July 1993 ISSN 0956-9979

VIRUS BULLETIN

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

CONTENTS

EDITORIAL

Certifiably Mad!

On the 16th of June, a meeting was held at the *DTI* in London concerning the Government *ITSEC* evaluation scheme, and how it could be applied to anti-virus software. This raised some interesting issues, and also provided an insight into the murky world of inter-vendor politics.

The structure of the day was coffee, meeting, more coffee, more meeting, repeat *ad nauseum* (lit.), and the day proved beyond doubt that to be a success in the cut-throat world of civil servicedom, a high tolerance to caffeine and a strong bladder are required. The meeting was well attended by the industry gurus and the debate promised to be lively.

The problems of using the existing official security evaluation criteria when examining anti-virus products are many. The old scheme was originally designed to be applied to static products, like access control packages, which could be tested once only and then safely purchased by government.

Anything (pretty much) can be certified. For example, we would not want the chaps responsible for screwing the lid on 'our' atomic bombs to have a second-rate word processor (or too much caffeine, for that matter!). On the whole the scheme is a very good idea and provides a level of confidence in certified products, ranging from 'it does what the manual says' to complex mathematical analysis of the product's functionality, design and implementation.

However, once this system is applied to virus detection, life becomes much more complicated: rather than a static product, one must now attempt to certify a 'moving target'.

The *ITSEC* scheme is an attempt to set up a Europe-wide certification process, which will enable governments to insist that departments may only buy certified products. The idea is that by extending the design of the original brief, it will be possible to extend certification to anti-virus software in a meaningful way.

The first question has to be whether such a scheme is desirable? From a government point of view the answer is a definite yes - it provides a sensible method of approving a product for purchase. The industry would probably say yes - if it (a) provided another legend to adorn the product's packaging. (b) gained *my* company some sort of advantage over *your* company.

Regardless of whether the industry wants such a scheme, it is going to have one. The UK government has decreed that product evaluation is necessary, and so by this time next month/year/decade an *ITSEC* scheme specifically designed for anti-virus software will be in place. The industry can either try and make that scheme as meaningful as possible, or it can complain when an inappropriate scheme is imposed upon it. *ITSEC* is not imposed on anybody directly - but if evaluation is to be taken seriously, it must have some marketing kudos: if company A is certified, company B must also seek this status, or suffer a marketing disadvantage, *regardless of how meaningless the certificate.*

The problems with such a scheme are numerous. Traditionally, the certified products have been things like secure operating systems, where there is a clear function of the product: to secure data from unauthorised access. The aim of anti-virus software (particularly virus-specific software) is far more nebulous: to stop *known* viruses. This is going to lead to tremendous problems with certification. If the industry cannot agree on the *names* of viruses, what chance is there of creating a meaningful virus test-set?

The difficulty is that in order for the government to evaluate anti-virus software successfully it needs the cooperation of the industry in order to have a library of viruses to test against. What should be in this library? Should it be available to those whose products are being certified? How can the test-set be managed? How often should it change? These are just a few of the problems concerning the test-set - there are thousands more on more general matters.

For those adept at the type of verbal karate which such events require, the struggle is an interesting one. Things do not stop here though - how can we have a certification of a scanner that is anything more than ephemeral? With code being updated on a monthly basis, the only way is to also 'evaluate' the company: is the company sufficiently 'well connected' within the industry that it can reasonably maintain its scanner? Quite how this could be done is not clear, but the current thinking is that it would be rather like BS5750 certification, that is, a demonstration of a measure of the quality control within the organisation.

It is possibly too cynical to say that every word uttered in the meeting was prompted by self-interest, but the important thing about the industry is that its intention is to make money. Will the vendors be able to cooperate to provide a meaningful evaluation scheme and forget their financial differences for the 'greater good'? Fat chance.

However, the scheme may just get enough support from the industry if it provides enough marketing leverage or is not seen to be detrimental to a company's image. What the *ITSEC* evaluation is trying to achieve could be a worthy cause, and it would be a shame to see it hijacked for purely political or financial reasons. How can this be avoided? Well, the *meisterplan* as *Virus Bulletin* sees it...

NEWS

IBM to Publish *MS-DOS 6* Bug Report

IBM is to announce its new version of DOS at *PC-Expo* in New York at the end of June. At a special press briefing in Austin, Texas, at the beginning of the month, company executives announced that its developers had identified a number of bugs in *Microsoft's* DOS 6 code which they had fixed and consequently *IBM's* new version would be known as PC-DOS 6.1.

The company will bundle *IBM Antivirus/DOS* and *Central Point's Backup* utility as well as an unannounced disk compressor with its new release of DOS. With *Stac Electronics* recently announcing an *OS/2* version of *Stacker*, even money is on that company providing *IBM* with its DOS product - but *IBM* is keeping very tight-lipped until 'legal formalities have been completed'.

Details of *Microsoft's* DOS bugs will, said the *IBM* executives, be made public knowledge. This appears to be yet another salvo in the increasingly bitter operating system war being waged on the *IBM* PC platform ❑

VB '93 - The Biggest and Best Yet!

The organisers of the third annual *VB* Conference are reporting unprecedented interest in the event so far and have high hopes of yet another well attended conference. *VB '93*, which will be held in Amsterdam on the 9th and 10th of September, is not only the world's leading computer virus conference, but also the only opportunity in Europe for users to meet the key players in the industry. [*Find out why the current Editor has recurring nightmares about jugglers, cigarettes and Nigel Kennedy... Ed.*]

This year's conference will have a distinctly international flavour with expert speakers from industry and academia representing 7 countries and delegates from at least 23. *Expotel International Groups* has been appointed to co-ordinate delegate accommodation and travel from the UK and are offering very good rates on both - a total package, including registration, return flight from London and 2 nights hotel accommodation, can cost as little as £860.00.

For those who are unable to attend *VB '93* in Amsterdam, the proceedings will be available from mid-September for £50.00 plus postage.

For information on any aspect of the conference, please contact Petra Duffield on tel. +44 (0)235 531889 or fax. +44 (0)235 559935 ❑

Virus Prevalence Table - May 1993

Viruses reported to *VB* during May 1993.

Virus	Incidents	(%) Reports
Form	24	40.7%
New Zealand 2	8	13.6%
Spanish Telecom	5	8.5%
1575	3	5.1%
Cascade	3	5.1%
Joshi	3	5.1%
Helloween	2	3.4%
Tequila	2	3.4%
Advent.3551	1	1.7%
Azusa	1	1.7%
Dark Avenger	1	1.7%
Dir-II	1	1.7%
Italian	1	1.7%
Jerusalem	1	1.7%
NoInt	1	1.7%
Twelve Tricks	1	1.7%
V-Sign	1	1.7%
Total	59	100.0%

Hack-Tic Summer Conference

Remember the *Galactic Hacker Party* which was held in 1989? Ever wondered if that sort of thing would ever happen again? Well, wonder no longer, because *Hack-Tic* is holding a three day conference in Holland.

According to a full-page advert placed in the summer edition of *2600* magazine, the conference will contain lectures and workshops (!) on hacking, phreaking, lockpicking, and viruses, and an 'intertent ethernet'.

The conference is to be held at the Larserbos campground, on the 4th, 5th and 6th of August this year, and, quoting directly from the advertisement, the conference is for 'hackers, phone phreaks, programmers, computer travellers, electronic wizards, network freaks, techno-anarchists, communications junkies... and law enforcement officers (appropriate undercover dress required)'. It is not yet known whether the FBI are will attend the conference, after the last fiasco, where they attempted to disrupt a *2600* meeting in Washington DC ❑

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 24th June 1993. Each entry consists of the virus' name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or preferably a dedicated scanner which contains a user-updatable pattern library.

