ISSN 0956-9979 APRIL 1995

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Inside ESaSS.** Frans Veldman is one of the principal designers and developers of *TBAV*, one of the fastest virus scanners on the market. Discover his views on computer viruses, product development and the underground on p. 6.

• **Comparatively speaking.** Perhaps the most critical point of your virus countermeasures is network protection. Jonathan Burchell puts the leading products through their paces against the same test-set which their DOS counterparts faced in January this year. Some readers and developers may be in for a surprise…

• **The road to recovery.** Although much of the emphasis in any anti-virus policy is on prevention, there is a great deal one can do to prepare for the worst before it happens. Some food for thought is on pp.12-14.

CONTENTS

EDITORIAL

Moving On

After 26 editions of *Virus Bulletin*, it is time for me to move on… this month's magazine is the last one on which I will wield the Editor's red pen. I leave *VB* in the safe hands of Ian Whalley (who will introduce himself to you next month), but suffice it to say that I have no concerns about the future of *VB*: the Editor elect is champing at the bit, and I await future issues with interest.

When I took over the job of editor from Edward Wilding, I was determined that *Virus Bulletin* should continue to report the truth as it found it. I hope that, despite legal threats and sabre-rattling, it has done just that, and I am confident it will continue to do so. While the anti-virus culture is much more 'corporate' than it was in the early years of *Virus Bulletin*, there are still quack doctors and snake oil salesman, ready to snatch the money of the unwary, and *VB* will continue to cast its watchful eye over new products and services.

"we need to push for the inclusion of courses in computer ethics in schools"

As editor, I have steered clear of making predictions, lest they come back and haunt me at some later date. However, one prediction which I will now make is that the virus problem is very definitely here to stay, and that the fundamental reasons which have caused it must be tackled.

Over the past two and a half years, a lot has changed in the industry. Oddly, much has also remained the same. All the same names are still involved (to a greater or lesser extent), and many of the products have changed very little in their appearance - at least, those which have survived! Throughout this, the virus problem has remained, and we are still no closer to a real solution.

In 1992, when I wrote my first article for *VB*, I believed the entire problem could be solved by tough legislation, designed to curb the distribution of virus code. Now, 26 editions wiser (?), I have lost that delightful certainty. Legislation is difficult, if not impossible, to enforce, and can only be effective when it represents the views and beliefs of the majority of society. Right now, the average man in the street probably has little or no idea of what a computer virus is, or how it functions.

While I do not claim that the entire solution lies in education, I now strongly believe that this is the direction in which we need to move. As an industry, and as representatives of large companies, we need to push for the inclusion of courses in computer ethics in schools, and also for the demystification of computers to the layman. With computers directly involved in so many aspects of our lives, threats to our IT resources can quickly become a threat to us. The sooner this education process can be begun in earnest, the better.

Although I am very much looking forward to moving on to pastures new, I will admit to a certain amount of regret as I type this last editorial. I have had a most enjoyable time, and have met a wide cross-section of people in the course of my job, ranging from the downright malevolent to the truly precious - and everything in between. In the course of my job, I've been arrested (twice now - see *Virus Bulletin* May 1994, pp.15-16), juggled at, and evacuated... never ever let it be said that editing is just about writing!

I sincerely thank all the staff and contributors who write the bulk of every edition, as well as Fridrik Skulason, for his support, level-headedness and absolute reliability, all those on the advisory board, and those within the industry who never tire of suggesting articles and additions. However, the most important contribution towards the magazine comes from its readers, through their telephone calls, letters and faxes. *Virus Bulletin* can only offer a valuable service if its readers continue to give their comments - praise for what is done well, and criticism for those articles which are not found valuable. It is this feedback which I have found most useful in my time as editor, and I urge you to continue to keep it coming.

For those comments, good and bad, I thank you.

Richard Ford
Editor

NEWS

A Poxy Misjudgement

Roger Riordan (*Cybec Software Pty*, Australia) has sent in a report on a 'Trojan which was not a Trojan'. The program, known as 'Burn in Hell', is also in the wild in Europe.

Late last year, *Cybec* received information on the incident from Mikko Hypponen (*Data Fellows*, Finland), and Vesselin Bontchev (*Virus Test Centre*, Germany). According to their reports, some customers' PCs would display the following message on boot-up:

```
A poX oN yOu!! yoU wiLl bUrN iN tHe fiReS of
HeLl!!
```

The message was eventually traced to a device driver named CMD640X.SYS, written by *CMD Technologies*. In due course, a sample was sent to *Cybec*, whose technicians were unable to determine how it was triggered. They knew, however, that the message had been seen in Europe, and the wording strongly suggested that it had been included with malicious intent.

Riordan and his colleagues had already seen two cases where PCs contained Trojans inserted during manufacture by disaffected employees. They assumed this to be simply another such program, and added a new signature to the next release of their product, *VET* (version 8.1).

Cybec was immediately inundated with reports from users who wanted to know why they were receiving alerts of the Trojan on PCs which everything else considered clean. *Digital Equipment Corporation* (*DEC*) also received many complaints, as the file was found on all of one model of PC made by them.

Discussions between *DEC*, *CMD*, and *Cybec* established that the Trojan was intended as an anti-piracy measure: the message was displayed if the visible copyright message did not match a hidden second copy. *CMD* has undertaken to supply a new version of the driver, without the message.

'This case demonstrates the dangers of using an inappropriate message,' said Riordan. 'The chosen message neither informed the user that he was using pirated software, nor the manufacturers that their software was being pirated. Instead, it caused everybody to assume that the software had been deliberately corrupted. However, if the message had simply stated "FOOBAR.SYS appears to be corrupted; please contact *Foo Inc.* on phone XXXX, fax XXXX", the firm would have been alerted if the file had been pirated, without the (presumably innocent) user being alarmed.

'It is extremely unwise ever to include a message accusing a user of piracy,' he continued. 'There is always the possibility that the message will be displayed accidentally - and there are few more effective ways to upset a user who has paid for your software.' ▮

Virus Prevalence Table - February 1995

Virus	Incidents	(%) Reports
AntiEXE.A	14	15.9%
Form	10	11.9%
Parity_Boot	9	10.2%
Monkey.B	7	8.0%
AntiCMOS	5	5.7%
JackRipper	5	5.7%
Jumper	4	4.5%
Natas	4	4.5%
Sampo	4	4.5%
AMSE	3	3.4%
Exebug.A	2	2.3%
Junkie	2	2.3%
Monkey.A	2	2.3%
Spanish_Telecom	2	2.3%
V-Sign	2	2.3%
Cascade.1701	1	1.1%
Crazy_Boot	1	1.1%
Jimi	1	1.1%
Lixi	1	1.1%
Peter-II	1	1.1%
Quox	1	1.1%
Stealth.B	1	1.1%
Stonehenge.B	1	1.1%
Tai-pan	1	1.1%
Tequila	1	1.1%
Wcwa	1	1.1%
XOR_Boot	1	1.1%
ZOID.1759	1	1.1%
Total	88	100.0%

'Black Baron' to Reappear in Court

Readers have been following with interest the case of the man alleged to be the 'Black Baron', author of the SMEG engine and related viruses. *Virus Bulletin* has been reporting on the incident since the story broke in February 1995. The man, Christopher Pile of Plymouth, appeared in *Plymouth Magistrate's Court* for a second time on 4 April: the case was adjourned, and will be heard again on 18 April 1995.

DS Simon Janes, of *New Scotland Yard's Computer Crime Unit*, said: 'It is anticipated that he will be committed to the *Crown Court* at his next appearance.' Anyone with information which might be relevant to the case is urged to contact the *CCU* on +44 (0)71 230 1177 ▮

IBM PC VIRUSES (UPDATE)

The following is a list of updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 21 March 1995. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

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

AntiCad.4096.K **CER:** Detected with the Anticad.generic pattern.

Ash.280.B **CN:** A minor variant, detected with the Ash pattern.

Beer.3192.B **CER:** A minor variant, detected with the Beer-2192 pattern.

Burger **CN:** There are three new variants of Burger this month: 441.C, 560.AZ and 560.BA. They are all detected with the Burger pattern.

BW.Mayberry **CEN, CER, CN, CR, EN:** One group of viruses was originally named the Mayberry family, but has now been reclassified as a group of BW-generated viruses. The origin of those viruses is rather interesting: according to the author, they were originally created to test 'leaks' from a specific anti-virus company into the VX community. The samples were created, uploaded to that particular company, and within a few weeks they appeared on the first VxBBS. Whether this story is true, or is pure fabrication, intended to discredit that particular company, is not known for sure. The viruses in this group were originally given specific names, but have now been assigned numeric names, which indicate infective length. The Mayberry viruses are 402 (CN, 'Velmalou'), 409 (CR, 'Opy'), 475 (CN, 'Jethro', encrypted), 496 (CEN, 'Barney', encrypted), 502 (EN, 'Floyd'), 609 (CER, 'Andy'), 687 (CEN, 'Gomer'), 732 (CEN, 'Aunt B'), 747 (CEN, 'Otis', encrypted), 758 (CER, 'Ms. Crump') and 828 (CER, 'Goober', variably encrypted - no simple search string is possible)

```
BW.Mayberry.402  B802 3DCD 2172 6E93 B800 57CD 2151 52B4 3FB9 1C00 8D96 9502
BW.Mayberry.409  B802 3DCD 2172 6493 0E1F B800 57CD 2151 52B4 3FB9 1C00 BA9C
BW.Mayberry.475  BB?? ??BE ED00 2E81 2F?? ??83 C302 4E75 F5
BW.Mayberry.496  BAF7 00BB ???? 2E81 37?? ??43 434A 75F6
BW.Mayberry.502  B802 3DCD 2172 3593 B800 57CD 2151 52B4 3FB9 1C00 8D96 F902
BW.Mayberry.609  B802 3DCD 2172 6493 0E1F B800 57CD 2151 52B4 3FB9 1C00 BA64
BW.Mayberry.687  B802 3DCD 2172 6E93 B800 57CD 2151 52B4 3FB9 1C00 8D96 B203
BW.Mayberry.732  B802 3DCD 2172 6E93 B800 57CD 2151 52B4 3FB9 1C00 8D96 DF03
BW.Mayberry.747  BB?? ??B9 7501 2E81 07?? ??43 43E2 F7
BW.Mayberry.758  B802 3DE8 44FF 726A 930E 1FB8 0057 CD21 5152 B43F B91C 00BA
```

Cascade **CR:** There are two new Cascade variants this month: 1701.AE and 1701.AF. They are both detected with the Cascade(1) pattern.

Clonewar **P:** In addition to a 547-byte variant, which is almost identical to the 546-byte variant reported earlier, there is a group of 923-byte variants. Each is a minor variant, and differs only in small details. The first '923' pattern below will detect the A, B, C and D variants, the second can be used for the E, F and G variants, and the last one can be used for the H variant. By discarding the last byte, the same pattern can be used for all the 923-byte variants.

```
Clonewar.546      93B9 2302 BA00 01B4 40CD 21B4 3ECD 21BA 5702 B903 00B8 0143
Clonewar.923  (1) BA1A 01B9 0000 B800 3DCD 21C3 BA1A 01B9 0000 B43C CD21 725F
Clonewar.923  (2) BA1A 01B9 0000 B800 3DCD 21C3 BA1A 01B9 0000 B43C CD21 7260
Clonewar.923  (3) BA1A 01B9 0000 B800 3DCD 21C3 BA1A 01B9 0000 B43C CD21 7262
```

Dark_Avenger.1800.O **CER:** Detected with the DA-related pattern.

