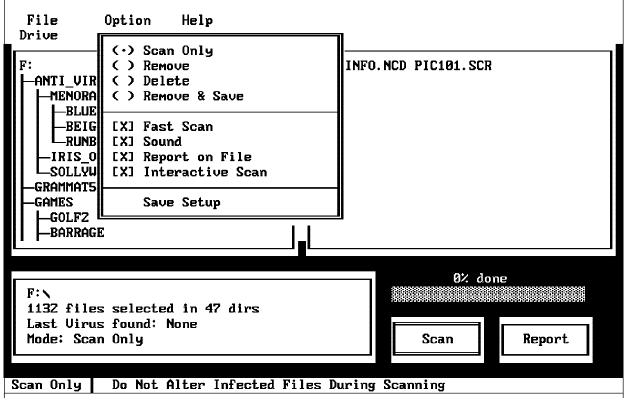
Installation to hard disk is very simple. It is merely a matter of choosing 'Install *Anti-Virus Plus*' from a menu, and the installation program then does the rest. A blank, formatted floppy disk is required during installation, which is used to store signatures of the 'system files and other sensitive information'. If this floppy disk is write-protected then the installation program issues an error and terminates. Surely a user prompt and a retry is in order?

During installation *Anti-Virus Plus* only allows the drive to be specified, not the directory where the files will be stored, and it insists on installing itself into a directory off the root directory of the chosen drive. The name of this directory cannot be changed; it must be called ANTVIRUS. This is poor: good software packages permit installation in any nominated directory. The manual states that *Anti-Virus Plus* will work on a network. I have no means of testing this, but it should be noted that users have no choice where to install the software.

I attempted to install *Anti-Virus Plus* on an 8088 PC with three floppy disk drives (A:, B: and C:), and a hard disk as drive D:. Even though I instructed the installation program to use drive D:, it tried to write its signature files to drive C:, did not succeed, and exited back to DOS without reporting an error. CURE was quite happy to scan drive D:, so the fault lies entirely with the installation program.

De-installation

Anti-Virus Plus uses the same program to de-install itself as was previously described for installation. However all did not go well when I tested de-installation; having been shown which files had been removed, the error message 'Warning \ANTVIRUS cannot be removed' was displayed. This was because it was trying to remove the *Anti-Virus Plus* directory when five files still remained in it.



```
(•) Scan Only
( ) Remove
( ) Delete
( ) Remove & Save

[X] Fast Scan
[X] Sound
[X] Report on File
[X] Interactive Scan

    Save Setup
```

```
F:\
1132 files selected in 47 dirs
Last Virus found: None
Mode: Scan Only
```

```
Scan Only | Do Not Alter Infected Files During Scanning
```

The windowing system within the package provides an easy-to-use interface. The user can store setup information to disk for use next time the software is run.

Licensing Anti-Virus Software

When *Anti-Virus Plus* is used after being installed, a banner makes it clear that it is a product licensed from *IRIS* computers in Israel, whose anti-virus products were some of the first ever available. This appears to be quite common practice, as some components of the anti-virus product *VirusCure Plus* which I reviewed only 2 months ago were also licensed from *IRIS* computers (see the October 1992 issue for details). *VB* first reviewed *IRIS* anti-virus software in the January 1990 edition, so I dug out that review: although the program did not perform too badly, I had complained that it had very skimpy documentation; this has not changed in the better part of three years.

There is still no technical explanation of what the memory resident components actually do, and no explanation of any error messages that may be encountered, indeed nothing much beyond descriptions of how to run the software. In my original review I summarised the attitude of the documentation as being ‘figure it out yourself’. This is still true.