Type Codes

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

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

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

Known Viruses

Albanian - CER: This 1991 byte virus is assumed to be of Albanian origin, but it is not related to the Albania family of viruses.

```
Albanian        AC34 EFF6 D0AA BE04 00C4 3CB0 CFAA 0E1F A018 000A C075 1C80
```

Aragorn - CEN: A 1522 byte virus of Italian origin. It activates on October 28th and displays a message.

```
Aragorn         A3DD 0133 C08A 265F 0188 2616 0233 C080 3E13 0201 742C B43D
```

Atas II - CR: Two new, encrypted variants of this virus, 3215 and 3233 bytes long.

```
Atas II.3215    8B3E 0201 B0?? B97C 0CBE 1300 01FE 3004 46E2 FB
Atas II.3233    8B3E 0201 B0?? B98E 0CBE 1300 01FE 3004 46E2 FB
```

Backfont.1172 - ER: This 1172 byte variant is detected with the Backfont (905) pattern.

Beer.3192 - CR: Closely related to the Beer.3164 virus reported earlier.

```
Beer 3192       FA90 80FC 3B75 03E9 18FF 3D00 3D74 0F3D 023D 740A 80FC 5674
```

Black Jec.378 - CN: A 378 byte variant. Detected with the Black Jec (Bljec) pattern.

CFSK - CR: This 918 byte virus has not been fully analysed, but it contains the message 'Sorry. I need some MHz today! (CFSK)'.

```
Cfsk            80FC CF75 04B8 CF0C CF80 FC4B 7503 EB06 902E FF2E 3804 5053
```

Chang - CER: A 1759 byte virus. Awaiting analysis.

```
Chang           FB9C 3D00 4B75 03E8 0600 9D2E FF2E 5001 5053 5152 5756 1E06
```

CHCC - CEN: A 2662 byte virus. Awaiting analysis.

```
CHCC            B045 8845 02E8 D204 E8BC 01C3 BFEF 01B0 2A88 05B0 0088 4501
```

Chr - CER: This 869 byte virus has the primary effect of displaying a '#' character when a program is run while it is active. It also contains code which only is executed if the date is the 24th of any month or later, but this code has not been fully analysed.

```
Chr             3DFD FF75 03B0 77CF 80FC 4B74 03E9 BC01 5053 5152 1E06 5756
```

Coib - CR: A 702 byte virus, which does not seem to do anything interesting.

```
Coib            80FC 3E75 0981 FBC7 0775 0393 9DCF 3D00 4B75 03E8 5300 9DEA
```

Cossiga.883.B - EN: Closely related to the other 883 byte variant, and detected by the same pattern.

Dark Avenger.1693 - CER: This variant seems to be derived from the standard 1800 byte virus, and is detected with the Dark Avenger pattern.

Error 412 - CN: This 465 byte virus may display the message 'Runtime error 412 at 0697:4870'.

```
Error 412       83C2 2690 B903 00B4 40CD 21B8 0242 33C9 33D2 CD21 8BD6 81EA
```

Fat Table - EN: This 6542 byte overwriting virus contains the text strings 'hitohana - shikaru bashoka ni ku shiet.' and 'FAT TABLE ERROR'. It is compressed with LZEXE, and for that reason no signature string is provided.

Filehider.1067 - CR: This is a 1067 byte variant of the Filehider.789 virus, which was originally reported just as '789'.

```
Filehider.1067  8BF3 B9FF 7F26 813C FF2E 7406 46E2 F6EB 0E90 268B 7402 268B
```

Fish 6.B - CER: A very minor variant of the Fish 6 virus, with two instructions swapped in the decryption routine - a small change, which nevertheless causes many scanners to miss the virus. The virus is 3584 bytes, and fully stealth, like the original variant. An examination of the virus, using the methods currently used to group viruses into families indicates that the Fish 6 and Frodo families should be merged into one, as the viruses share significant amounts of code.

```
Fish 6.B        E800 005B B958 0D81 EBA9 0D2E 8037 ??83 C301 E2F7
```

Fisher.1100 - CR: The detection of this 1100 byte virus may be slightly complicated by the fact that it can be found anywhere in a file - not just at the beginning or at the end.

```
Fisher.1100     1274 1780 FC4E 7415 80FC 4F74 1080 FC3D 742E 3D03 CC74 09E9
```

Freak - CN: This virus is unusually easy to spot, as it regularly displays messages announcing its existence. It is 938 bytes long and probably of Turkish origin. The name is derived from the text 'YOUR SYSTEM HAVE A HARMFULL VIRUS : FREAK !!!'

```
Freak           33C9 49BA E6FF 8BD8 B802 42CD 21BA 4102 B902 00B4 3FCD 21B4
```

Grunt - CN: A group of three viruses, 346, 427 and 473 bytes long, which contain typical destructive code which overwrites disk sectors. The viruses are all encrypted.

```
Grunt.346       E819 00EB 1DE8 1400 3E8B 9657 028D 9E30 01B9 7400 3117 83C3
Grunt.427       E829 00EB 2DE8 2400 408D 9E40 0148 3E8B 96A8 0240 B993 0048
Grunt.473       E81D 00EB 21E8 1800 B9D1 0040 8D9E 3401 903E 8B96 D602 F7D0
```

Halley - P: A 7856 byte 'companion' virus, written in a high-level language.

```
Halley          EC5D C305 2A2E 6578 6504 2E63 6F6D 0055 89E5 B800 019A 7C02
```

Hamster - CN: One of the few viruses that are actually found 'in the wild'. It is 546 bytes long, contains the text 'Turbo Hamster Virus!', and does nothing but replicate.

```
Hamster         03FD F3A6 83F9 0074 C180 3D00 74BC BE3F 0303 F5AD 2D03 002E
```

Infector - CN: Six new variants of the Infector virus are now known, in addition to Infector.822 (originally reported as '_822'). One of them, Infector.782 is detected with the Infector.822 pattern.

```
Infector.444    24FE 0C1E A2E6 028A 36E9 028A 16E8 028A 2EE7 028A 0EE6 028B
Infector.624    A200 01A0 DC02 2EA2 0101 A0DD 022E A202 01B9 9000 BB00 002E
Infector.726    A200 01A0 D402 2EA2 0101 A0D5 022E A202 01B9 0001 BB00 002E
Infector.933    A200 01A0 9503 2EA2 0101 A096 032E A202 01B9 0001 BB00 002E
Infector.984    A200 01A0 E303 2EA2 0101 A0E4 032E A202 01EB 0190 B500 B11C
```

Invisible Man - CER: Two variants of this Italian virus are known, 2926 and 3223 bytes long. Both are polymorphic, and cannot be detected with a simple search pattern.

James - CR: The name of this 356 byte virus is derived from a text message it contains: 'James Bond is alive!' This virus infects when files are opened or executed, but does not seem to have any significant side-effects.

```
James           9C50 5351 521E 0657 5680 FC4B 740C 80FC 3D74 078B D780 FC6C
```

Little Girl.1004 - CER: Very similar to the 1008 byte variant reported earlier, and detected with the same pattern.

Log - CR: A 320 byte virus. Awaiting analysis.

```
Log             3D00 4B74 052E FF2E C403 5253 1E06 89D7 B82E 4338 0574 0347
```

Murphy.Delyrium.1780 - CER: Closely related to the Delyrium virus reported back in 1991, but of a different size. Detected with the HIV pattern.