Ear **CEN:** There are two new variants of this virus: Ear.1024.C and 1026. Both are detected with the Ear-6 pattern. Their variants are all flawed in the same way, and will damage almost every EXE file they attempt to infect.

Fax_Free **ER:** A large number of new variants in this family has appeared over the last year - mostly coming from Italy, although the virus-writing activity there seems to have decreased considerably in the past few months. The 1536(1) pattern below is a 'generic' pattern which will detect 1536.Pisello, 1536.Pinniz.A, 1536.Pinniz.B, 1536.Pinniz.C, 1536.Pinniz.D and 1536.Pinniz.E.

```
Fax_Free.608.A    80FC 3B74 E580 FC4B 757D 0E1F B860 022E A34F 021F B800 43CD
Fax_Free.608.B    80FC 4B74 03E9 5D01 0E1F B860 022E A34F 021F B800 43CD 2173
Fax_Free.622      80FC 4B75 7C0E 1FB8 6E02 2EA3 5D02 1FB8 0043 CD21 726B 2E89
Fax_Free.623      80FC 4B75 7D0E 1FB8 6F02 2EA3 5E02 1FB8 0043 CD21 726C 2E89
Fax_Free.1536(1)  0026 8706 0C00 508C C826 8706 0E00 50CC 589D 5826 8706 0E00
```

Gotcha **CER:** There are two new variants this month: 828 and 1778. Both are detected with the Gotcha-D pattern.

HLLO.8608 **EN:** An unremarkable overwriting HLL virus.

HLLP.8304 **CEN:** A prepending HLL virus.

Icelandic.1600 **ER:** Similar to the other 'Mix' variants, and detected with the Mix1 pattern.

Intruder **EN:** There are two new variants of Intruder this month: 1336 and 1353. Both are detected with the Intruder pattern.

IVP **CEN, CN:** There are a few new IVP-generated viruses this month: 365 (CEN, overwriting), 510 (CN) and Thursday (CEN, 675).

Jerusalem.1808.Blank.D **CER:** A minor variant, detected with the Jerusalem-1 pattern.

Jerusalem.Sunday.O **CER:** Detected with the Sunday pattern.

Kaos4.C **CEN:** A partially-corrupted, minor variant, detected with the KAOS4 pattern.

Murphy.Tormentor.1072.C **CER:** Minor variant, detected with the HIV pattern.

NRLG **CR:** The number of NRLG viruses continues to grow. This month brings the following variants: 666, 813 and 1001.

Proto-T **CR:** There are two new variants of Proto-T this month; 602, which is detected with the Proto-T.515 pattern, and 654, which is detected with the Proto-T pattern.

PS-MPC It should not come as a surprise to anyone that there are several new PS-MPC-generated viruses this month. The only interesting one is PS-MPC.DemoExe.32947, which should really have been 179 bytes long, but one bit has been changed in the virus, resulting in a huge difference in infective size. The virus is actually flawed, and will damage most of the files it attempts to infect. The new PS-MPC viruses this month are: 310 (CN), 441 (EN), 487 (CEN), 574.F (CEN), 598.E (CEN), 598.F (CEN), 1295 (CER), DemoExe.32947 (EN), DK.693 (CER), Dork (EN, 553), Mema.1187 (CR), Mema.1201 (CER), Mema.1203 (CER) and Mema.1217 (CER).

Sentinel.4636.B **CER:** A minor variant, detected with the Sentinel pattern.

Shirley.C **ER:** A minor variant, detected with the Shirley pattern.

Skater **CR:** This is a family of four viruses of Australian origin, some of which contain the text 'Australian Parasite'. Two of them (977 and 1021 bytes long) mention Tonya Harding, hence the family name. The third (699 bytes long) mentions Patsy Cline, and the fourth (714 bytes long) contains no text messages. Those four viruses are placed in a separate family instead of being grouped together with the other Australian_Parasite viruses, because of the unusual encryption method they use: Triple XOR, with a constant, a decreasing value and an increasing value. The encryption is also slightly variable, and a search string for the viruses, while possible, would contain a very large number of wildcards, making the risk of false positives unacceptably high.

Stinkfoot **EN:** There are two new Stinkfoot variants this month: 1283.A, which is detected with the Stinkfoot pattern and 1283.B, detected with the Stink-D pattern.

Sylvia.1332.F **CN:** A minor variant, detected with the Sylvia pattern.

VCL Despite the age and the flaws of the VCL tool, and the fact that most anti-virus products will detect all VCL-generated viruses, the number of new VCL-generated viruses is still growing. Recently, a number of variants have appeared which contain the string '[VCL-MUT]'. As the text implies, they are not regular VCL-generated viruses, but closely related - perhaps generated by a 'mutation' tool. The VCL 'companion' viruses this month are: 208, 279. 315, 316 and Heevahava.520. The new overwriting viruses are: 288, 302, Mindless.423.F, Mindless.423.G, Monet.267 and Monet.466. Finally, the COM-appending viruses are: 2037, Catholic (1230), Genocide.981, Grail (1341) and Lobo (704).

INSIGHT

Going Dutch

Megan Palfrey

Frans Veldman is a name which is not particularly well-known outside the anti-virus industry. However, if one mentions the name '*ThunderBYTE*', a product renowned for its consistently high detection rates and breathtaking speed, bells start to ring. It is Veldman, in conjunction with Robin Bijland, who writes and develops it.

In the Beginning…

The two men, friends since high school, own and run the Netherlands-based *ESaSS BV*. They set the business up in 1987 to respond to a need they perceived in the 'grey area' between hard- and software, developing special equipment on demand. Veldman was not originally a virus researcher - it was not until a 'business relative' of theirs became infected via 'rather a large company' that Veldman had any direct contact with viruses. After disassembling and analysing that virus, he decided to write a program to cure it.

He realised that everything he wrote into the program would be countered by the virus if it were already in memory: 'It is important to be active before the virus is executed. The best way to make your program active is by a ROM BIOS; then you are guaranteed to be first in memory, even when you boot from a diskette.'

'A ROM BIOS cannot be changed by software, which is a big advantage. Since we already had experience in designing PC hardware cards, we decided the next logical step would be to develop an anti-virus hardware card. This was six years ago!'

The hardware anti-virus product fitted in well with their business profile. From a technical point of view, there are both advantages and disadvantages to using a hardware-based solution: on the down side, the PC must be physically opened in order to install the card - 'A bit of a burden if you have to install it on 5000 PCs,' admitted Veldman - compatibility problems occur more often, and it is more expensive.

'We quickly learned,' he said, 'that many customers preferred a software solution, so we developed *TBAV*. Our sales grew as soon as we released it. Today, we still sell the hardware card, but the majority of our customers go for the software.'

'When I started to develop the card, viruses were still quite rare, and difficult to obtain. Only a few people had any kind of collection, and those who did were often not willing to give them away. So I didn't look much to the few viruses I had, but just imagined how they might work, what their common behaviour patterns would necessarily be, and, most importantly, how viruses could be distinguished generically from sound application programs.'

'Actually, I have never made a real disassembly of a virus! I often peek inside them using a debugger, to see what they do, but I'm not that interested in the tiny details.'

Branching Out

Since those early days, the company has grown apace, and has already had to relocate to larger premises twice. Although *ThunderBYTE* is still their core product, they also own other companies in the computer security field.

'From a business point of view,' he explained, 'it is important not to rely on just one product. In the software business, you can never know what will happen in the future - we don't even know what operating system people will be using two years from now! As long as the development of our anti-virus product is not affected, there is nothing to be said against developing new products. In fact, this sort of diversification secures the continuity of the company, and therefore also of the anti-virus product.'

Veldman's main interest in anti-virus research is heuristics; the non-specific detection of computer viruses. *ESaSS* is one of the first 'big league' developers to have invested heavily (in terms of time and effort) in the area. His goal is to make their product as virus non-specific as possible.

'Heuristic detection is an excellent way to enable a product to find a majority of yet-unknown viruses,' he explained. 'At the moment, we see 100-200 new viruses each month. If you have a monthly update schedule, by the end of the month there are about 100 viruses out there that you don't detect. Heuristics fills this gap. With the expected increase of viruses, heuristics becomes more and more important.'

The product is continually being developed and improved - a recent addition was the implementation of a Generic Decryption Engine, which allows the heuristic engine to look inside encrypted viruses.

The Underground Subculture

The computer 'underground' seems to have almost an obsession with *ThunderBYTE*; playing with it, writing viruses targeted to defeat it - Veldman was worried and angry when the situation first became apparent, but his views have changed with time.

'Virus writing is often a competition against the anti-virus industry, so virus writers target their programs against what they consider to be the strongest anti-virus product. Virus authors consider our product a difficult one to cope with, because of the heuristics, the generic decryption engine, and anti-stealth techniques. It is no fun to avoid detection by a "normal" scanner - every new virus will go by undetected.

Frans Veldman: 'Companies which are able to play the virus-specific detection game for the longest time have the best chances for survival.'

'However, because of the heuristics, *ThunderBYTE* is often able to detect a new virus even before it has been released. There is a lot to explore with my product, and the code is hard to unravel. It just keeps them busy. I consider the infatuation of the underground with *TBAV* [*ThunderBYTE Anti-Virus*] as some kind of virus writer-award, maybe the highest award you can get as an anti-virus developer.'

Veldman is conscious of the fact that the computer underground consistently returns to his product, but has turned the situation to his advantage: 'I consider them almost as beta testers; I'm playing a game with them. I love to read in their magazines how much time they waste trying to find out how some things work: they can't write viruses at the same time! Of course they sometimes discover things I would rather keep secret, but on the whole the underground helps me improve my product; they point out loopholes - they prevent me falling asleep. I see their obsession almost as a kind of quality assurance.'

He has learned, he says, to play a psychological game with the underground, a 'cat-and-mouse' of programming. He made his generic decryptor, for example, only just powerful enough to deal with known viruses, commenting out much of the code which was not needed at the time. The underground invested considerable time and effort in developing a new encryption to foil *TBScan's* attempts to detect it - this, however, was exactly what Veldman had anticipated.

'It took only one minute,' he remembered, laughing, 'to enable the code which I already had available. That made their new encryption engine useless.'

That particular situation has happened more than once with *ThunderBYTE*. Veldman's approach is to enhance the capabilities of the product, but only slowly, and never more than required to detect new viruses. This makes the virus underground waste a great deal of time and effort writing programs to defeat the latest version of *TBAV* - but Veldman then needs only minimal time to repel their latest attack.

'People are afraid,' commented Veldman, 'that when the underground discovers something new in anti-virus techniques - for example, how our heuristics works - my product will become weaker. While I have to admit it is possible to escape heuristics, I do not see this as a problem.' Although techniques exist which make life more difficult for signature scanners, heuristics, and behaviour blockers, Veldman is confident that writing a virus which is capable of exploiting all these techniques at the same time is more complex.

'Polymorphic viruses complicate the life of signature scanners,' he stated, 'but it is exactly this polymorphism which makes them suspicious to heuristic analysers. Viruses which escape the attention of a heuristic analyser can be found easily with a conventional signature. Heuristics adds to the complexity of writing viruses, which indirectly slows the increase in the number of viruses. Isn't that our goal?'

The 'discoveries' of the virus underground, in Veldman's opinion, have a smaller impact than many people think: 'Despite the anti-heuristic techniques used by virus authors, our product can still detect the majority of new viruses.'

The Way Forward

Veldman believes that virus-specific detection will ultimately die. Even if a product can detect as many as 1000 new viruses each month, a monthly update might still miss about the same amount at the end of the update cycle. When this sort of growth is recorded, he feels, the best virus-specific countermeasures would be daily updates. This, he readily admitted, is simply not a practical solution.