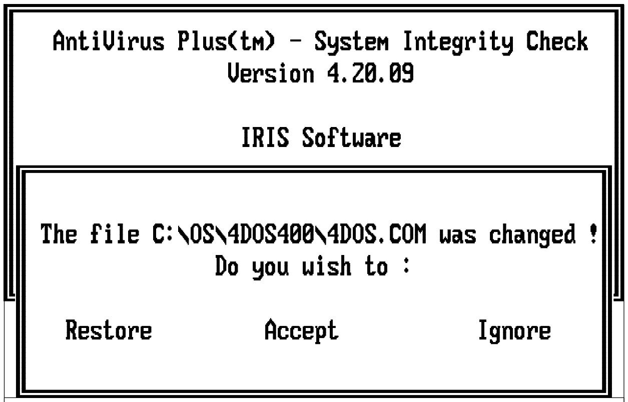
Scanning

The *Anti-Virus Plus* program called CURE invokes the virus scanner and other associated facilities. This is controlled by a windowing system with drop-down menus. It is unconventional when contrasted with most *Windows* programs, but is easy enough to use with either a keyboard or a mouse. The Escape key is required to back out of some menus so it cannot lay claim to being completely ‘mouse-driven’, but in these days of software conformity I found it a refreshing change.

By default, CURE will attempt to disinfect any file that it believes is infected. This can cause problems when a false positive is detected. Although the facility to remove a virus infection is a good idea, it should never be the default.

Anti-Virus Plus scanned my hard disk (680 files in total [20.6 Mbytes], of which 214 were checked), in 22.2 seconds when executed under DOS. The same scan time was recorded when CURE was executed under *Windows 3.1* - a creditable achievement. When all files on the disk were examined, the scan time increased to 50.3 seconds, and when *all* parts of each file was scanned the time increased again to 1 minute 42 seconds. When executing under DOS, *SWEEP* from *Sophos* (v2.43 running in quick mode) scanned the same disk in 16.2 seconds, and *Dr Solomon’s Anti-Virus Toolkit* (v6.01 running in turbo mode) took 18.3 seconds. *Anti-Virus Plus* is not the fastest scanner around, but the times reported above are quite acceptable.

The virus test set described in the *Technical Details* section was used to measure *Anti-Virus Plus’s* accuracy of detection. Regular readers of *VB* should note that this test-set has recently been extended somewhat by including more viruses that are known to be ‘in the wild’. *Anti-Virus Plus* correctly detected all of the boot sector viruses, and only missed 2 of the 209 parasitic virus samples. This is an excellent score. For the record, the two viruses not detected by CURE were one of the three Darth Vader samples, and the Spanz virus. I also tested CURE against 1048 samples of viruses generated by the Mutation Engine. It found 989 of the test samples to be infected, a success rate of 94%. Although this is quite good, there are several products around which can achieve 100% on this test-set.



```
AntiVirus Plus(tm) - System Integrity Check
Version 4.20.09

IRIS Software

The file C:\OS\4DOS400\4DOS.COM was changed !
Do you wish to :
```

One irritating bug within the software was that it refused to recognise the substitute command interpreter 4DOS. When this was removed, *Anti-Virus Plus* functioned as described.

Dynamic Detection

IMMUNE occupies 15K of memory while memory-resident, and detects virus signatures when files are copied or executed. It only examines files which have an executable extension - a point which the manual fails to discuss. IMMUNE is not quite as good as the scanning program (CURE) at detecting viruses, as it failed to detect 8 samples of the virus test set described in the *Technical Details* section (2 copies of Voronezh, 12 Tricks, 2 copies of Darth Vader, Maltese Amoeba, Tequila and V2P6).

Rather curiously, the two copies of the Darth Vader virus that were *not* detected were the ones that CURE *did* detect. However, IMMUNE *did* detect the one copy that CURE failed to detect! The manual states that IMMUNE will detect virus-infected files when they are executed, copied or renamed. This is not true, as IMMUNE detected nothing while renaming viruses stored on a floppy disk.

It is worth noting that IMMUNE did not access the hard disk during these tests, therefore it had to be holding its virus signatures in memory. These signatures appear to be a different set to those used by CURE. This explains how (but not why!) they can give differing results.

Signature Checking

EXAMINE inspects what it considers to be important areas of the hard disk (the command processor, system files, partition sector and boot sector) and checks that signature files created during installation have not been altered. The documentation gives no information on what algorithm EXAMINE uses. If any alteration is found, EXAMINE can rebuild the disk from the signature files.

When I first installed *Anti-Virus Plus*, it consistently reported that the file ‘4DOS.COM’ had been altered. This was not true, and I eventually traced the error to EXAMINE becoming confused by 4DOS being used as a replacement command interpreter. When I re-installed *Anti-Virus Plus* with the normal MS-DOS command interpreter present (COMMAND.COM), then all was well, and EXAMINE detected every bit-change that I introduced during testing.