Omt - CN: The name of this virus is derived from a text message it contains: 'one more thing'. The virus contains code that will attempt to trash drive C:, but only if the year is 1993 or higher - which probably indicates that the virus was released in 1992.

```
Omt             EB01 C38B 3601 018D BC17 01B9 8901 8035 2A47 E2FA
```

Oxana - ER: Three new variants are now known.

```
Oxana.1436      B435 B090 CD21 8CC8 8ED8 2B06 6802 A368 028C C03D 0000 755D
Oxana.1572      B890 35CD 218C C88E D82B 069E 03A3 9E03 8CC0 3D00 0075 59B8
Oxana.1671      B890 35CD 218C C88E D82B 06F7 03A3 F703 8CC0 3D00 0075 64B8
```

NaziPhobia - CEN: Three primitive, overwriting viruses, which seem to be written in Pascal.

```
NaziPhobia A    0005 2A2E 636F 6D05 2A2E 6578 6555 89E5 83EC 04C6 46FF 00BF
NaziPhobia B    0005 2A2E 636F 6D55 89E5 83EC 04C6 46FF 00BF 1901 0E57 B820
NaziPhobia C    9A96 007C 28BF 7019 1E57 9A0C 098F 2889 EC5D C204 0005 2A2E
```

Perfume.653 - CR: This is probably an old virus, as it claims to be version 1.2, whereas version 1.3 has been known for a long time.

```
Perfume.653     FCBF 0000 F3A4 81EC 0004 06BF 9800 57CB 0E1F 8E06 3C00 8B36
```

Pick - CR: This 843 byte virus does not infect COMMAND.COM the same way as other files, but uses a method similar to that used by the Lehigh virus.

```
Pick            148B 6EFE 83C5 0336 C744 1400 01B9 1F03 8DB6 2C01 8BFE AD35
```

Sleepwalker - CR: Awaiting analysis. This virus is 1266 bytes long and contains the string 'Sleepwalker. (c) OPTUS 1993'

```
Sleepwalker     9C3D 00FF 7511 B801 FF9D FA2E 8E16 2101 2E8B 2623 01FB CF53
```

Stsv.B - CN: 200 bytes in length, just like the original STSV virus, and detected with the STSV (200) pattern.

Talking Heads - CN: An overwriting 519 byte virus. When it activates it displays a line ('This ain't no party...'), taken from a song by the Talking Heads group.

```
Talking Heads   B43E CD21 B404 B001 B500 B101 B600 B200 CD13 7237 B45B B920
```

Tchantches - CER: This 3303 byte virus activates on the 1st of April, displaying a message. It is known 'in the wild' in Belgium and France.

```
Tchantches      5249 2E8B 16DE 0D81 FB9A 0375 03BA C5AF 2E31 1783 C302 3BD9
```

Techno - CN: A 1123 byte virus. Awaiting analysis.

```
Techno          D3E8 0E5B 01D8 8ED8 891E 2600 A32A 00FF 2E28 00BE 0300 BF00
```

Terminator II - CER: This is a 2294 byte virus, which is not related to the other 'Terminator' viruses. It is encrypted, slightly polymorphic, and uses stealth techniques to avoid detection. This virus has been reported 'in the wild' in the Netherlands.

```
Terminator II   5E56 B95D 048B FE06 1E0E 0E07 1FFC BB?? ??AD 33C3 AB?? FA1F
```

Timid.557 - CEN: A badly written, 557 byte variant, detected with the Timid.306 pattern.

Uruk-Hai.427 - CR: A new, 427 byte variant. Not significantly different from the others that are known.

```
Uruk 427        5052 5351 1E3D 004B 7503 E85F 001F 595B 5A58 EBE7 B003 CF49
```

Vampirus - CER: This 1499 byte virus is encrypted, but contains the text 'ROMANIAN VAMPIRUS'. The search pattern below should be used with care because of the high number of wildcards.

```
Vampirus        BD?? ??81 7600 ???? 4581 FD?? ??72 F4BD ???? C3
```

VCL - CN: Several new variants have been made available to researchers recently: (423 - CN, 476 - CN, Mindless - overwriting). In addition a new 408 byte overwriting variant is detected by the generic VCL string published for the VCL.394 virus.

```
VCL.423         B41A 8D56 80CD 21E8 1600 5AB4 1ACD 218B E533 C08B D88B C88B
VCL.476         B41A 8D56 80CD 21B9 EB09 B805 FEEB FC80 C43B EBF4 8D9D 4401
VCL.Mindless    E900 00B9 EB09 B805 FEEB FC80 C43B EBF4 1E2B C050 B42A CD21
```

Vienna.1239 - CN: This 1239 byte variant may overwrite COM files with 'reboot' code, just like the original 648 byte version.

```
Vienna.1239     ACB9 0080 F2AE B904 00AC AE75 EDE2 FA5E 0789 BC16 008B FE81
```

Wilbur.C - CN: This variant is 512 bytes as the other two that are known, but the text message is different - it says that Wilbur is not Russian, but American, and not related to Akuku. The reason for this is probably that SCAN mis-identifies the earlier Wilbur variants as Russian-A (Akuku), but as the author correctly claims, the viruses are not related.

```
Wilbur-C        F7DE E8CF FE83 FE00 7414 32E4 8A86 0502 8BCE F6F1 32C0 86E0
```

WWP - CR: This is a 382 byte virus which infects files when they are opened, renamed or executed.

```
WWP             3DD0 D075 03B0 2BCF 3D00 4B74 1480 FC43 740F 80FC 5674 0A80
```

XAM - CER: A 797 byte virus which has not been analysed. This virus contains the text messages 'Wait viruses XAM' and 'Hi, Dimitriy Nikolaevich!'

```
XAM             80FC B075 04B8 FCDE CF50 5351 5256 5706 1E55 8AC4 3C4B 7465
```

Ziuck.1372 - CER: This virus is detected with an old Darth Vader pattern, but that should be ignored. The virus is 1372 bytes long, but variants with other sizes have been reported as well.

```
Ziuck.1372      5886 E05F 073D 4B00 7503 E903 013C 3D75 03E9 FC00 3C4F 7503
```

INSIGHT

Mark Hamilton

Ray Glath - Fighting Back

Arizona is probably one of the most spectacular states in North America. It is an area of stark contrasts, from the hot and dusty desert bowl to the south and west to the pine forested mountains to the north and east. It is from here that Ray Glath, one of the anti-virus 'old guard', wages war on virus writers and their supporters.

Glath's home is in Scottsdale, a small city adjoining the north-eastern edge of Phoenix, which is a beautiful oasis in the harsh desert. The day I arranged to visit Ray, it was already 89 degrees at 8am, so I gratefully accepted his suggestion that we sit outside on the patio to his house, next to the pool. As we talked, his wife Bev brought us freshly-made lemonade squeezed from the fruit of their own trees, and the radio gently played country music in the background.

Many people in this industry were just starting their formal education when Glath joined the computer industry, others were not even born: 'I began my career in data processing as a programmer back in 1964 working on mainframes and a variety of technologies. I started *RG Software* in 1984 with our first product called *PC Tracker*.'

Ahead of Time

In 1987, Glath developed a utility for disaster prevention. 'It was called *Disk Watcher* and was a TSR which worked very closely with the operating system. When the Lehigh virus appeared in the fall of 1987, I realised that since I had a TSR which worked with the operating system, it was not a giant leap forward to add in virus detection methods at a low level within DOS.' The philosophy was that *Disk Watcher* would prevent an infection taking place and was, arguably, the first anti-virus monitoring program.

Glath released the virus-detection version of *Disk Watcher* in the following spring. 'At that time, the whole situation with viruses was viewed with scepticism by most people. It was about that time that Peter Norton had been quoted in the popular press as saying that viruses were an urban myth like the alligators in the sewers of New York City - they plain did not exist. This was at the time when we were, of course, dealing with real live viruses!'

Disk Watcher, he believes, was ahead of its time: 'People didn't want to think about preventing viruses . If you don't think the problem is real, why protect against it?'

Glath could see the demand for a more reactive approach to fighting viruses which he believes reflects life in general: 'the predominant view was that if I get sick, I take a tablet rather than indulge in any preventative medication.' So during the summer of 1988, he started work on the development of *Vi-Spy*, a program to detect and get rid of viruses, and this was eventually released in the autumn of 1989.