'The market, however, is heavily focused on scanners,' he commented ruefully. 'Companies which are able to play the virus-specific detection game for the longest time have the best chances for survival. At the same time, whether any such company wins that game or not doesn't affect the fact that you have to prepare a virus non-specific product.'

The Other Side

Veldman is not totally consumed by computers and viruses, thankfully - he leads a full and interesting life outside his work. He lives in the countryside, in what he describes as one of the most beautiful parts of the Netherlands, and is a nature-lover. He is also a radio enthusiast, and is eagerly awaiting his licence to become a HAM radio operator.

In tandem with this, he is taking a diving course, and is an avid fan of old *Mercedes* cars. Where time permits, he has various other interests, including his animals: tropical fish, several cats, and a Kuvasz dog (excellent, he says, for guarding the virus collection).

The next generation of the family is already computer-literate: Veldman's one-year-old daughter enjoys nothing so much as chewing diskettes and resetting her daddy's server. Doubtless, however, he will find a way to turn even her games to his advantage!

VIRUS ANALYSIS 1

2UP-manship

Eugene Kaspersky
KAMI Associates

A new PC 'monster' has appeared in the wild in Russia: a 6000-byte parasitic COM and EXE file infector which is called 2UP (after internal text strings). First reports on this virus indicated that it is written in a high level language - viruses over 3K long are usually put together with Pascal, Basic or C compilers, but 2UP is written in Assembler. This virus requires 6K for its code, in addition to 18K of system memory (which is allocated on installation) to install its routines and keep places for data buffers.

Installation

2UP is encrypted, and execution of an infected file passes control to the decryption routine. A simple XOR loop restores the virus body to its original unencrypted form, following which control passes to the installation routine.

The installation routine checks system memory with an 'Are you there?' call for a TSR copy of the virus. 2UP finds the address of Int C5h with the GetVector function (Int 21h, AX=35C5h). If that vector points to segment 5CA6h (indicating that the virus is active in memory), 2UP restores the host code to its uninfected form and passes control to it.

If the call returns any other value, the virus begins to install itself into memory. 2UP records which version of DOS is installed (storing it for future use), traces Int 21h to look for the original Int 21h DOS handler, allocates system memory using the Int 21h calls ChangeMem (4Ah) and AllocMem (49h), and copies itself into the allocated memory block. The memory manipulation block is executed in two parts: the first before control passes to the host program, and the second after host program termination. The virus temporarily hooks Int 22h (Program Termination Address) to intercept the termination of the host program.

Hooking Int 22h allows the virus to move itself to other addresses after termination of the host program. When an infected program is first executed, it copies itself into the top of conventional memory. After termination of that program, the virus moves itself down to lower addresses.

2UP hooks Int 21h to intercept file access functions for infection and stealth, in a manner similar to, but more complex than, that of Frodo (aka 4096, 4K). It traces Int 21h for the CMP instruction usually found at the beginning of the Int 21h handler, and replaces five bytes of that handler with a FAR JMP instruction to the virus. This tracing routine is too complex to function without problems: in my tests, 2UP halted the system if certain configurations of system drivers and memory managers were used.

All functions of the virus' Int 21h handler are encrypted with an 'on-the-fly' encryption/decryption algorithm - this will make it more difficult to detect 2UP in memory. Thus, each time the virus' Int 21h handler receives control, it has to decrypt itself before any further action is taken.

Infection

2UP prepends its own code to the start of infected COM and EXE files. In the case of COM files, it simply writes itself in front of the file body, as do the Jerusalem viruses. Where an EXE file is being infected, 2UP leaves the EXE header in the same location, moves the host file's code and data 6000 bytes down, and writes itself at the addresses between the EXE header and the code of the program being infected. The virus also corrects the EXE header fields on infection, altering the initial values of the SS, CS and IP registers, and modifies the module length fields and corrects the relocation table, so that processing passes to its own code.

2UP's manner of infection is unusual, but less so than the way in which it intercepts Int 21h functions to call the infection routine. Three Int 21h functions are hooked, CreateFile (AH=3Ch), WriteHandle (AH=40h) and Exec (AX=4B00h). However, the virus only writes its code into a file on Write Handle calls. When a new file is created, the virus checks its name, infecting only if it does not begin with one of the following strings:

```
AID COMMAND ANTI AV HOOK SOS TSAFE -V SCAN
NC VC TNT ADINF
```

i.e. AIDSTEST.EXE, COMMAND.COM, ANTI*.*, etc. It then calls the original Int 21h to create the file, stores its handle and returns.

The CreateFile function is usually called on file copying. A new file is created, and data from the source file is written to the new file block by block. As infection is carried out when the first block of the file is written, subsequent writes to the file will not (in most cases) infect it.

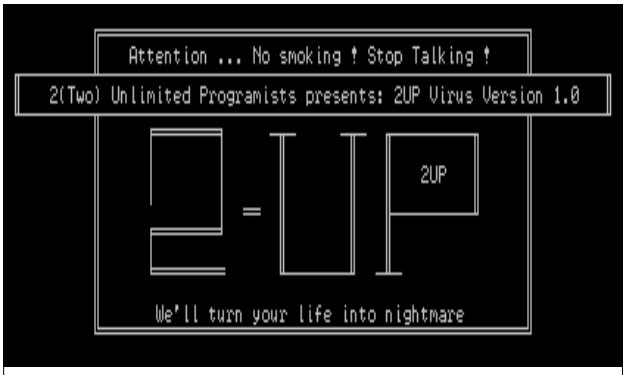


Figure 1: The message displayed by the virus when AID*.* files are executed.

To ensure that the executable, and not the data file, is being infected, the virus checks the file name and header. It will infect only files with the extension COM, or those with the word 'MZ' (which identifies an EXE file) at the header.

When such a file is executed (Int 21h, AX=4B00h), the virus infects it, using the CreateFile hook. It creates a new temporary file, OBJXCREF.COM, copying the original file into it. This file create call and the subsequent write call are also intercepted by the virus, with the result that the file OBJXCREF.COM contains an infected copy of the original file. Then the virus deletes the source file, renaming OBJXCREF.COM so that it has the name of the source file.

Self-recognition and Stealth

The infection process is not the most unusual aspect of 2UP: that honour goes to its self-recognition algorithm, which is divided into two parts. The first is very ordinary, and simply compares the file's beginning with 15 bytes of virus code.

The second, however, is one of the most complex I have ever seen. When a file is infected, the virus modifies the directory sectors on disk, writing the virus' ID stamp into the directory entry of the file. These entries occupy 20h bytes per file and contain such information as the file name and its length, the date and time stamp, and the number of the disk cluster on the first file.

When the file is closed after infection, the virus calculates the address of the sector containing the directory entry of the file. Ten bytes are reserved in the file's directory entry which are usually filled with zeros, and 2UP overwrites these reserved fields with the string '2UP(C)1994'.

On accessing the file, the virus reads its directory entry to check for the presence of an ID string, used on DOS calls FindFirst/Next (the virus intercepts AH=11h, 12h, 4Eh, 4Fh functions, decreasing the lengths of infected files), and on file execution (AX=4B00h). This should prevent multiple infection of any one file.

The file body is compared with the virus code on Write (AH=40h) calls to skip already infected files, and on ReadHandle (AH=3Fh) calls. When 2UP reads an infected file, it attempts to decrypt the body of the file, and compares this decrypted code with its own.

Trigger Routine

The payload may trigger as soon as the virus receives control. While installing itself into system memory, the virus calls Int 21h with AX=F666h; if the interrupt handler returns 4F6Bh in the CX register, the virus displays this message:

```
Hello BOBBY ! (BOBBY-Trash Soft & Hardware)
```

If an error occurs when system memory is allocated for the virus' TSR copy, the video effect routine is called, and a line on screen is selected at random. Characters are raised from their places by a few lines, then drop back, line by line.

The video effect routine is also called from other parts of the virus code, usually if an error has occurred on certain operations; for example, when a file is closed, and on memory allocation during infection. The routine can also be executed when a file is opened. Whether this happens is dependent on system date and time - the conditions are that the month is an even-numbered one (February, April, June, etc.), and the current seconds value is 58 or 59.

On execution of the AID*.* files (AIDSTEST.EXE) on the same date and time as above, the virus displays the message shown in Figure 1.

In some cases, 2UP overwrites newly-created data files with the same strings as above. The virus also contains the following internal text strings:

```
Hullo ! Welcome to 2UP virus. Don't try so
hard! Hallo Mr.Virusolog,now you decod me
! It's about fucking time.What do you
think about 2UP Virus ? This Virus Was
Designed in 1992-1994 .It Dedicated For
Nobody.. I Want To BreakFree ! Right Now
.com.COMobjxcref.com 2UP(C)1994 .EXE.COM
```

2UP

Aliases:	None known.
Type:	Memory-resident, parasitic, encrypted file infector with stealth capabilities.
Infection:	COM and EXE files.
Self-recognition in Files:	a) compares the first 15 bytes of its code with the file's entry. b) compares reserved fields of file directory entries with the string '2UP(C)1994'.
Self-recognition in Memory:	On 'Are you there?' calls with Int 21h, AX=35C5h (Get Vector C5h), the memory-resident virus returns 5CA6h in the ES register and the segment value of its TSR copy in the BX register.
Hex Pattern (in files and in memory):	E851 00EB 6290 33C0 8EC0 BD6C 0426 8A56 002E C606 1305 0090
Intercepts:	Int 21h for stealth, for infection, and for the trigger routines Int 22h (Program Termination Address) for installation in system memory.
Trigger:	Displays messages, 'drops' letters onto the screen, overwrites data files.
Removal:	Under clean system conditions, identify and replace infected files.

VIRUS ANALYSIS 2

Sampo Revisited

Dmitry O Gryaznov
S&S International plc

[*In March 1995, Virus Bulletin published an article entitled 'Sampo: Packing a Wllop?' Dmitry Gryaznov has since kindly supplied the following additional details on how this virus works. Ed.*]

Getting Started

Sampo (aliases Turbo and Wllop) is a Master Boot Sector (MBS) or Partition Table (PT) sector infector. It allocates 6KB of conventional memory just below the 640KB limit of base memory, using the technique most often found amongst common MBS viruses - decreasing the ROM BIOS memory limit value at offset 0413h in segment 0. The virus then copies itself to this allocated memory, beginning to show its interesting and distinguishing features even before the memory allocation.

Several other MBS/boot sector infectors are known to this virus. One of these is Stoned (and most of its variants) - I did not have time to unravel what the other three viruses were. Sampo first checks to see if any of these viruses are already resident, comparing two first bytes at the top of memory to the first bytes of the viruses.

Recycling and Infection

If any of these viruses are found, Sampo uses the values they have obtained, such as the address of the original (ROM BIOS) Int 13 handler, etc. The other virus(es) will already have modified the interrupt vector table, and when Sampo copies itself to the top of memory, it will overwrite the interrupt handler code installed by the other virus. If it first obtains the original interrupt vector from the body of the other virus(es), it can patch the vector table to point to itself, bypassing the other virus(es).

Sampo can recognise itself in memory by comparing 27h bytes of its code with whatever happens to be at the top of memory. This self-recognition is essential, as the virus is able to survive a warm reboot (Ctrl-Alt-Del).

Last month's article incorrectly described Sampo's use of Int 09h. This virus does indeed scan for a keypress, but the keypress in question is Ctrl-Alt-Del, not Ctrl-Alt-S. The reason for the confusion is obvious - the virus is checking for key code 53h, but at this level, the 53h refers to the keyboard scan code for Del, not to the ASCII code for S.