Other Anti-Virus Checks

PREVENT is a memory-resident program (occupying 17K of memory) which looks for ‘signs of viral activity’, and ‘utilizes artificial intelligence to trap new strains of computer viruses’. Many questions come to mind, but to put it bluntly the explanations quoted above do not actually mean anything unless they are explained in more detail. PREVENT may be many things, but it is almost certainly not ‘Artificial Intelligence’ within any normally accepted definition of this phrase. It may work extremely well - without more details I just can’t tell one way or the other.

Conclusions

When I first reviewed anti-virus software developed by *IRIS* I concluded that the documentation was ‘appalling’. Three years on, it is not much better. If there was any explanation of the file WAV.DLL which was included on the disk, I would have tested it out. It is obviously a *Windows 3* Dynamic Link Library (DLL) file, but without knowing what it is supposed to do, it is a bit hard to test.

The scanner part of *Anti-Virus Plus* is not the fastest program around, but it is quite acceptable. Given that I am using a set of test viruses collated in the UK, and *Anti-Virus Plus* is developed in Israel, the accuracy of virus detection is very good, missing less than 1% of the test-set. The ability to detect Mutation-Engine-derived viruses approaches 100%, but needs more work to achieve this target. I have reviewed far worse scanners in the past, and there is nothing basically wrong with *Anti-Virus Plus*.

The other component parts of the software concern me more. On the surface they appear to work satisfactorily, albeit with some minor flaws. However, the lack of documentation about what they do means that neither I nor any other user would be able to tell if they were not performing satisfactorily. This brings me round full circle - the content of the manual needs drastic extending and improving. [*The shortcomings of the documentation are being addressed by Menorah, which is re-packaging the software completely. The new version of the software should be available in the first quarter of 1993. Ed.*]

Technical Details

Product: *Anti-Virus Plus*

Developer: *IRIS Software*, 6 Hamavo Street, Givataim 53303, Israel. Tel +(972) 3 5715319. Fax +(972) 3 318731

Vendor: Menorah Software Ltd., Menorah House, 13 Newton Avenue, Muswell Hill, London N10 2NB, UK.
Tel (and Fax) +(44) 81 883-4269

Availability: PC or compatible, DOS v3.30 or higher

Version evaluated: 4.20.09G

Serial number: IAV11579

Price: £79.00 + VAT

Hardware used: 33 MHz 486 PC, with one 3.5 inch (1.44 M) floppy disk drive, one 5.25 inch (1.2M) floppy disk drive, and a 120 Mbyte hard disk, running under MS-DOS v5.0

4.77 MHz 8088, with one 3.5 inch (720K) floppy disk drive, two 5.25 inch (360 K) floppy disk drives, and a 32 Mbyte hardcard, running under MS-DOS v3.30

Viruses used for testing purposes: This suite of 135 unique viruses, spread across 215 individual virus samples, is the current standard test set. The test set contains 6 boot sector viruses (Brain, Form, Italian, Michelangelo, New Zealand 2, Spanish Telecom), and 209 samples of 130 parasitic viruses. When more than one variant of a virus is included, the number of each virus is shown in brackets. A specific test is also made against 1048 viruses generated by the Mutation Engine.

1049, 1260, 12 TRICKS, 1575, 1600, 2100(2), 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 777, 800, 8 TUNES, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), AntiCAD (2), Anti-Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Captain Trips (2), Cascade (2), Casper, Dark Avenger, Darth Vader (3), Datacrime, Datacrime II (2), December 24th, Destructor, Diamond (2), Dir, Diskjeb, Dot Killer, Durban, Eddie, Eddie 2, Fellowship, Fish 6 (2), Flash, Flip (2), Fu Manchu (2), Hallochen, Hymn (2), Icelandic (3), Internal, Itavir, Jerusalem (2), Jocker, Jo-Jo, July 13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (3), LoveChild, Lozinsky, Macho(2), Maltese Amoeba, MIX1 (2), MLTI, Monxla, Murphy (2), Nina, Nomenklatura(2), Number of the Beast (5), Oropax, Parity, PcVrsDs(2), Perfume, Piter, Polish 217, Pretoria, Prudents, Rat, Shake, Slow, Spanish Telecom (2), Spanz, Subliminal, Sunday (2), Suomi, Suriv 1.01, Suriv 2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Syslock, Taiwan (2), Tequila, Terror, Tiny (12), Traceback (2), TUQ, Turbo 488, Typo, V2P6, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virdem, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Whale, Yankee (7), Zero Bug.