User Contact

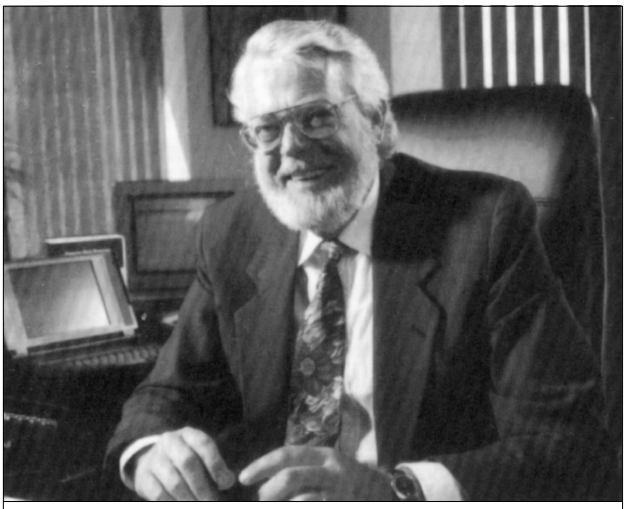
Vi-Spy is now in its tenth major release and has gone from strength to strength both technically and in sales. It is not, however, a product that many will have heard of, particularly outside the United States of America. So, I wondered, who does buy *Vi-Spy*? 'We predominantly sell to the corporate marketplace; corporates and the Government are the areas where we have the greatest expertise. We have been servicing this marketplace since 1984 with *PC Tracker* and we have always had a reputation for very stable software requiring a low level of technical support. Our products do what they say they will do.'

He no longer routinely hears about commonplace virus infections although he does keep in very close contact with his clients and they occasionally tell him of infections of Stoned, Form and Michelangelo that his software has successfully detected and eradicated. But what of the rarer viruses, does he hear about those? 'The rate is starting to increase a bit. We are probably now seeing two viruses per month that we have not heard about travelling or making the rounds in the United States, although we do have customers who have *Vi-Spy* installed throughout their worldwide operations.'

The Little Black Book

It was Glath who drew *Virus Bulletin's* attention to Mark Ludwig (of *Little Black Book* fame); was Ray alarmed by people such as him? 'Very much so. The Freedom of the Press banner can sometimes get over-used and people do not take into consideration the far-reaching aspects of their actions. Too many people treat viruses as a joke - but it wouldn't be a very funny situation if a virus were to shut down a medical diagnosis computer. Any time that a computer is used in a life and death type of environment, any time that kind of machine can get infected by any outside source, it's no longer a game and it's no longer a matter of dullards striking back at the "establishment". It's a case of jeopardising human life and the people who are writing viruses and the people who are encouraging the writing of viruses, such as Ludwig, have no regard for that aspect.'

Glath feels that many virus writers have no understanding of the consequence of their actions. 'I think, in many cases, the young folks who are writing viruses don't have enough



Glath: 'As far as I am concerned, his [Ludwig's] work should not be published and he should be doing some jail time.'

foresight to think about where they [the viruses] could go or what they could do in the way of damage. Ludwig, in particular, claims that his books are for educating people in the defence against viruses. However, in *2600 Magazine*, Ludwig had a paid advertisement soliciting people to learn how to write viruses by buying his book. So obviously, he tailors his marketing efforts to the audience he seeks.'

'This type of behaviour is totally irresponsible and it's a damn shame that we have no laws to convict him. As far as I am concerned, his work should not be published and he should be doing some jail time. He has no regard for general welfare of the public with that kind of attitude.'

Collaboration and Support

Glath collaborates, at an unofficial level, with a number of other researchers around the world, but because of the high degree of confidentiality in such dealings, he was unwilling to name any names. 'We deal with small, dedicated companies who, like us, are on the front-line of fighting the virus situation', he said, coyly.

Glath has a low regard for those companies selling anti-virus products who themselves have little or no virus expertise - companies who simply buy-in, badge and sell-on third-party technology. 'It comes to the situation of support. An anti-virus product is like no other product in the computing field. There are a lot of very unique aspects to anti-virus software: firstly, a potential customer has no way of testing it unless he has a collection of viruses - most people do not want to have a collection of viruses as they can be very dangerous in the hands of the inexperienced. So you can't test it properly, and consequently it's very hard to know if the product is doing the job that's being claimed by the vendor.

This is why the small specialist firms fare better, claims Glath. 'When a vendor has developed the product themselves, they know from whence they speak. I can tell you unequivocally how *Vi-Spy* will react in a given situation. The companies who buy or license products for re-marketing purposes don't fully understand what it is they have got. Since they are selling to the masses, they have a standard support model which usually means a couple of clerical-type personnel trained in giving pat answers to the top ten list of problems that can occur - they are not able to deal with a suspected virus. And also under this kind of a support model, when you are dealing with the masses, you have got a high volume of support calls from people with varying degrees of experience, not knowing where to turn when they have a problem.'

'We are not confronted with that situation. We have expert support staff that can get in and answer almost any question that comes up; if we don't have the answer, we will get the answer. Quite often we find we are answering questions about problems in other people's products - other anti-virus products. One of the most frequent tech calls we get has to do with the *Central Point* - and now the *Microsoft* - anti-virus product where *Vi-Spy* reports a virus in memory, when they run *Vi-Spy* having run the *Central Point* (or *Microsoft*) *Anti-Virus*. The reason is that *Central Point* doesn't clean up after itself - it leaves the signatures in memory.'

Vi-Spy is written entirely in assembler and compiled using SLR's Optasm. Glath maintains that since his staff are experts in assembly language, its use doesn't slow down the development of new versions. It has its advantages too, he believes, as assembler programs tend to be smaller, faster and have a higher level of control than those written using higher level third-generation languages such as C or Pascal.

Closing Thoughts

Glath thinks 'it's a damn shame that the people who are developing new virus techniques are not devoting their talents elsewhere.' He was referring to such techniques as tunnelling - where the virus strips back interrupts to the BIOS entry point. 'I think it's going to be an ongoing battle that's never-ending. With the virus authors looking for new ways to subvert the system and the defenders providing counter-measures - it is a giant chess game that's going on.'

Glath is a colourful figure in the anti-virus world. He will often be found swimming in his pool, or driving his red Corvette around the Arizona countryside. Glath is a giant of a man, very much the gentle knight in the anti-virus wargame, who has a strong sense of right and wrong. Will the virus writers eventually win? Nobody knows, but one thing's for sure, he has been around longer than most and has the stamina to stay the course.

FEATURE

James Beckett

When is Not a Program a Program?

'Executable Code' is the cry from many anti-virus vendors, as the final word in virus control. 'A virus has to modify executable code or it cannot spread further' is the familiar maxim of the vendors - but which files on your machine *are* executable, and exactly what do we mean by 'executable code' anyway?

The concept of 'executable code' used to be very simple - viruses append and overwrite COM and EXE files. These are the files containing the machine code instructions which are loaded and run when you type a command line. Data - the information which is manipulated using programs to do useful work - are ignored by viruses. One can't run data, as it doesn't mean anything to the machine without the applications to use it.

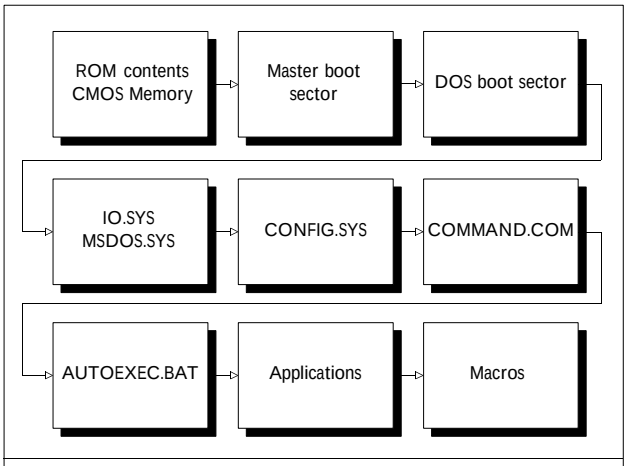
The Thin Red Line

In recent years however, the dividing line between data and executable code has become ever more blurred; data is in a way directing the action of the PC, and what is 'code' anyway but data which the CPU understands as instructions? Outside the environment of an IBM-compatible PC, COMMAND.COM becomes just non-executed data.