If the Del key is pressed, the virus checks whether the Ctrl and the Alt keyboard flags are set; i.e. if it is the combination Ctrl-Alt-Del. If so, the virus simulates a warm reboot, clearing the screen, clicking the speaker and invoking the Int 19h (System Reboot) interrupt. All screen manipulations are performed via direct access to the video controller's I/O ports. Thus, the virus remains memory-resident even after the user has 'rebooted' using Ctrl-Alt-Del.

The virus checks the current CMOS date, and intercepts Int 08h (hardware timer interrupt). If the date is 30 November, the virus will trigger approximately two hours after the computer has been booted, and display the following message, which is kept encrypted in the virus body and decrypted on a character-by-character basis:

```
        S A M P O
       "Project X"
Copyright (c)1991 by the
SAMPO X-Team. All rights
reserved.
 University Of The East
         Manila
```

Next we see Sampo's most peculiar feature. If memory-resident, it tries to infect a diskette when its boot sector is being read (e.g. after a DIR command). The virus first reads the floppy disk's boot sector, checking whether it is already infected. If so, the virus simply returns control to the caller. Thus it does not attempt to stealth infected floppies; however, it does stealth the infected MBS of a hard disk.

If the floppy disk's boot sector is not infected, Sampo immediately tries to write that sector back again. The obvious aim here is to check whether or not the diskette is write-protected: if that is the case, the write attempt will fail and an error will be returned to the virus.

The diskette will be infected if it is not write-protected. If it is write-protected, Sampo decrypts an image of the Telefonica.A-infected boot sector, which is stored encrypted in its own code, and copies it to the caller's buffer. That is to say, the caller's read request will return a boot sector infected with Telefonica.A, when in fact the floppy is clean.

Sampo's Final Revenge

When scanning a write-protected floppy on a PC infected with Sampo, a user might get a totally misleading alert, such as (for example): 'Telefonica has been found in the boot sector of drive A'. The following natural attempt to remove Telefonica would be disastrous. Another nasty side effect is that, if a user takes a copy of the boot sector of a write-protected diskette on an infected machine, or just runs something like DISKCOPY, the boot sector of the copy will actually be infected with Telefonica.A!

Such a situation further strengthens the argument for accurate virus detection in memory from anti-virus software. [*Or the importance of proper clean booting. Ed.*]

VIRUS ANALYSIS 3

Jumper: Jumping the Gun

Ian Whalley

Jumper is a Master Boot Sector (MBS) virus which has little to draw undue attention to itself, save being encountered in the wild. Despite the fact that it is neither innovative nor particularly well-written, it does have a few features which are interesting not for their originality, but their rarity. This month's analysis takes a look at this oddity.

Infection

From the point of view of the user, infection of the hard disk proceeds in the usual manner. Internally, the virus begins in a manner typical to this type of virus: first, it determines the size of base memory by pulling the value at memory location 0413h in the BIOS Data Area (segment 0). It reduces this value by 2, and writes it back, decreasing the top of memory by 2K, and copies itself into this 'hole'.

Before execution jumps to the high copy, the virus records the interrupt 1Ch vector in the vector table at the base of memory and points that vector to its own code at the top of memory. Interrupt 1Ch, which is the System Timer Tick interrupt, is called by Interrupt 08h, and is generated approximately 18.2 times per second.

Processing now jumps to the high copy, pushing the segment and address onto the stack and issuing a RETF (Return Far) instruction.

At this point, Jumper retrieves the location of the copy of the original boot sector and loads it into memory. The virus then determines if it was run from a floppy disk; if so, it infects the first hard drive. Finally, processing is passed to the original boot sector code.

Clocking on - The Interrupt 1Ch Handler

Once the virus has become resident in memory, it does not immediately begin infecting floppy disks, because it has not yet hooked the appropriate interrupt vector. To understand how Jumper does start to infect floppies, we must examine the Int 1Ch handler in detail.

This is the routine which modifies the Interrupt 21h vector, giving control to the virus code and allowing it to infect floppy disks. However (as stated above), certain conditions must be met before it can actually infect.

Every time the timer ticks, Jumper compares the second lowest byte of the timer value in the BIOS Data Area with the value at offset 01C6h in the image of the boot sector containing the virus at the top of memory. Until the timer value exceeds the value stored within the boot sector, the interrupt vector is not altered.

The fact that the byte concerned is at offset 1C6h is interesting: in the case of an infected hard disk, this byte is part of the partition table - to be precise, it is the lowest byte of the value stored as the starting sector of the first entry in the partition table. This value will almost inevitably contain the number of the first sector on the second track of the disk, but its numerical value will vary depending on the layout of the disk in question.

If a floppy disk is infected, the value at offset 01C6h is part of the message which is displayed if the diskette does not contain a bootable copy of DOS - it is the space (ASCII value 20h) between 'Replace' and 'and'.

As soon as the value from the timer exceeds that from the boot sector, the Int 21h vector is hooked, and the value in the memory image of the boot sector is changed to 0. This serves as a flag which will ensure that the vector is not altered twice.

At first sight, this seems like an incredibly convoluted way to hook an interrupt vector. However, when the virus code is first executed and becomes memory-resident, DOS is not yet operating, so Int 21h does not point to the DOS Int 21h handler. Thus, the virus delays hooking the interrupt until a later time.

Interrupt 21h Handler

Once this handler is in place, Jumper proceeds with its dirty work. It intercepts only two of the multitudinous Int 21h functions - 0Eh (Select Default Drive), and 0Ah (Buffered Keyboard Input). Its behaviour is dependent on which of these functions called it.

"this leads me to suspect that the author of the Jumper virus is deliberately targeting the FORMAT command"

If it is subfunction 0Eh, and the new default drive is A or B, that drive is infected immediately, and the original interrupt handler is executed.

If, on the other hand, it is subfunction 0Ah (the Buffered Keyboard Input service provided by DOS), a number of tests are performed. Firstly, Jumper examines offset 043Fh in the BIOS Data Area (BDA). The byte held at this location contains a bitmap of the status of the drive motors. If a bit is set, the motor in the drive corresponding to that bit is running. Jumper checks to see if the motor in drive A: is active. If so, it continues to the next test; if not, control passes to the original handler.

Next, Jumper re-examines the clock in the BDA. It will only infect if the lowest bit is set. This amounts to a 50% chance of infection, and is effectively random, as this bit is changing more than 18 times every second, giving Jumper the properties of a sparse infector.

Int 21h function 0Ah is used by the DOS format program in order to accept keyboard input. Taking into account the 50% chance discussed above, Jumper seems to infect diskettes as they are formatted fairly reliably. This leads me to suspect that the author of the Jumper virus is deliberately targeting the FORMAT command.

If so, it is obvious why he checks to see if the motor on drive A: is running - in order to make the user less suspicious of strange activity on the disk drive. This would certainly fit with targeting 'FORMAT'.

Trigger

The trigger routine of this virus is nothing remarkable - sometimes, whilst booting up, it locks the machine by repeatedly displaying the character '∈'. The conditions for this are based (once again) upon tests of the clock count in the BIOS Data Area, but the trigger routine is called more often than can be accounted for here.

This is probably because the virus also calls the trigger if the attempt to read the original boot sector with Int 13h fails. As such attempts are supposed to be made three times to allow for the hardware, this is perhaps not surprising.

Jumper

Aliases:	Viresc, French_Boot, 2KB.
Type:	Non-polymorphic, memory-resident virus with no stealth capability.
Infection:	Master Boot Sector of hard disks, boot sector of floppy diskettes.
Self-recognition on Disk :	The first word at offset 0 in boot sector, which is set to EB01h if the sector is infected.
Self-recognition in Memory:	None.
Hex Pattern on Disk (at offset 3Eh in boot sector):	`BB00 7C31 C08E D089 DC8E D889 C7A1 1304 2D02 0090 B106 A313`
Intercepts:	Int 1Ch and Int 21h.
Trigger:	Repeated display of '∈' on boot-up, effectively at random.
Removal:	Under clean system conditions, use the FDISK /MBR command.

FEATURE

When the Chips are Down

Most articles on computer viruses deal with the subject of prevention in great detail; yet, it must be said that there is a shortage of good information on what to do when the worst does occur. At the best of times, dealing with a virus outbreak is painful. Fortunately, forewarned is in this case definitely forearmed: by preparing for such an eventuality in advance, the damage and disruption can be minimised.

Get Your Backup

There is much preparation which can be made to minimise the impact of a virus outbreak. Perhaps the most important task, and one which is too often neglected, is to ensure that there is a regular, reliable and well-managed backup procedure in place.

The most valuable part of a computer is often the data stored on its disk: this can be irreplaceable, and may represent many man weeks or even years of work. Often, only when it is no longer available do we appreciate its value.

Instigating a sound backup procedure represents a way to protect this asset. The importance of backups cannot be overstressed: every user should be aware that they are a vital part of a successful policy, as they provide a minimum fall-back position in case disaster strikes.

Make certain that you can locate and identify backups, and that you have tested the software which is designed to restore them. There is nothing worse than deleting a file, then discovering that it cannot be restored from a previously created backup.

When discussing recovery from a virus attack, it is a worthwhile process to back up every single file on a computer, including the executables. The *Windows* environment can be difficult and time-consuming to set up and configure, especially with some packages arriving on dozens of disks. The twenty or thirty minutes taken to scan and back up an already-configured PC can save a great deal of effort further down the line.

Naturally, the choice of how often to back up, and which files to copy, will vary depending on the role of the computer within the organisation, the type of software installed on the machine, and the value of the information stored within the system.

The Boot Disk

A clean boot disk is also indispensable when attempting to scan and repair infected machines. Although a disk created on a clean machine using the DOS command FORMAT /S will

suffice, one can do better by building up a small toolkit of useful utilities. If you decide to take this route, it is worthwhile including some of the following:

- A copy of HIMEM.SYS, loaded at boot time through CONFIG.SYS on the diskette
- A copy of DEBUG
- A small text editor
- A copy of a disk editor
- Any device drivers which are needed to access the fixed disk (e.g. access control or disk compression software)
- A copy of the FDISK and SYS programs

Note that although DOS is backwardly compatible (that is, a DOS 6.22 boot disk will work on a machine which has DOS v3.31 on the fixed disk), the reverse is not true. More importantly, if you intend to use the SYS or FDISK commands, you need to use a boot disk created using the same version of DOS as is stored on the target machine.

Ideally, you should create a different boot disk for each version of DOS used within the company. Each diskette should be tested, to ensure that it functions correctly on the PCs on which it is designed to be used. Armed with a series of regular backups and a collection of suitable boot disks, one has most of the tools which are required to deal with a virus outbreak.

Detection

There are a number of ways in which a virus attack can be detected within an organisation: an infected diskette may have been discovered, a computer may have been scanned and found to be infected, or a virus may even have triggered on a machine, alerting the user to its presence. Regardless of how the virus is found, before any action is taken, the contingency plans which have been prepared in advance should be consulted. It is frequently the case that more disruption is caused by well-meaning clean-up attempts than damage by the virus itself.

The first thing a user should do on discovering a virus is to contact the member of staff or department responsible for virus countermeasures. If the virus is found on a machine, that computer should not be used until such time as further instructions are received. Additionally, details of the incident should be written down (i.e. the exact virus name reported by the scanner, or a description of the text displayed by the virus).