CHKDSK ALERT

A Wolf in the Fold

A major bug has been discovered in certain versions of CHKDSK running under both MS-DOS and PC-DOS Versions 4.01 and 5.0). When running the program with the /F option on certain hard disk configurations, data on the drive will be lost. *Microsoft* has allegedly been aware of this problem for some time (to the extent of releasing a special upgrade version of MS-DOS v5.00A), although it has not made information about this bug public.

CHKDSK is a standard tool which comes with MS-DOS and is used to ensure that entries in the File Allocation Table are valid. The /F option allows CHKDSK to fix any errors it encounters. The FAT is rather like the index to a book - its loss means that the operating system cannot locate data stored on disk.

The story was first made public on the Internet, and rumours quickly spread about this potentially serious bug. Problems arise when running CHKDSK with the /F option on hard disk drives which have been formatted with between 65280 and 65,535 allocation units in machines running either MS-DOS or PC-DOS, versions 4.x or 5.0.

If your hard drive lies in the relatively small danger zone, CHKDSK writes 256 copies of the FAT onto the hard disk - this amounts to some 32 Mbytes of data. This operation will overwrite the system files on the disk, and any programs or data stored in the first 32 Mbytes of the disk. While it is still possible to recover much of the rest of the disk this still represents an unacceptable loss of data.

Microsoft has not really tackled the problem. There has been no widespread alert, and registered users of MS-DOS have not been informed of the bug. It is forgivable that the bug existed in the first place - it is unforgivable that once it was discovered users were not notified. This bug is potentially more damaging than many viruses - data is being put at risk for purely commercial concerns. The length of time between the discovery of the problem and the information becoming public begs the question: what else is rotting beneath the floorboards?

Microsoft's internal reference documenting this bug is located in its Knowledge Base (ref. Q80496) which can be accessed via *CompuServe* (GO MSKB). *Microsoft UK's* PR company, *Text 100*, has played down the significance of the bug, and claims that 'only about ten users worldwide have encountered it.' According to *IBM*, an update fixing the bug has been issued in PC-DOS v5.00.1, and also on all Corrective Service Diskettes numbered 36603 or higher.

CONFERENCE REPORT

'News from a Radiant Future'

Chicago has come a long way since Al Capone; its streets are safe and clean, the buildings outrageously tall and beautiful. With its glamorous setting on Lake Michigan and its culture and night-life, the city is a good conference venue. In November, several hundred delegates descended on the Windy City for the 19th annual *Computer Security Institute* conference and exhibition, a mammoth three-day event covering all aspects of computer security.

While the organisers attempted to make viruses a conference theme, the virus-related sessions were a little banal; no matter how good the speaker, one is disinclined to suffer yet another explanation of how viruses replicate. The 'Virus Management Day' included a 'virus testing laboratory' in which increasingly glum-faced vendors (for they had paid dearly for the privilege) implored delegates to submit disks for checking. A total of one diskette was forthcoming, which the owner already knew to be infected.

The exhibition community was interesting; companies promoting anti-virus products included *Cheyenne Software*, *Command Software* (*FRISK*), *Commcrypt Inc*, *Datawatch* (*Microcom*), *Digital Equipment Corporation* (*Sophos*), *Leprechaun Software*, *NetPro Computing* (*McAfee*), *Network-1 Inc*, *PC Guardian*, *RG Software*, *Safetynet Inc*, *Symantec*, *Trend Micro Devices* and *Xtree*.