Program files are the most intuitive location to expect to find executable code, and thus are virus targets, but others have long been noted: some of the earliest viruses have used the boot sectors of disks for their code, so that when a machine is booted with an infected disk left in the A: drive (even a non-system disk which otherwise contains only raw data) the virus will become active.

Several other areas of the system have been marked as potential targets for infection, though the result would severely limit the spread of the virus. It is possible that people have written such varieties but, they have not been successful enough to come to light.

IO.SYS and MSDOS.SYS *could* be infected, but the virus could only live on system disks; OVL overlay files *could* be infected if they followed a known format but nobody uses a standard format so the virus would have to restrict itself to a small range of programs; BAT batch files *could* be infected but the PC batch file language is so retrograde it is barely powerful enough to write a reasonable virus - and any would be trivial to spot. This will be discussed later.



The boot process of the IBM PC. As application software becomes more powerful, it is becoming possible to write viruses in the Macro languages used by spreadsheets.

The Executable Path

To pin it down, a virus must get itself into the executable path of the system. This does not mean the DOS PATH variable, but the entire sequence of operations executed from powerup to powerdown. This path has been diagrammed to death but so far only the simple DOS *run* process has been considered in depth. Even this has gaps.

The basic boot process for DOS is shown above. The last of these steps, the macro languages provided in applications, is rapidly being developed to the extent that they are really fully-fledged programming languages with the power to do many tasks - even this could become a virus threat.

Obscure Objects of Desire

Almost universally, scanners take the names of files to be indicative of their contents. They retrieve directory listings and access those files their designers *expect* to find viruses in - COM, EXE, SYS, OVL etc. Usually a further discrimination is made on the contents of the file - you cannot assume that a file with a COM extension follows the COM format - if it starts with 4D5Ah or 5A4Dh it is processed as an EXE file. But this is rarely taken any further.

For example, although DOS refuses to acknowledge a program unless called *.COM or *.EXE, CONFIG.SYS can load a device driver of *any* extension, not just *.SYS or *.EXE - a correct treatment of the system would be to parse the config file and check the files mentioned there.

This is another domain - surely CONFIG.SYS is 'data'? It certainly isn't 'executable code' in the traditional sense and one starts to see that, just as 'code' only constitutes instructions within the confines of the 80n86 CPU, certain data files

become instructions in the Gestaltof the operating system. So 'executable' really means 'represents instructions to the system', and for the purposes of this article we are talking about those instructions which have sufficient power of expression to support areplicating program.

CONFIG.SYS thus highlights two problem areas which can be considered separately. The first is that there are a number of ways in which true 80n86machine code can be given control, and that such code can be kept innon-obvious places; the second is that, in the wider environment of the operating system, a*pointer* to executable code may have to be considered in a sense executable.

A Pointer to Ponder

So where does this leave us? These 'pointer' files do not contain true viruscode, but is the information a virus could insert sufficient to raise thealarm? Is there other, legitimate, information which may disrupt the testswe devise? In the case of CONFIG.SYS, the task at first seems simple enough: look for DEVICE= statements and scan the corresponding file. Of course, theanti-virus producers need to be on top of any new developments in DOS, forexample, DEVICEHIGH= and INSTALL= are now pointers in CONFIG.SYS. With DOS 6the option switches (eg DEVICE?=) must be checked too.

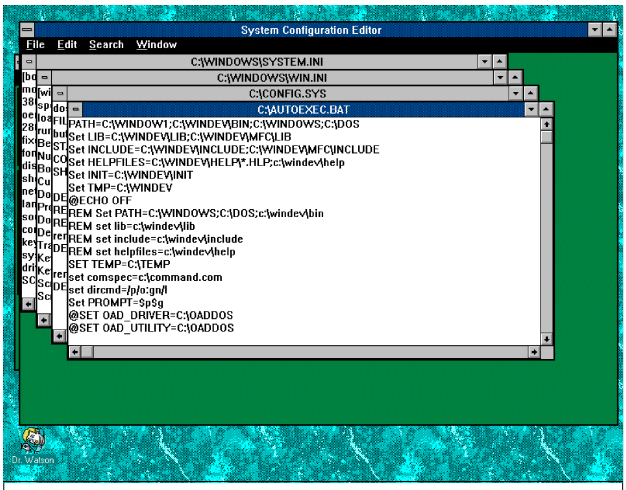
Worse, there are somethird-party drivers designed to do this sort of thing, which load otherdrivers mentioned on the command line. This sort of knowledge is the hardestto incorporate into a product. Taking it to extremes, one could provide a user-updatable rule-based system for finding executables, but this isgetting perilously close to programming, which most users shy away from.

So perhaps this approach is simply flawed? Is static analysis 'neither necessary nor sufficient'? Should one instead be thinking along the lines of a resident scanner which can trap a loading file regardless of how it isaccessed?

There seem to be enough problems just looking at CONFIG.SYS - the firstconfiguration file in the DOS load process! More complexenvironments can be expected to have more and subtler problems; before considering*Windows 3.1* let us look briefly at batch files.

The New Batch

The field of virus writing plods along fairly steadily with predominantly dullcopycat viruses, and new ideas are usually one-offs. The Mutation Engine,for example, was claimed to be the end of virus scanning, but notall that many MtE-based viruses have been seen and anyway, most scanners can now reliably detect the Mutation Engine.



Think that you understand *Windows*? Think again. This is the System Configuration Editor which comes with *Windows*. A simple utility, but did you know it was there?

A few months ago a batch filevirus caused a slight stir, and employed a method considered in this officesome time ago (Good to know we are ahead of them sometimes, anyway!). The virus, called BATMAN, bootstrapped itself out of the limitations of the DOS batchlanguage by picking on a particular piece of text at the start of the BATfile which also happened to be a valid machine code instruction. When executed as a batch file, it would copy itself to a COM file and run that. The initialinstruction jumped to a point beyond the copy-and-run instructions whichcontained real 8086 code, and thus the virus was installed in memory.

This hybrid file is an excellent example of a self-contained pointer system: the batch file is not usually considered a virus threat, but the system isfunctional enough to initiate something that is. Still, the end result isthe running of machine code.

Windows of Opportunity

Windows has proved to be a very popular operating environment - the majorityof users are uncomfortable with the unfriendly command line prompt andquickly learn that typing 'HELP' usually doesn't. After a few minutes using any windowing system, one can quickly pick up the basics of point-and-click, and every window has 'Don't Panic' inscribed in large friendly letters.

What is simpler on the surface, though, is supported by levels of complexitythat would overwhelm an average user. *Windows* programming initiallysurprised me too, despite having programmed for otherwindowing systems. At a middle level, one requires a file system which supports the runtime system, but users are insulated from all this by the interface. The vast majority of users haven't the faintest clue

what most of the files in their *Windows* distribution are for, or even what they *do*. *NT* and *OS/2* will surely bring more of the same.

The two sides of the problem mentioned above again guide us around *Windows*: For one, the *true executable* or machine code is not confined to EXE files - *Windows* makes much use of the 'segmented executable' file format for such things as dynamic link libraries. A file no longer consists of a program which merely starts at the beginning and continues to the end; files may have several *entry points*, defined by the *Windows* interface, which are invoked by the kernel or other programs as necessary. Different file extensions are used to denote the way in which a program file is expected to be used, but all still have the EXE file format and can contain a virus in a variety of new ways:

*.DLL - Dynamic link libraries

*.SCR - Screen savers

*.MOD - Module files

*.CPL - Control Panel libraries

Dynamic link libraries are a feature of *Windows* that allows code to be called from several different programs, while only having one copy of the DLL shared between them. This reduces memory and disk usage and enables behaviour modification of a number of programs. In fact, *Windows* relies completely on these - most of the functionality of *Windows* is contained in KRNL386.EXE, USER.EXE and GDI.EXE, and most of the standard utilities use features from SHELL.DLL and COMMDLG.DLL.