One useful preventative measure which every company could implement is to ensure that all diskettes which are sent out to clients or even (depending on site organisation) between departments are scanned prior to use. This has two principal benefits:

- It prevents sending an infected diskette to a customer (this can be a very quick way to lose business)
- Should any company machine become infected, it is likely that infected objects will be found on the diskettes. Such a procedure can provide an early warning of infection on company machines

Containment

The first priority when dealing with a virus infection is to prevent it spreading to other machines. The response required will vary depending on the nature of the virus discovered and on the anti-virus software installed.

If a stand-alone PC is found to be infected, the first part of the containment process is to track down any floppy disks which have been used in the infected machine. On PCs which are part of a network, the immediate question is whether the virus has infected the network or not. Pure boot sector viruses cannot spread to other PCs via the server, whereas all other types of virus should be assumed to be able to infect files on the server and thus spreading to other PCs via the file server.

If on-access file-scanning is enabled on the network, and the scanner detects the virus concerned, it is unlikely that it will be necessary to shut the network down - indeed, this may well be the way in which the virus is discovered for the first time. However, if the network is not protected, it may be a worthwhile precaution to shut the entire LAN down.

This is a difficult decision to make, and involves taking into account the trade-off between network downtime and the likelihood of the virus spreading. It is important to prevent the cure being more painful than the disease!

Regardless of which type of virus is discovered, it is a good idea to detach all infected PCs from the network physically, and to write-protect all those floppy disks which are not intended to be infected on purpose.

Cleaning Up

When the virus has been successfully contained, it is time to begin searching for all infected floppy disks. During this process, it is important that no diskettes or files are exchanged between those machines known to be clean, and those which are not, as this can lead to re-infection of checked machines. If that happens, of course, the whole process will need to be repeated.

To be completely thorough, every machine and associated floppy disk should be scanned for viruses. This is often best done at weekends or in the evening, when disk traffic is minimised. Omit this step at your peril: if you have missed a single infected object, it is easy for the virus to spread within the system once again. Depending on what was infected, and how much the computers concerned were used, the areas to search can be limited accordingly.

It is also important that staff check their home machines. Several companies offer a cheap (or even free) extension of a site licence to cover home users. This is highly recommended, as one of the

most common ways of introducing a virus into the office is from an infected machine at home. Once all infected media has been identified, it is time to begin the recovery process. Again, this will vary according to the type of virus encountered.

First Stages of Recovery

Before trying to remove a virus, ensure that a copy has been preserved for analysis on a clearly-marked, write-protected disk. To delete the virus from the hard disk, first cold boot from a known clean, write-protected system floppy. It is not sufficient to reboot using the Ctrl-Alt-Del key combination (see Sampo, p.10): more and more viruses can remain active in memory after a 'warm' boot.

Recovering from an infection by most boot sector viruses is perhaps the easiest (and most common) procedure: simply replace the infected boot sector (Master or DOS, depending on the virus type) with a clean one. This can usually be carried out using a disk-editing tool, but if your software does not provide boot sector virus disinfection facilities, the process can often be carried out by hand - the exact procedure will vary slightly for each type of virus.

The DOS boot sector can be replaced by the SYS command, while FDISK /MBR will replace the Master Boot Sector in DOS v3.31 and later. Any infected floppy disk should be reformatted under clean system conditions after valuable data has been backed up.

It is often more difficult to clean up after an attack by a file-infecting virus. In this case, the best advice is always to replace infected files from backups. However, it is frequently the case that there are no backups available, and one is left with the choice of using anti-virus software to 'disinfect' the file, or deleting affected files and replacing them from master disks.

In security terms, it is obviously better to re-install the software, as a user can then be certain that the files are fully functional. Unfortunately, as mentioned above, this can be an extremely time-consuming operation.

As a result of the frequent lack of backups and the substantial time it can take to reconfigure a PC, most anti-virus packages have developed a facility to 'disinfect' a file. Disinfection attempts to reverse changes made by the virus, returning the file to its original state. It has the advantage of being very quick and convenient, allowing a PC with many infected files to be cleaned in a short period of time. Unfortunately, in most cases, file disinfection cannot be 100% successful: there is a small but finite chance of file corruption, or of repaired files being subtly different from their uninfected originals, leading to problems later.

The performance of several products and the risks associated with virus disinfection are outlined more completely in *VB*, September 1994 p.11. Disinfection is a poor replacement for a backup; you should be aware of the potential pitfalls in adopting this approach.

In the rare case of infection by a virus which causes gradual corruption of data, it is necessary to find out when the machine(s) became infected. In addition, the consequences of using potentially corrupt data must be ascertained, as must those for restoring the data from the latest clean backup, or carrying out a sanity check on the data.

Backtracking and Recriminations

When clearing up a virus infection, it is often worthwhile to make careful notes on what was infected, and where. Although in many cases it is impossible to trace the source of the outbreak, it is usually possible to learn how the policy failed and what can be done to improve matters.

One of the problems is that unless only a small number of computers or disks are infected, it is difficult to determine when the infection happened. It is easy to jump to conclusions, and blame the wrong person for bringing in an infected object.

This can create a considerable amount of bad feeling, and can make staff very defensive when the subject is raised. Should users be afraid of disciplinary action if a virus is found on one of their personal disks, there is a possibility that the disks at highest risk will not be submitted for checking. When the outbreak in the company has been cleaned up, these disks can provide a route for the virus to start the entire process once again.

Contingency Planning

In every organisation, there should be a plan of action which will be followed in the event of a virus outbreak. When you discover that machines for which you are responsible are infected, it is important to carry out the clean-up in a controlled manner. The process needs to be documented in a clear, well-thought-out manner, so that mistakes are not made when under stress and also to prevent further damage being done.

Every installation is different and has different requirements. The precise way in which you will implement a recovery from a virus attack will vary from site to site, and from virus to virus. Unfortunately, there is no 'one size fits all' answer to the problem.

By drawing up plans for what to do in the event of an outbreak, the process can be considerably simplified. Following good computing practices (e.g. taking regular backups) and creating write-protected boot disks will make recovery easier, as will considering which steps you need to take to enable your company to return to normal operation as quickly and as painlessly as possible.

Finally, you should attempt to learn and profit from outbreaks which do occur. Treat them as an opportunity to increase staff awareness of the problem (without resorting to scare tactics), and also as a chance to re-examine your policy and procedures, in an attempt to ensure that, as far as possible, your defences remain unbreached in the future.

COMPARATIVE REVIEW

NetWare Protection

Jonathan Burchell

This month sees the second *VB* comparative review of anti-virus packages for *NetWare*. The test-sets used are identical to those in the last DOS scanner review (excepting the boot sector viruses), and are considerably more difficult than those which were used in the last *NetWare Comparative*. This caused some discomfort for certain products.

Central Point Antivirus

The *Windows*-based administration of this product is gorgeous, putting it at the top of 'feature-packed' user interfaces together with the packages from *Intel*, *Symantec* and *Cheyenne*. The administration is intuitive, and configuring even complex options such as scheduled scans is simplified to a 'point and click' operation. Many aspects of the user interface, including keyboard short-cuts and the toolbar appearance, can be completely customised.

CPAV for NetWare allows groups of servers to be placed into single logical administrative domains, considerably reducing the work involved in maintaining a multi-server site. This is further enhanced by extensive messaging options provided by the *Central Alert* NLM which includes the ability to generate messages via broadcasts, Email, pagers and SNMP (Simple Network Management Protocol).

This product might have a place in large distributed sites. I say 'might' because, *CPAV* unfortunately falls short in delivering the goods: poor virus detection lets down the hard work which has been put into the user interface. Whilst the score on the Standard test-set is noteworthy, In the Wild and Polymorphic test-set detection suggest that the product lacks muscle. The poor detection results in the real-time scan are also cause for concern.

Firebreak

Firebreak from *Norman Data Defense* is sold as a package containing an NLM, and a copy of *Norman Virus Control* for DOS, *OS/2* and *Windows* workstations. The workstation software can send incident messages to the server component for inclusion in the log file. Administration of the server component is via the server console - no workstation administration controls are provided.

Firebreak does not allow multiple servers on a network to be configured into domains from the point of view of configuration or sharing of virus patterns; it does, however, allow a server to be nominated as the central communication hub. In this case, copies of the NLM running elsewhere will send messages to the hub.

The administration interface is clean and simple, and the facilities offered place *Firebreak* in the middle ground, as do the messaging and logging features. One notable problem is that the list of types of files to be scanned is fixed: unfortunately, it is incomplete, for instance it does not include any *Windows* enhanced mode drivers (386 type files) in the scan list. This shortcoming needs to be addressed.

The performance of the detector on both 'Standard' and 'In the Wild' test-sets is extremely creditable; the polymorphic detection ratios show promise, but need to be improved. Of perhaps more concern than the complete zero scores (which presumably just means that *Norman* has no logic to detect the virus) is the fact that the scanner failed to get 100% on the polymorphics it could detect - this may indicate some weakness in the detection algorithm.

H+BEDV's AntiVir for NetWare

Although the only copy of *AntiVir for NetWare* from the German company *H+BEDV* we could obtain was a beta version (a full release is expected shortly), *VB* decided to take a sneak preview of what will be on offer.

AntiVir for NetWare is configured and managed from the server console, and although it does not allow for servers to be grouped into domains or for cross-server updating of set-up or signature databases, it does appear to offer a full set of scanning and configuration options.

Our initial evaluation of this product was limited to collecting the data for the review, partly because, being a beta copy, we did not have any documentation and the on-line help (which appears quite good) is still in German!

Detection ratios show promise: Standard test-set results were excellent, and nearly 100% was scored on the In the Wild viruses. The polymorphic rates place this product towards the end of the first division group, but I suspect could be improved to compete at the top. I look forward to giving this product a full work-out in a forthcoming issue of *VB*.

IBM Anti-Virus for NetWare

IBM Anti-Virus for NetWare consists of a single NLM which provides both background and real-time checking. Administration and configuration of the scanner is carried out from the server console.

	Cruncher (25)	Uruguay (75)	Satanbug (100)	Girafe (1024)	MtE (500)	One_Half (1024)	Pathogen (1024)	Smeg (1024)	Score (%)	Standard (230)	Score (%)	In the Wild (126)	Score (%)
Background scanning													
CPAV	0	0	51	0	475	0	0	0	8.2	215	93.5	95	75.4
Firebreak	0	3	100	1004	477	0	1	1	27.9	223	97.0	122	96.8
H+BEDV	0	0	100	1024	500	1024	1024	322	78.1	229	99.6	122	96.8
IBM	0	53	100	1020	500	1020	1024	1024	86.6	228	99.1	120	95.2
InocuLAN	2	75	100	1023	500	1023	247	840	69.0	228	99.1	120	95.2
Intel	0	0	100	0	27	0	0	0	5.1	220	95.7	103	81.7
NAV	0	0	100	1024	500	0	0	0	34.8	225	97.8	111	88.1
Net-PROT	0	0	100	3	462	0	1024	903	45.2	216	93.9	106	84.1
NetShield	0	0	0	20	67	0	0	0	1.4	224	97.4	105	83.3
S&S AVTK	14	75	100	1024	500	1024	1020	1019	90.3	230	100.0	126	100.0
SWEEP	0	75	100	1024	500	1024	1024	1024	96.5	230	100.0	126	100.0
Real-time scanning													
CPAV	0	0	51	0	466	0	0	249	12.0	212	92.2	87	69.0
Firebreak	0	3	100	1004	471	0	1	1	27.8	223	97.0	122	96.8
H+BEDV	0	0	100	1024	500	1024	1024	78	74.3	229	99.6	122	96.8
IBM	0	53	100	1020	500	1020	1024	1024	86.6	228	99.1	120	95.2
InocuLAN	2	33	41	764	251	350	338	458	35.0	195	84.8	103	81.7
Intel	0	0	100	0	27	0	0	0	5.1	220	95.7	103	81.7
NAV	0	0	100	1024	500	0	0	0	34.8	225	97.8	111	88.1
NetShield	0	0	0	20	67	0	0	0	1.4	224	97.4	105	83.3
Net-PROT	0	0	100	3	462	0	1024	903	45.2	216	93.9	106	84.1
S&S AVTK	0	0	0	0	0	0	0	0	0.0	225	97.8	108	85.7
SWEEP	0	75	100	1024	500	1024	1024	1024	96.5	230	100.0	126	100.0

Detection results: The inclusion of a significantly enhanced Polymorphic test-set has stretched the field in this review, with scores ranging from 0-96.5%. Of the total score for the polymorphic test-set, 75% is derived simply from the total number of polymorphic viruses detected. The remaining 25% is made up scoring 3.125 marks for each polymorphic virus which was detected with 100% reliability.