The Sense of Humour Prize goes to Roger Thompson of *Leprechaun Software*, who sat cheerfully in an empty booth space in front of a sign which read 'Virtual booth (engage imagination)'. The Low-Key Marketing Prize goes to *Digital* which promoted anti-virus software without the word 'virus' anywhere on its large and elaborate booth. The Silly Claims Prize is taken by *Trend* - 'Removes all known and unknown viruses!', while the politically incorrect Least Facially Challenged Salesperson Prize goes to *RG Software* and its secret weapon, the charming Miss Danette Ripper.

Man cannot live by viruses alone: so *VB's* correspondent braved the North wind to the *Chicago Institute of Art* and an exhibition of Soviet porcelain from the 1920s. 'News from a Radiant Future', as one artist described this work, consists for the most part of exquisite dinner plates decorated with Marxist-Leninist motifs and slogans. Such skill applied in a misguided cause is reminiscent of the current crop of Russian viruses. 'KTO HE PAbOTAET, TOT HE ECT', says one such plate, quoting Lenin: 'He who does not work, does not eat'. This is certainly true of the virus researcher; rumours abound of some 1100 new Russian viruses. If only they would make fine porcelain instead!

END-NOTES AND NEWS

***Central Point Software Inc.*, has released a free virus scanner**, *Central Point Anti-Virus Scan-Only System* (*CPAV-SOS*). The program is being distributed by *Central Point's* own BBS, *Compuserve*, and various other BBSs. After downloading the program individual users may make unlimited copies of the scan program, and use it freely for virus detection. For further information contact Diane Paternoster. Tel 081 848 1414. Alternatively, download the package directly from the *Central Point* BBS. Tel 081 569 3324.

The DOSHUNTER virus is reported to be 'in the wild' in the Netherlands, with at least 41 confirmed reports of the virus. It was discovered when users noticed erratic behaviour of their machines. *IBM* has issued a minor alert over the situation.

***Symantec* has signed a series of bundle deals to supply many of its utilities pre-loaded onto PCs.** All desktop and notebook PCs manufactured by *Akhter Computer Systems* and *Ti'Ko Computer Corporation* will now have *The Norton AntiVirus* pre-installed on them at no extra cost. For an additional £100, however, the users will be provided with updates and technical support. For more information contact Gideon Luke. Tel 0628 776343. [*At the time of going to press, Akhter Computers is suing Symantec for breach of contract. This story of unbridled passion will be reported in detail next month. Ed.*]

The *National Computer Security Association* (*NCSA*) has announced the availability of a generic 'Corporate Virus Prevention Policy'. This document has been designed to help anyone who is responsible for drafting a corporate-wide anti-virus policy. The document is available in a variety of formats ranging from printed ($5) to WordPerfect format ($12.95). Tel (1) 717 258 1816.

***Leprechaun Software* has announced a new hardware based anti-virus product**. Before installation, the hard disk is partitioned into two drives, C: and D:. All executables that the user wishes to protect are placed on the C: drive. The device, C:CURE, is then placed between the hard drive and the IDE interface card, and prevents all writes to the C: drive, while leaving the D: drive unchanged. *Leprechaun Software*, USA. Tel (1) 404 971 8900.

***Fifth Generation Systems* has introduced its Virus Insurance Protection Program** (VIP). The package is designed to keep customers up to date, and includes regular signature updates, a 24-hour virus hotline, and a 48-hour emergency service. Any user with a virus problem which they cannot solve can send the virus to *Fifth Generation*, who will provide a full diagnosis within two working days. Further information is available from *Fifth Generation Systems*, USA. Tel (1) 504 291 7221.

The *Business Software Alliance* has begun a worldwide campaign to shut down illegal computer Bulletin Boards. The BSA has announced the first results of the campaign - a sweep by the Berlin police of illegal BBS operators throughout the city, seizing equipment at 13 BBS operations that have been distributing illegal software in Germany. Tel 071 491 1974.

Virus News International is set to abandon the handy, loose-leaf, pre-perforated, soft-paper format which has helped build its loyal user base. Regular recipients should be aware that this product will be supplied in A4 size from January 1993.

VIRUS BULLETIN

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 Science Park, Abingdon,
OX14 3YS, England

Tel (0235) 555139, International Tel (+44) 235 555139
Fax (0235) 559935, International Fax (+44) 235 559935

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA
Tel 203 431 8720, Fax 203 431 8165

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