Although clicking on, say SSTARS.SCR in File Manager will not run it (producing instead a *Windows* error message), renaming SSTARS.SCR to SSTARS.EXE *will* let it run, immediately, bringing up the configuration screen. Moreover, it can be run directly - the WIN.INI file has an entry specifying which extensions denote files which have the EXE format and may be run - by modifying this you can click on any such file and *Windows* will run it. Checking just *.EXE files is not sufficient.

Again, the WIN.INI file could be parsed, the list of runnable extensions found, and those files checked. An alternative would be to examine *every* file on the system for an EXE header and scan them all.

The friendly interface in *Windows* is largely handled by the Program Manager. One can access every file on the system by using File Manager and run programs from it, but there is a lot more on the disk than just programs. Users do not like to hunt through the file system to find what they need. Thus all the most used programs are given an icon in a window, and split up into functional groups.

Putting a program into a Program Manager group doesn't create a new program, it just makes a link between the program file, an icon on screen, and some extra information for housekeeping. These are collectively known as *Properties*, specifying the name which appears in the group, and the path of the executable file which runs when the icon is double-clicked. By selecting a program item and hitting Alt-Enter, one can see and change this information, but most users are not aware of this.

This provides a disturbing possibility; it opens a way for a virus to infect without even touching an EXE file: modify the run path of all the Program Manager icons to point to a non-EXE file and tell *Windows* it is allowed to run it. Even existing checksummers in their default mode would fail to pick it up. If such a virus were to pick random extensions, we would have to check every file on the system in one way or another.

The information is stored in the GRP files in the *Windows* directory. These files cannot be checksummed in a simple way because they contain much more information which changes regularly. In a GRP file, whenever a Program Manager subwindow is moved or has its size changed, or a group is iconised, or an item moved, the corresponding file will change. Once again, the file could be analysed but does one need or want to go to all this trouble?

In the Real World

In the world of 'real' operating systems, similar problems exist. In the Unix varieties for example, executable files have no extension to signify their status, but there is a flag in their *inode* entry which does. One still has to look at all files on disk though. The script and utility languages are replete with the abilities to copy and replicate and run in the background, and must certainly be considered executable. There are pointers upon pointers to lead cause to effect. Thankfully, in a well protected operating system, a virus would be highly impotent and exposed, and privileged features could trap and isolate any that were developed.

Is an infected but renamed EXE file still a virus? Is an infected Unix program without the execute permission a virus? A compressed file? If one goes back far enough talking about potential 'virusness', one would have to flag compilers as being potential sources of viruses... of course they are, but one can't ban them on those grounds! The point is that one has to think carefully about the objects which need protection - just because *you* do not consider a file to be executable does not count! This applies far more to generic virus detection than virus specific detection, as for once the scanner manufacturers have it easy. The problems posed for checksummers are rather large, and should be addressed now, before the next wave of problem viruses occurs.

VIRUS ANALYSIS 1

Eugene Kaspersky

Peter-II - Three Questions of The Sphinx

How much does a user really need to know to operate his IBM PC effectively? Certainly he needs to understand how to use the keyboard, and with the advent of so many graphical interfaces, knowledge of the mouse is also necessary. Basic knowledge about how the computer works also helps - how to use the disks and, most importantly, how to turn the machine off and on. Reading the manuals is also a pretty good idea, so that the applications which are used are understood. Is this enough? Maybe.

What a user needs to know in order to defend his computer from computer viruses is a more difficult question. Should he simply know how to use a virus scanner? Does he need to understand how lots of different viruses work, and exactly how and what objects they infect? It is a difficult question, but I think my answer to it will come as some surprise.

The latest news from the antiviral battle-front is that if the user wants to defend the contents of his computer from viral attack, he should know have an outstanding knowledge of trivia. For example, to be fully prepared, the user should know the names of rock-superstars and their popularity. Unbelievably, this knowledge can be invaluable in the fight against viruses... especially if the computer in question happens to be infected by the Peter-II virus and the date is February 27th.

The Installation Routine

Peter-II is an ordinary memory-resident master boot sector virus. It is six sectors long (0C00h bytes), made up of five sectors of virus code, and one sector which is used as a data area to store the original boot sector which is replaced by the virus.

The virus is executed when the user boots the machine from an infected disk. Its first action is to decrease the effective size of the system memory (by decreasing the word at the address 0000:0413 by four), read the rest of the virus in from the disk, and copy itself to the memory at address 9F00:0000. This has the effect of reducing the total system memory by 4K.

As the virus is executed before DOS has loaded, the DOS services Get System Time and Get System Date are not available, and the virus has to use other more complex means to ascertain the system date. It does this by directly reading the data stored in the CMOS: the virus outputs the address value into port 70h and reads the result which is returned in port 71h. If the current date is set to February 27th, the virus calls the trigger routine (see below).

If the trigger conditions are not met, the virus reads in the Int 13h vector and stores in within the virus code. It then initialises its own Int 13h handler, and sets the relevant entry in the interrupt vector table to point to it. If the computer has been booted from an infected diskette, the virus now attempts to infect the hard drive. When this process is finished, the original boot sector is executed.

However the virus does not check the contents of memory before installing itself, which can cause the machine to crash in some instances. Consider the case of attempting to boot a machine with an infected hard drive from an infected floppy diskette. The virus is first run from the infected floppy disk, and, after hooking Int 13h, it executes the floppy disk boot sector. If this disk is not bootable the familiar 'Non-System disk or disk error. Replace and press any key when ready' message will be displayed.

At this point, if the disk is removed and a key pressed, the copy of the virus on the hard drive is executed. This hooks the current Int 13h address (which already points to the virus). Therefore the next time an Int 13h interrupt is encountered, the machine will crash.

Infection and Int 13h Handling

The master boot record is infected during virus loading. The virus reads the original sector and checks the virus ID byte - if the byte at offset 01FDh is equal to BBh, the virus assumes that the disk is already infected, and the routine aborts. If not, the virus saves this sector on the hard drive at sector 6, head 0, cylinder 0 and writes itself into the first physical sector of the hard drive, and to the next four sectors.

The virus monitors all calls using Int 13h to provide an infection mechanism and stealth. Whenever there is a request to read or write to the sectors, the virus substitutes appropriate register values so that it appears that the disk is not infected. If the request concerns the original sector one, the contents of the relocated boot sector are returned/altered. If the request concerns any of the sectors two to seven, the call is passed on to sector eight. This works on most machines, as these sectors are usually filled with zeros.

Unfortunately life is not always so simple. On early*NetWare* servers (versions 2.xx) this space is used for the start of the *NetWare* boot code, and in this case extensive damage will result. Many boot sector viruses use this 'dead space' as storage, and for this reason, viruses of this type on*NetWare* servers almost invariably cause a disaster.

Whenever a floppy disk is used, the virus checks to see whether it is already infected by examining the contents of the disk's boot sector. If the value of the byte at offset 01DFh in the boot sector is 11h, the infection routine aborts.

The virus then checks another section of the boot sector - the byte at the address 0018h. This contains the number of sectors per track on the floppy disk. If the value is not equal to 15 (i.e. if the disk is not a 1.2Mb 5.25 inch disk), the infection routine terminates.

If the disk is deemed suitable for infection the virus attempts to format an extra cylinder at the end of the disk. A normal 1.2Mb disk has 80 cylinders which are accessed by DOS, numbered 0 to 79. Although it is not possible to access tracks outside this range using standard DOS calls, some drive controllers are capable of using these extra cylinders, and the virus takes advantage of this in order to infect the disk without decreasing its storage capacity.