The interface is a little strange and takes some getting used to. Configuration can be carried out by specifying command line options at the time of invoking the NLM, or by typing them into the program via the console interface. This is not a menuing program: options are typed at a prompt in the same format as they would have been placed in the command line. Some contextual help is available in the event of errors.

The interface differs not only in its configuration, but in what happens to files in which a suspected virus is detected. Most products automatically process an infected file according to current configuration (delete file, rename it, move it to a quarantine directory etc): with this product, file access is restricted, and the file is place on an internal list. The administrator must process this list via the console interface and indicate the action to be taken for each file.

Although these actions are comparable with other packages, there are no global options, so it is not possible to give such commands as 'delete all files on the list'. I can understand the philosophy of making any viral detection event so important that it must be individually processed, but, combined with the somewhat eclectic user interface, this is a product you will either love or hate.

InocuLAN

InocuLAN has always struck me as rather a dark horse. It is unique in offering full administration and configuration from DOS or *Windows* workstations, and also a slightly feature-

reduced interface via the server console. The features available in the workstation interface put it in the 'big league'. Groups of servers can be built into domains, and administration of the whole is graphical and simple.

InocuLAN includes some of the best reporting and messaging facilities I have seen. Messaging can be via broadcast, Email, pagers, SNMP and even fax. Like the *Intel* product, the NLM can automatically download new signatures and propagate them to other servers in the same domain. Unlike the *Intel* product, downloading is initiated by the NLM itself, via a modem accessible to the server.

The exciting point about this product is its combination of a 'state-of-the-art' user interface with an extremely good scanner. The background scanner missed the Cruncher and Uruguay samples, and did not detect 100% of the Pathogen and Smeg test-sets, but nevertheless shows promise.

Unfortunately, the real-time scanner did not produce such good results; indeed, real-time results are only fair-to-good. A product which uses different technology for real and background scanning runs the risk of one or the other falling behind - this seems to have happened here. I encourage *Cheyenne* to attend to this, and to strive for the all-important 100% across all test-sets. If they can, then, combined with the user interface, the big prize will be open to them.

Intel LANDesk

The Vprotect part of *Intel's LANDesk* product range attempts to provide a complete network and server administration facility, including advanced features such as remote workstation monitoring and configuration via SNMP. VProtect has *Windows* and DOS administration facilities: the DOS utilities are based on a graphics package which provides a similar look and feel to *Windows*. The administration facilities and user interface are excellent; the whole concept is geared towards multi-server, multi-domain sites.

The package also has excellent scheduling, reporting, logging and messaging features and a great deal of support for workstations, including a standalone DOS scanner. Amongst the nice touches are the ability to automate the download of new virus signatures from the *Intel* BBS. This can be carried out periodically from any workstation equipped with a modem - new patterns automatically propagate around servers in the same domain.

'Nomadic user support' ensures that when a transient user logs in, the protection software on the machine is automatically updated, if necessary, and that incident logs created during the user's 'time away' are collected from the transient machine and added to the central database. *LANDesk* licensing allows for unlimited nomadic users. With the current polymorphic detection ratio of this product, it really cannot be considered as a serious contender, which is a great shame, given the otherwise excellent interface and facilities.

NAV for NetWare

Norton Anti-Virus for NetWare from *Symantec* is another product which features a truly stunning administration and configuration interface. Although this is only available from a workstation running *Windows*, it is a joy to use: *NAV* really scores in ease and flexibility of configuration.

Like *InocuLAN* and *Central Point*, it supports extensive messaging and alert options. Alerts may be generated via broadcasts, EMail and pagers. Good facilities are provided for managing groups of servers, and for customising the operation on a per server or per domain basis.

NAV has very good logging facilities, and an excellent report generation module. A dialogue box allows over eight event types (for example, start/end of scan, error messages, and detection messages) to be filtered and printed from the activity log. As well as specifying events, a date range for the report, and a list of specific workstations/users, can be configured. These features make it simple to filter the activity log to extract information about the events involved in a viral outbreak.

Given the excellence of the user interface and the reporting options, it is a great shame to have to report that the detector is not up to scratch. The Standard test-set results are good, but In the Wild results are only fair, and Polymorphic results are poor. Those polymorphics which are detected, however, are detected with 100% reliability.

Net-PROT

Command Software Systems' Net-PROT is a product which should provide a good solution for small, single-server networks. Administration is carried out via a DOS-based interface providing for mouse as well as keyboard input. Options provided are somewhat simplistic: for instance, only one event may be stored in the scheduled scan list, and alert, notification and logging facilities are limited. Documentation is a brief 16-page A5 manual, and on-line help consists of a single prompt line at the bottom of the screen.

Despite the simplicity and the reduced feature set, at the right price there should be a significant market for products like *Net-PROT* for protecting small, user-administered networks, providing they detect viruses well. Unfortunately, the default scanning engine shipped with *Net-PROT* missed the polymorphic viruses. A second scanning engine is shipped with the product, but is not installed by default. When, on manufacturer's advice, we installed this, detection ratios improved to those shown in the tables.

This product appears to be at a crossroads: if detection ratios can be improved, it may find a market in the niche described above; however, at the moment, although detection is credible, it is too low to be a first-line defence option.

	CPAV	Firebreak	H+BEDV	IBM	InocuLAN	Intel
NetWare versions supported	3.1x, 4.x & SFTII	4.0x	3.1x	3.1x, 4.0x	3.1x, 4.0x	3.1x, SFT III, 4.01
Specific 4.0 features	Compressed, Migrated	No	Soon	No	No	No
Name space support in box	DOS, MAC, OS/2	DOS, OS/2	DOS	DOS, OS/2	DOS & MAC	DOS, MAC, OS/2
Other name space support available	None	None	OS/2 Soon	None	None	None
Viruses detected	DOS & MAC	DOS	DOS	DOS	DOS & MAC	DOS & MAC
Realtime detection						
Executables	Yes	Yes	Yes	Yes	Yes	Yes
Any file	Yes	No	Yes	Yes	Yes	Yes
Specific inclusions	Yes	No	Yes	Yes	Yes	Yes
Specific exclusions	Yes	No	Yes	Yes	Yes	Yes
Processing delayed	Yes	No	Yes	Yes	Yes	No
Immediate scanning	Yes	Yes	Yes	Yes	Yes	Yes
Scheduled scanning						
Executables	Yes	None	Yes	Yes	Yes	Yes
Any file	Yes	None	Yes	Yes	Yes	Yes
Specific inclusions	Yes	None	Yes	Yes	Yes	Yes
Specific exclusions	Yes	Yes	Yes	Yes	Yes	Yes
Flexible schedules	Very	None	Yes	Yes	Yes	Yes
Multiple schedules	Yes	None	Yes	Yes	Yes	No
Administration						
Console configuration	No	Yes	Yes	Yes	Yes	No
Console monitor	Yes	Yes	Yes	Yes	Yes	Yes
DOS utility	Yes	No	No	No	Yes	Yes
MS Windows utility	Yes	No	No	No	Yes	Yes
Servers can be grouped	Yes	No	No	No	Yes	Yes
Cross-server updates	Yes	No	No	No	Yes	Yes
Messaging & alerts						
NetWare messages	Yes	Yes	Yes	Yes	Yes	Yes
Email	MHS	No	No	No	Yes	MHS
SNMP	Yes	No	No	No	Yes	No
Pager	Yes	No	No	No	Yes	No
Fax	No	No	No	No	Yes	No
Reporting & log files						
Display of log file	Yes	No	Yes	No	Yes	Yes
Filtering of log file	Yes	No	No	No	Yes	Yes
Server-based checksums	No	No	Yes	No	Yes	No
Server-based file repair	No	No	No	No	No	Via DOS
Workstation integration						
Login checks	Yes	No	No	No	Yes	Yes
Force logout	Yes	Yes	Yes	No	Yes	Yes
Centralised messaging	Yes	Yes	N/A	Yes	Yes	Yes
Built-in encyclopædia	Limited	Yes	Yes	Yes	No	Yes
Signature updates						
Frequency/Cost	Quarterly	4-6 times p/a	By arrangement	As required	As available	As available
Electronic access	BBS, CIS, FAXBACK, AOL	BBS	BBS, Compuserve	BBS	BBS,CIS	BBS, CIS
FTP Internet	No	No	Not direct	Yes	No	No
Automated download	No	No	Yes	No	Yes	Yes
Workstation software in box	DOS Windows	Yes	10+	No	Yes	Yes
Scanner	No	Yes	Yes	No	Yes	Yes
Checksummer	DOS Windows	No	Yes	No	Yes	No
Activity monitor	DOS Windows	No	No	No	Yes	DOS Windows
Encyclopædia	DOS Windows	No	Yes	No	Yes	Yes
Other	BootSafe	Windows	N/A	DOS/Windows message	N/A	Licence covers home users
Mac workstation in box	Yes	No	N/A	No	Yes	Yes

Norton Anti-Virus	Net-PROT	McAfee NetShield	S&S AVTK	Sophos' SWEEP	
3.11, 4.0x	3.1x & 4.0x	3.1x, SFT III, 4.01	3.1x, 3.12, 4.0x	3.1x, 4.x, SFTIII	NetWare versions supported
No	No	No	No	No	Specific 4.0 features
DOS & MAC	DOS	DOS	DOS & MAC	DOS	Name space support in box
None	None	None	None	None	Other name space support available
DOS & MAC	DOS	DOS	DOS & MAC	DOS	Viruses detected
			*See review	*See review	Realtime detection
Yes	Yes	Yes	Yes	Yes	Executables
Yes	Yes	Yes	Yes	Yes	Any file
Yes	Yes	Yes	Yes	Yes	Specific inclusions
Yes	Yes	Yes	Yes	Yes	Specific exclusions
No	No	Yes	No	No	Processing delayed
Yes	Yes	Yes	Yes	Yes	Immediate scanning
	Config shared with realtime	Manual only			Scheduled scanning
Yes	Yes	Yes	Yes	Yes	Executables
Yes	Yes	Yes	Yes	Yes	Any file
Yes	Yes	Yes	Yes	Yes	Specific inclusions
Yes	Yes	Yes	Yes	Yes	Specific exclusions
Yes	Limited	Some	Yes	Yes	Flexible schedules
Yes	No	No	Yes	Yes	Multiple schedules
					Administration
No	No	No	No	Yes	Console configuration
Yes	Yes	Yes	Yes	Yes	Console monitor
No	Yes	No	No	No	DOS utility
Yes	Yes	No	No	No	MS Windows utility
Yes	No	No	No	No	Servers can be grouped
Yes	No	Signature DB only	No	No	Cross-server updates
					Messaging & alerts
Yes	Yes	Yes	Yes	Yes	NetWare messages
MHS	No	No	No	No	Email
No	No	No	No	No	SNMP
Yes	No	No	No	No	Pager
No	No	No	No	No	Fax
					Reporting & log files
Yes	No	Yes	Yes	No	Display of log file
Yes	No	No	No	No	Filtering of log file
Yes	No	Yes	No	No	Server-based checksums
No	No	No	Yes	No	Server-based file repair
					Workstation integration
No	No	No	No	Yes	Login checks
Yes	No	No	No	Yes	Force logout
Yes	No	No	Yes	Yes	Centralised messaging
Yes	No	No	No	Yes	Built-in encyclopædia
					Signature updates
As available	Every 4-6 weeks	Monthly	Monthly	Monthly	Frequency/Cost
BBS, CIS	BBS, CIS	BBS, CIS ,AOL	BBS	BBS	Electronic access
No	MCI	Yes	No	No	FTP Internet
No	No	Yes	No	No	Automated Download
No	No	Yes	Yes	Yes	Workstation software in box
No	No	DOS Windows	Yes	Yes	Scanner
No	DOS	No	Yes	No	Checksummer
No	DOS	No	Yes	Yes	Activity monitor
No	DOS	No	Yes	Yes	Encyclopædia
None	DOS/Windows message display	None	Realtime needs workstation software	Realtime needs workstation software	Other
No	No	No	No	No	Mac workstation in box

McAfee NetShield

McAfee NetShield is unique in that it is available as shareware. As one might expect, the package includes a utility to automate the downloading of further updates and other products from the company's BBS.