The virus uses these extra sectors to store part of the virus code. The relocated boot sector is stored separately in the last sector of the standard root directory (sector 14, head 1, cylinder 0)

Trigger

On February 27th, as described above, the virus calls the trigger routine. This routine is encrypted, and the first step is for the code to be decrypted. Once completed, the routine displays the message:

```
Good morning,EVERYbody,I am PETER II Do not
turn off the power, or you will lost all of
the data in Hardisk!!!

WAIT for 1 MINUTES,please...
```

Then the virus encrypts all the sectors of the physical hard drive: all the words are XORed with the value 7878h. If the machine is switched off at this point, all data on the drive will be lost, and the user will have to restore from a backup. However, it is possible to recover the disk by correctly answering the three questions which the virus displays next:

```
Ok.If you give the right answer to the
following questions,I will save your HD:

A. Who has sung the song called "I`ll be
there" ? 1.Mariah Carey  2.The Escape Club
3.The Jackson five  4.All  (1-4):

B. What is Phil Collins ? 1.A singer  2.A
drummer  3.A producer  4.Above all   (1-4):

C. Who has the MOST TOP 10 singles in 1980's ?
1.Michael Jackson  2.Phil Collins (featuring
Genesis) 3.Madonna  4.Whitney Houston   (1-4):
```

The user should give three correct answers, in this case the virus decrypts and restores the hard drive sectors and types:

```
CONGRATULATIONS !!! YOU successfully pass the
quiz! AND NOW RECOVERING YOUR HARDISK ......
```

and the disk is recovered. If any of answers are wrong, the virus displays:

```
Sorry!Go to Hell.Clousy man!
```

and all the data on the drive is lost.

While this trigger routine doubtless caused the virus author great mirth, the casual disregard for other people's data makes this a rather nasty piece of malicious code. Fortunately, there is no time limit on the questions, so the user can ring up his friends to find the answers! And what are the answers to these 'three questions of the Sphinx'? Easy... Four, four and two.

PETER-II

Aliases:	None known.
Type:	Memory-resident Master Boot Sector. Fully Stealth.
Type:	Floppy Boot Sector and Hard Drive Master Boot Sector.
Self Recognition:	
Disk	Checks the byte at the location 01FDh for the value BBh or 11h.
Memory	None.
Hex Pattern:	`fa0e 1f33 c08e c08e d0bc 007c 2683 2e13 0404 fb0e 07b9 0300`
Intercepts:	Int 13h for infection and stealth.
Trigger:	Displays three questions and encrypts the contents of the hard drive sector by sector. If the questions are answered correctly the disk is recovered.
Removal:	Specific and generic removal is possible. Under clean system conditions, replace original contents of Master Boot Sector from sector 6 (hard drive) or from logical sector 28 (floppy).

VIRUS ANALYSIS 2

Jim Bates

BFD-451

Yet another so-called 'whiz-kid' seems to have made his bid for stardom in this latest virus reported at large which attempts to capitalise on some of the security loopholes within DOS. Once again however, the code is riddled with errors but the virus as a whole may still survive and spread.

The virus is obviously intended to be multipartite, infecting both the DOS boot record of fixed disks and the boot sector of floppy diskettes, as well as certain types of EXE files, and under fairly restrictive circumstances it will function as such. The limitations will become obvious as the details are explained and even though there is no deliberate attempt to introduce damage, there will be system malfunctions with possible corruption to both data and program information on infected systems.

Installation

Since this virus can arrive in a system from either an infected disk or an infected file, there are two distinct installation procedures to consider.

Firstly, let us consider the case of booting from an infected diskette. As the virus code is located in the boot sector of the floppy disk, it is loaded into memory at boot time. When executed, it immediately locates the top of available memory and moves itself into a position two kilobytes below this. A total of four kilobytes is subtracted from the memory indicator (the virus uses some of this as workspace) and processing then transfers to the high memory copy of the code.

The original floppy boot record is collected from track 0, head 1, sector 3. On 360k floppy disks this is the position of the last sector of the root directory, other formats of floppy disks are infected without relocating the original boot sector, although the boot installation procedure attempts to load an alternative record from track 0, head 0 sector 12. This will prevent certain formats of floppy disk from booting properly.

A routine then hooks the virus into the system disk services and checks the fixed disk for infection. If the fixed disk is not infected, the virus attempts to infect it and finally passes control to the original boot sector. Once a fixed disk has become infected, the installation process is similar but the original DOS Boot Sector is collected from Track 0, Side 0, Sector 12. On most machines this sector is unused but there are several proprietary boot and access control systems which do use it. In this case it is possible that serious system damage will result.

Infection from a file follows a similar pattern although in this case, once the code is in high memory an 'Are you there?' call is issued to determine whether the virus has already been installed. This consists of placing a value of F0h into the AH register and issuing an Int 13h interrupt request. If the virus is resident, this call will return a value of 19h in AH, in which case processing returns to the control of the host file. Otherwise the virus is installed permanently in high memory by manipulating the system memory control blocks.

Operation

Once resident and active, this virus hooks Int 13h calls. This is firstly to answer the 'Are you there?', call and secondly to intercept read requests so that infection of files and disks can take place. There is no trigger routine and no payload although as mentioned above, there may be some damage to disks and files which cannot be repaired.

"The novel method of storing the virus code in unused header space is undoubtedly the grande idée *of the virus author"*

Boot Sector Infection

As with the installation routine, there are two distinct infection mechanisms, one for boot sectors and one for files.

An Int 13h read request is intercepted and allowed to complete under virus control. The first word of the buffer that has been read into memory is then checked to see whether it contains the 'MZ' header. If it does not, processing jumps to the boot infection routine. This checks to see whether the request is for access to track zero of the first floppy drive - if it is not, processing jumps back into the original INT 13h routine. Otherwise the main infection routine is called.

This reads the boot sector of the target floppy into memory and checks it for infection. If it is not infected, the format of the floppy disk is checked and on 360k disks the original boot sector is written to track 0, head 1, sector 3.

On other floppy types this relocation routine is skipped and results in the corruption of the boot process described in the section on Installation. The code portion of the boot sector is

then replaced by the virus code and written back to track 0, head 0, sector 1 before processing returns to the routine which placed the original call.

On fixed disks, the process is slightly different - after reading the Master Boot Sector, the first entry of the Partition Table is checked and the address of the active Partition Boot Sector is collected. This record is then copied to track 0, head 0 sector 12 and an infected record is written in its place. Another bug here means that the fixed disk boot infection will not occur on certain types of machine because of some differences in the construction of the Partition Table.

File Infection Routine

Attempting to infect files by intercepting low-level system requests is always a hit or miss affair, especially within the confines of virus code, and this virus is no different. If the Int 13h interception finds the 'MZ' marker at the beginning of the buffer, the virus assumes that the buffer contains a valid header record for a standard EXE file.

It attempts to check this by first examining the file size field (files longer than 65023 bytes are rejected), testing how many relocation entries there are and rejecting it if there are too many. It then checks to see whether the program file requires memory beyond the file image (rejected if it does) and finally it checks that the header size is the standard 32 paragraphs (512 bytes).

All of these checks are presumably to ensure that the virus is dealing with a genuine EXE file which is the requisite format to enable infection to take place. Predictably, these checks are not sufficiently detailed and there will be occasions when other files (including text and data files) are injected with the virus code and thereby irreparably damaged. It should particularly be noted that some types of *Windows* executable files will malfunction after infection by this virus.

A successful infection consists of inserting the virus code (which is 451 bytes in length - hence the name) into the 'spare' space in the EXE file header. The file header usually contains the relocation information for the executable, and the start-up values of the registers. However, the length of the header is *usually* much larger than is required, and is therefore padded with zeros. This altered file will now no longer begin with the identifier 'MZ' and will henceforth be treated by DOS as a COM file, thus neatly sidestepping the need for virus self-recognition.

Since this space may not be 'spare', this is another point where corruption can occur. Once the buffer has been infected and rewritten to the disk, processing returns to the original caller.

Conclusions

As usual, the catalogue of errors associated with this virus is considerable but this is unlikely to prevent it from spreading very widely. The novel method of storing the virus code in unused header space is undoubtedly the*grande idée* of the virus author and he is presumably congratulating himself on his brilliance. However, this technique does not present any insurmountable problems for anti-virus vendors, and the virus is just another notch in the numbers game.