Administration of *NetShield* is carried from the server console, via a standard *Novell* type interface. The range of facilities and options place the product in the middle ground, ideal for single server and geographically-limited sites. Messaging and logging options are very limited, as is scheduled scanning. Documentation is limited to a simple printed manual and rudimentary on-line help.

It seems strange that one of the benefits of registration is not a full documentation package, but perhaps the biggest surprise of all with this product is the extremely poor detection results. Polymorphics are almost completely missed and, whilst the Standard test-set results are excellent, In the Wild detection was also disappointingly low.

S&S' AVTK for NetWare

The user interface of this product can be described very simply: non-existent. The *AVTK* must be considered in two halves - the background/scheduled scanner and the real-time protection component. Background scanning is provided by the NFINDVIR.NLM, which, despite detecting the largest number of infected files, suffers from the complete absence of any administration or configuration interface. Scan configuration is carried out via command line switches supplied at the time of invoking the NLM. No workstation configuration or administration tools are supplied.

Scheduled scanning is little improved. Scheduled scans are configured via the server console after loading the NTOOLKIT NLM, which provides simplistic and unsophisticated options to configure scheduled scans. When the scan time arrives, the toolkit NLM simply launches NFINDVIR. Logging and reporting facilities are limited and there is no facility to group servers into domains.

This product is described as a toolkit; it should be pointed out that the NLMs can be configured via a 'programming language' which, once mastered, should allow powerful environments to be constructed. Real-time scanning is provided by a workstation TSR which is not linked to the NLMs at all, other than being able to send alerts to the server and being configured remotely from NTOOLKIT.

As can be seen from the test results, real-time protection is not good: the 'enhanced' TSR only claims to be able to detect polymorphic viruses when they become memory-resident, not during the copy process - a solution which provides only limited detection. Background detection, however, is simply excellent.

In the hands of a consultant or expert the *AVTK for NetWare* can be a powerful product, but mere mortals may have a difficult job constructing an environment which requires less than a rocket scientist to control, configure and interpret the results of. However, I understand that a new version has just been released which includes a GUI interface, and I look forward to reviewing it.

Sophos' SWEEP

Several improvements have been made to this product since we last looked at it, including the ability to decompress PKLITE, LZEXE and DIET-compressed files to check for viruses, and the ability to look for Macintosh viruses. The detection ratio is superb, although the background scanner came in a whisker behind *Dr Solomon's* in terms of the total number of infected files found - this was because a fault in the version we tested (which *Sophos* claims to have corrected in time for the version of the software currently being shipped) caused the scanner to miss DIET-compressed Cruncher infections.

SWEEP offers the best real-time detection of any product - a workstation TSR, *InterCheck*, submits files which require checking to the same detector as that used for background scanning. Files are submitted on the basis of whether or not the TSR finds that the file, having been through the checker, is in an 'Authorised Files' database. Another improvement is that the background scanner automatically builds a shared centralised database of authorised files on the server: this reduces the TSR's overhead dramatically.

Sophos has obviously spent its time building a great detector: now that the 'holy grail' appears to be within its reach, we can only encourage them to concentrate on further improving the user interface, reporting and logging facilities. With such solid foundations, the product should go far.

Conclusions

The enhanced polymorphic test-set has stretched the field in this comparative review, with scores on the test-set ranging from a dismal 0 out of a possible 100, to 96.5. Note that no single product was capable of detecting all replications of each of the eight polymorphic viruses used.

Full marks go to *Sophos*, for its outstanding virus detection scores in both background and on-access scans. The principle limitations of the product are the configuration and control issues: while *SWEEP's* configuration and control is easy to use, it cannot compete with the flexibility and power offered by the *Central Point's* of this world. However, the virus detection results easily make up for this.

The test scores for some of the products in the review range from mediocre to abysmal. Readers are well advised to look carefully at the tables, lest they remain oblivious to the weaknesses which some NLMs show.

PRODUCT REVIEW

Enforcing Security

Dr Keith Jackson

Disk authorisation is an extremely powerful way of ensuring that all incoming and/or outgoing diskettes are scanned, and products which provide this functionality have been available for quite some time. This month, *VB* looks at a new entry into this market: *Enforcer*, by *Precise Publishing*.

Enforcer provides disk authorisation, access control and anti-virus management for *IBM*-compatible PCs. It aims to prevent the use of unauthorised software, and to provide the means to enforce a company's own anti-virus policy. Its facilities can also be used across a network, but such features are beyond the scope of this review.

Through means of a device driver occupying 6.6 KB of RAM, *Enforcer* provides a boot-up password, floppy disk boot protection, and 'refusal of all floppy disks which have not been authorised (virus scanned)': i.e. if a user tries to use a floppy disk, *Enforcer* checks that the disk has been validated. This is done either by checking a 'signature' on the disk, or by checking a checksum of the entire disk. This checksum must be recalculated whenever files on that diskette are changed.

Installation

The installation procedure is straightforward: the name of the subdirectory used by *Enforcer*'s files is entered, as is confirmation that this particular installation of *Enforcer* will be a 'Gateway' - this is '*Enforcer*-speak' for a computer which checks that all floppy disks are authorised before executable files can be accessed. An onscreen bar graph shows how things are progressing during installation.

Enforcer requests confirmation that a clean boot sector be forced on to all diskettes as they are authorised (more bar graphs indicate further progression), and asks whether it should refuse to access diskettes containing executable files. A 'customisation' screen is then displayed, and a response required which accepts or rejects the setup values. The installation process might be cleaner if the questions requiring keyboard responses were asked together, and the user could let the installation program finish uninterrupted.

When installation is complete, the attributes of the subdirectory used by *Enforcer* are marked as Hidden, and all files contained within it are marked as Hidden and Read-Only. Still worse, *Enforcer* makes the DOS file CONFIG.SYS Hidden, System and Read-Only.

If any attempt is made to change these file attributes when the software is active, it changes them back again. Thus, this change to the attributes does go some way to preventing tampering with these critical files. However, naïve users, unable to locate CONFIG.SYS, may assume it has been deleted, and try to restore it themselves manually.

Exemptions

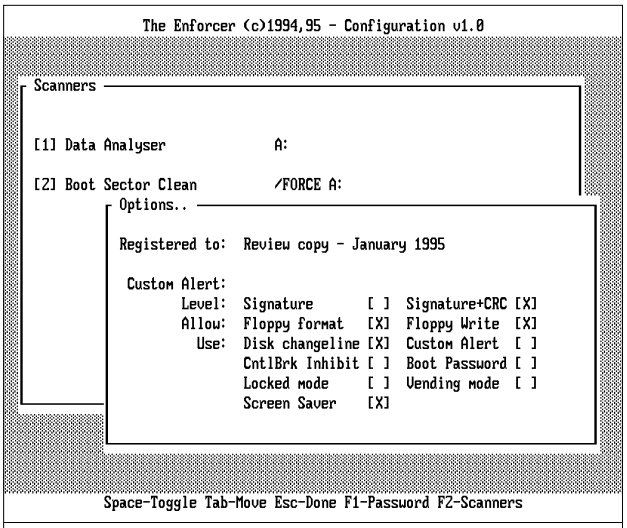
The package sent to *VB* was preregistered as an evaluation copy. The documentation describes how an 'authorisation code' is normally created when the software is registered by entering the company name or other identifier. After this, a chosen 'SUPER password' is entered, which is required whenever the configuration program is executed. As neither feature was functional on the software submitted, they could not be evaluated - nor could the disk-locking facility.

Inside the installation program is the following text message: 'The disk-locking facility is disabled in this evaluation copy'. *VB* always requests manufacturers to supply a 'normal' copy of their product for review; however, there is no practical method of enforcing [*Groan. Ed.*] this point.

Documentation

The manual provided is a 67-page A5 book which is quite easy to read, but somewhat short on detail. The error messages are documented in the manual: although this is to be applauded, further explanation of each such message would help users immensely. Additionally, a bit more consistency in the seemingly interchangeable use of the terms disc, disk and diskette would not go amiss.

There is no index in the manual. There should be. Apart from lack of time or idleness, I see no reason why documentation which pretends to be even halfway decent should be published without one. Creating an index is not overly difficult, with the



Enforcer provides either signature-based stamping of company diskettes or a complete CRC checksum of the diskette's contents.

facilities offered by word processors today. I have always pointed an accusing finger at products which cannot be bothered to include an index, and shall continue to do so.

Configuration

Enforcer provides an authorisation program which 'stamps' floppy disks with an authorisation code unique to each installation. Use of this authorisation program should be restricted to personnel and/or computers which are permitted to introduce executable code to a particular site. The program forces a diskette to be scanned (by up to six scanners) before it allows authorisation to proceed.

The software has preinstalled knowledge of most well-known scanner programs, and any scanner can be nominated. This is an immensely practical feature which I rate highly. As the number of known viruses continues to grow, scanners are having difficulties keeping detection rates up, with only a few of the best really succeeding (see *VB*'s recent comparative scanner review for evidence). It is eminently sensible to use the best scanners from within a product offering other anti-virus functionality.

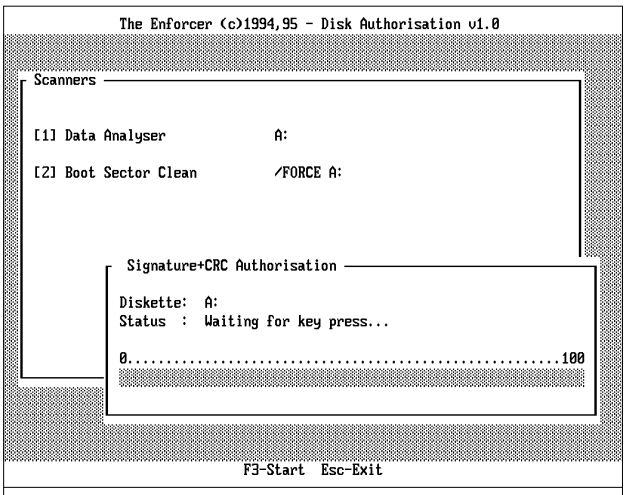
There are various setup options too numerous to discuss individually, but I feel that one in particular should be mentioned. Before a disk can be authorised, it is necessary to select whether the authorisation process only calculates a signature, or if a checksum should be calculated for the entire content of the diskette. The decision can be changed at a later date by reauthorising all floppy disks already in use: a time-consuming chore.

An option is also available which allows *Enforcer* to make use of the disk change line (if present) to restrict authorisation checking to occasions when the floppy disk has been changed, rather than every time it is accessed. The documentation mentions that verification of an authorisation checksum can 'take a few moments', and advocates use of the disk change line feature to ensure that 'subsequent accesses should not be delayed as long'.

Authorisation

The authorisation process itself is relatively painless. The diskette is taken to any PC which has *Enforcer* authorisation software installed. After entering a password, any number of floppy disks can be authorised. Each floppy is scanned by the prescribed scanners. If no viruses are found, authorisation information is written to the floppy: this requires that the hardware write-protection of the diskette is disabled, as write access is required. Great care must thus be taken to ensure that the computers used for authorisation are not infected by a virus. Given that several scanners will be available, this should not be too hard to arrange!

The time taken to authorise a 3.5-inch diskette was measured first when the disk was empty, and then when it contained 670 Kbytes of executable programs (spread across 20 files). For either 720 Kbyte or 1.44 Mbyte floppy disks, an empty floppy could be authorised in 3.5 seconds when just a signature was required. When a checksum was also required, authorisation time rose to 4.5 seconds for a 720 Kbyte diskette, and 9.6 seconds for a 1.44 Mbyte floppy. When these tests were carried out using diskettes containing the set of executable programs, authorisation times were on average about three seconds longer than those measured for empty floppy disks.



Although *Enforcer* could identify all of the trial changes to files on the diskette, enabling this option increased overheads.

None of these measured times is particularly onerous, and given that I was not using a fast computer to carry out the tests, I doubt that anybody would complain at the overhead induced by the authorisation process.

Overheads

When *Enforcer* is being used, its device driver monitors PC operation. Inevitably, this introduces some overhead. I tested this by copying first one short file, then a large set of files (1.25 Mbytes spread across 40 files), from the hard disk to a floppy (1.44 Mbyte). The measurements were repeated when the files were copied back from the test diskette to the hard disk.

The overhead on copying one file was barely measurable, an increase of about 0.2 seconds on a 6.5 second copying time. Even when the large set of files was copied, the increase in copying time was barely measurable (one second at most), despite the fact that copying took about two minutes. Curiously, the time taken to copy files when a checksum was used was less (by 1.4 seconds) than that taken to copy the files without *Enforcer* present. File copying proceeded normally no matter whether reading or writing was taking place. I failed to measure any significant overhead.

This seemed too good to be true, until it dawned on me that all these measurements had been made with the disk change line option activated. *Enforcer* had spotted that the floppy disk had not been removed, and was reducing the amount of work that it had to do accordingly. Thus, all the file copying tests were repeated with the disk change line option disabled. When files

were copied from floppy to hard disk, a small and barely discernible overhead was measured: from a few percent increase for the worst case of a single file being copied, to less than 1% when a large set of files was copied.

The picture was rather different when files were copied *to* a floppy disk. Then, the time taken to copy a single file was 6.1 seconds without *Enforcer*, 8.9 seconds when signature-only authorisation was in use, and 23.2 seconds using signature plus checksum authorisation. Copying a large set of files to diskette took 2 minutes 5 seconds without *Enforcer*, 2 minutes 8 seconds with signature-only authorisation, and a whopping 11 minutes 19 seconds with signature plus checksum authorisation. The large set of test files took considerable time to delete (literally minutes) when signature plus checksum authentication was active.

The first time a diskette is installed, a large overhead on floppy usage will be incurred, even if the disk change line option is in use. Only on second and subsequent accesses will this option have much effect.

Operation and Utilities

Whenever *Enforcer*'s device driver detects a problem with floppy authorisation, it pops up a message box, warning that it has barred use of anything on the diskette. A *Windows* program is also provided which can be used to make such messages appear onscreen when *MS Windows* is executing.

If a floppy authorised by *Enforcer* is left in a PC disk drive whilst a reboot is in progress, a warning message is produced and the PC continually issues audible beeps - a crystal-clear warning that something is amiss. Similarly, if attempts are made to delete files and/or format disks (assuming the relevant configuration options have been selected), the user sees a message that the 'Disk is write-protected' - even though, in fact, it is not.

I tested how *Enforcer* operates first by authorising a floppy disk with both a signature and a verifiable checksum. I copied several executables onto it, and used the DOS program DEBUG to make single bit changes at randomly chosen points within the executable(s). If the changes were made on an unauthorised PC (one not running *Enforcer*), the floppy could not be accessed when returned to the protected PC: a message box popped up stating 'Authorised disk - A: failed check'. Alterations to data files were also noticed, as evidenced by the fact that when *Norton Commander* was stored on diskette, it could not be executed if a single-bit alteration had been made to its associated INI file.

If any of the above changes were made on a PC protected by *Enforcer*, no error messages were displayed; in fact, nothing untoward happened. Obviously *Enforcer* updates the checksum dynamically, to cope with any alteration. When these were carried out on a floppy which had only been authorised with a signature, the alterations were never spotted by *Enforcer*. Authorisation in this mode seems merely to test that the diskette is one which *Enforcer* has authorised; file content is not verified, and other files can be added by other PCs without *Enforcer* provoking an error.

Several utilities are also provided: a small memory-resident program will prevent a user breaking out of a program with the Ctrl-Break or the Ctrl-C keystroke; another can overwrite the boot sector of any floppy with a 'known clean' boot sector, which automatically removes any boot sector virus from the diskette.

A further utility offers more control over file attributes, and yet another program can inspect files on a diskette and decide whether the floppy contains executable code. It does this by explicitly spotting files with COM and EXE extensions, and opening other files to check their content. Rightly, the manual states that there is 'no absolutely foolproof way of doing this'. However, it does claim to be capable of spotting 'several of the popular archive files'. This was a general claim, and I have no means of verifying it. For these utilities, I offer no comment other than that they seemed to work correctly and efficiently.

Conclusions

Enforcer does its job quite well. The manual states that the developers hope that users find it 'easy to use and effective'. With the exception of alteration of file attributes, and a caution applied about possible execution overhead, I would endorse both claims. *Enforcer* does involve extra work in using floppy disks, as it must be correctly set up on all PCs. However, there is a positive gain for all this effort: the software exerts considerable control over what can arrive into a PC via a floppy disk.

Considering the above measurements, I feel that the documentation is not really explaining the overhead introduced by its 'signature-plus-checksum' authentication method. The comment in the manual that *Enforcer* 'may extend by a few seconds' some operations is a gross understatement.

My main criticisms of this package concern the documentation and the way in which file attributes are used as part of *Enforcer*'s operation. Both points can be considered minor and, more importantly, eminently fixable.

Technical Details

Product: *Enforcer.*

Developer/Vendor: *Precise Publishing Ltd.*, PO Box 3731, Halesowen, West Midlands, UK, Tel. +44 (0)1384 560527, Fax +44 (0)1384 413689, Compuserve: 100043,2441.

Availability: Any *MS-DOS* PC v3 upwards, *Windows* compatible, requires 6.6K of RAM.

Version evaluated: v1.0.

Price: Single copies, £29.95 each; 100 user licence, £1500; 200 user licence, £2500; 500 user licence, £5000; 1000 user licence, £7500. Optional 30% maintenance in the second year.

Hardware used: A *Toshiba 3100SX* laptop PC (16MHz 386) with one 3.5-inch (1.4 Mbyte) floppy disk drive, 5 MB of RAM, and a 40 Megabyte hard disk, running under *MS-DOS* v5.00.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

Tel. 01235 555139, International Tel. +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email virusbtn@vax.ox.ac.uk
CompuServe 100070,1340@compuserve.com

US subscriptions only:

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

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

END NOTES AND NEWS

Symantec Corporation, with the latest release of *Norton Anti-Virus for NetWare*, has 'revamped' the package to make it more attractive on the marketplace, repackaging it and reducing the price. *NAV* for DOS and *Windows*, previously sold separately, is also included. **The product can scan DOS, *Windows*, and *Macintosh* files on a *NetWare* server.**

Product developer *Norman Data Defense Systems* has released a database with **information on some 6500 *Amiga, Atari, PC* and *Macintosh* viruses**, Trojan horses and joke programs. Topics covered include origin, virus removal, and publications on viruses. Details from the company's head office in Norway (Tel. +47 3281 3490), or its US office in Fairfax, Virginia (Tel. +1 703 573 8802).

The next rounds of **anti-virus workshops from *Sophos Plc*** will be held on 26/27 April, and 24/25 May, at the training suite in Abingdon. Day one is an introduction to computer viruses; day two, an advanced virus workshop. One session costs £325.00; both, £595.00. Contact Karen Richardson on Tel. +44 (0)1235 559933 for details.

Jeremy Gumbley, co-founder of the Italian firm *Symbolic*, is the new Technical Support and Development Manager for *Command Software UK*. At *Command's* US head office, the computer manufacturer *dan Technology plc* has begun shipping its systems bundled with *F-PROT*. Dyan Dyer, *Command* president, said: 'That **the company chose *F-PROT Professional*** to safeguard its systems is testimony that our product has become the new standard in virus protection.'

Last month's edition of *End Notes and News* contained an incorrect telephone number: **to contact *RG Software*** in Scottsdale, Arizona, call +1 602 423 8000, not +1 612 423 8000, as printed in March's *VB*.

There was an error in the ***IBM-PC Virus Update* table** in March's *VB*: under the entry for the Milan family, Milan.AntiNazi's last four bytes were given as 0B0 BA00. They should have read 0B01 BA00.

The fifth annual *Virus Bulletin* conference, ***VB 95*, will be held at the *Park Plaza Hotel* in Boston, Massachusetts**, from 20-22 September 1995. Internationally-renowned virus and security experts will address the problems of virus protection in the 1990s. For more information, contact Petra Duffield, Conference Manager, on +44 (0)1235 555139.

***LAN/SEC 95 (Europe)* will be held in London, UK**, from 23-25 May, with optional workshops on 22 May. The conference is on network security, and is sponsored by *MIS Training Institute* in association with *Euromoney Publications*. *MIS* has also organised other conferences on related topics - *Cruising the Internet Securely: A Security and Audit Practitioner's Guide* (London 12-14 June 1995, Brussels 19-21 June 1995, Frankfurt 26-28 June 1995), and *Firewalls and Internet Security* (London 15-16 June 1995). Contact Mandy Moore, Tel. +44 (0)171 779 8795, Fax +44 (0)171 779 8944, for details.

Precise Publishing will be holding a **one-day live virus workshop** at the *Business and Technology Centre* (Oldbury, West Midlands, UK) on 26 April. The session will cost £395 + VAT. For more information, contact Kevin Powis on Tel. +44 (0)1384 560527.

The American *Computer Emergency Response Team* (*CERT*) recorded 2,241 computer break-ins last year, an increase of over 75% on 1993, said spokesman Terry McGillen. **As the *Internet* grows, intrusive behaviour is increasing** proportionately, and the risk of hackers planting malicious viruses is getting larger. James Settle, a spokesman for the *FBI*, said government agents have always lagged behind hackers, and that the gap may be widening.

***S&S International's* next computer virus workshop** will be held at *Ashridge Management College* (Berkhamsted, Hertfordshire, UK) on 15/16 May. Cost for the two-day course is £680. Contact *S&S* on Tel. +44 (0)1296 318700, Fax +44 (0)1296 318777 for further information.