Sadly for everyone, this virus is now at large and is just another nuisance to add to the growing virus problem. Fortunately, this particular piece of code is not difficult to detect and identify and it will cause no problems to reasonably good anti-virus software.

Damage or corruption will usually be limited to the root directory structure of 360k floppy disks but may also occur occasionally in other files and on other disk formats.

BFD-451

Aliases:	None known.
Type:	Multipartite - infects DOS Boot Sector and some EXE files of less than 65025 bytes length.
Self Recognition:	
Files	Infected files no longer begin with the characters 'MZ'.
Disks	Compares 451 bytes at the offset 3Bh in the DOS Boot Sector.
Memory	Int 13h with AH=F0h returns AH=19h.
Hex Pattern:	(on disk, in files and in memory)
	`5FFA 2E8E 55F8 2E8B 65FA FB2E FF6D FC9C 80FC F075 04B4 199D`
Intercepts:	INT 13h for infection.
Trigger:	None.
Removal:	Disinfection of infected files is possible but it is best to delete and replace infected files under clean system conditions. Disinfection of fixed disks is possible by replacing the original DBS under clean system conditions.

VIRUS ANALYSIS 3

Tree - Leafing Through Users Disks

Boot sector viruses have shown themselves to be highly successful - the two most common viruses at the moment are widely recognised as being Form and New Zealand 2, a DOS boot sector virus and one MBS virus respectively. The redeeming feature of these is the fact that they are easy to disinfect since they must not prevent the system from booting, they store the original boot sector somewhere else on the disk, loading and running it after they have installed themselves in memory. This provides us with a safe and reliable short-cut to removal, by checking and copying back the original contents of the boot sector.

A few viruses exist which cut out this stage, and after installation continue the boot themselves. This may be an attempt towards trying to hamper any removal. Fortunately this comes a bit too late in the day to have any great impact on computer users, as a disk editor can still be used, and FDISK /MBR (in DOS5) and SYS can clear up most such boot sector viruses.

Found In The Wild

The Tree virus was discovered in the UK on a number of disks brought to a research lab by a visitor from China, along with a number of known viruses and a couple of virus scanners. These, unsurprisingly, did not detect the virus!

The virus is very similar to Azusa, which is itself a variant of New Zealand, albeit with different side-effects. On booting from an infected disk, the virus will go resident (taking 1K from the top of conventional memory) and randomly print the message 'Tree 92.3' and beep.

On floppy disks, the original boot sector is loaded in and run to continue; on a hard disk, the virus loads and checks the MBR itself, and then loads and runs the boot sector of the active partition.

In a Spin

Once memory-resident, accessing a floppy which is not currently spinning will cause an infection test. In most viruses, only a read or write or both will instigate an infection, but the test instruction used here intercepts many more commands - the author used a TEST instead of a CMP (compare) which is probably accidental - an all too easy slip to make during a hard programming session.

Also, the spin test is flawed - although most computers have no more than two floppy drives, there can be several, and this test only accounts for two-drive machines. This doesn't particularly impede the virus but indicates the slapdash way in which the programmer approached the task.

As an infection flag, the virus uses the offset of the initial JMP instruction - most genuine boot sectors jump over the data table at the start, perhaps only about 50 bytes, whereas the offset in Tree is some 200 bytes. It's not perfect but it does not have to be - it has an *a priori* chance of about one in about 500 of mistakenly thinking a disk is already infected, but this is hardly likely to be a major hindrance.

Trigger And Recovery

The original boot sector on floppies is stored at track 39, head 1, sector 7, one sector before that used by Azusa. It has the same problem that it may cause damage to data on 80-track disks, while if the virus code is overwritten the disk will become unbootable.

If an infection is attempted on a floppy which is already infected, a counter is incremented; after every 16 increments, the virus is written to sector 3 of the FAT, corrupting the disk. However, the disk is still be recoverable at this point, as there are two separate copies of the FAT stored on it.

TREE

Aliases:	None known.
Type:	Memory-resident Master Boot Sector.
Self Recognition:	
Disk	Checks the offset of the inital JMP instruction in the Master Boot Sector.
Memory	None.
Hex Pattern:	`f6c4 0274 5bf6 c280` `7556 501e 31c0 8ed8`
Intercepts:	Int 13h for infection.
Trigger:	Overwrites sector 3 of the FAT after attempting to re-infect media 16 times.
Removal:	Specific and generic removal is possible. Under clean system conditions replace contents of Master Boot Sector, using the command FDISK /MBR on systems running DOS 5 or later.

CONFERENCE REPORT

InfoExpo '93

InfoExpo is an annual event hosted by the *NCSA*, which is designed to tackle the complex issues of computer security and virus prevention. This year's conference was held at the *Sheraton Washington*, Washington DC. With six sessions running simultaneously, it was quite an event, and attempted to cover everything an IT Manager needs to know in the tricky information security arena.

This year the conference had been timed to coincide with 'National Computer Virus Awareness Day' in the United States. The Virus Awareness Day had been sponsored by *3M* and the *NCSA* - the idea being to raise the level of consciousness among the user community.

Virus Awareness Day

The day began with presentations by various groups, representing the anti-virus industry, industry in general, the press, and law enforcement authorities. The story was the same from every quarter: the United States needs laws to prevent the continued organised spread of computer viruses... *now*. As Sharon Webb (*X-lock Corp.*) put it, in a paraphrase of Gore, 'If the present shouts and we do not listen, the future will be silent.'

The highlight of the day was Peter Tippett's presentation to the 'Telecommunications and Finance Subcommittee of the House Energy and Commerce Committee.' At the time, the committee was investigating the area of toll fraud, encryption, viruses, data security and individual privacy.

The Virus Awareness Day Team. 'This is a 3.5 inch disk with a copy of *F-Prot* on it, the hottest scanner in the city. It could clean your disk right up. Are you feeling lucky, punk?'

At the hearing, the committee, chaired by Representative Ed Markey, was told about the dangers of computer viruses and hacking, and Tippett requested that firm legislation was needed now in order to stop the continued increase in the number of computer viruses.

Also present at the hearing was the editor of *2600*, 'The Hacker's Magazine', Emmanuel Goldstein. Goldstein was attempting to defend the information published in his magazine. He explained that 'hackers were not analogous to criminals. The common bond which we [hackers] all share is curiosity. We are not out to rip people off or invade people's privacy because we realise how precious they are'. He went on to explain that although *2600* explained how 'phone systems worked, it did not encourage its readers to break the law.

Goldstein's arguments gained little credence among an understandably sceptical panel. Markey likened *2600* to 'posting the combination of a house lock in the local grocery store, along with its address.' Did Goldstein think this was wrong? The committee and Goldstein agreed to disagree.

Multi-Threaded Conference

The opening address of the conference proper (held the next day) provided interesting food for thought. Lee Curtis, from *Deloitte & Touche*, discussed the implications of the World Trade Centre bombing. While many companies are partially prepared for a virus attack, it is surprising how easy it is to forget the more serious things which can affect the operation of IT within a company - for example, a large explosion is unlikely, but if it happens to your company, you will be glad that you had contingency plans drawn up.

The conference was split up into six different streams: Communications Security, Computer Viruses, Data Encryption, Disaster Recovery, Network Security and Physical Security. There were many familiar faces speaking, including representatives from *Microsoft, Novell, McAfee Associates* and *Central Point.*

The opening session in the Virus stream was a talk by John McAfee, who gave users an overview of 'Dangerous Virus Design Trends'. After such a cheerful talk, there was only one course of action: a stroll around the trade stands to see if anyone had the answer. Many of the usual exhibitors were present at the trade show, along with some new ones. One of the new faces in the anti-virus industry was *Digital Enterprises*, who was unveiling its new product, a hardcard which guarantees a clean boot, regardless of what the user does. This was coupled with an integrity checker, and, if the system does everything it claims to do, should provide a reasonably high level of protection against viruses - at a cost: flash memory is expensive.

Markey: 'I will be fighting to pass legislation to protect the privacy of consumers in Cyberspace.'

The hardware manufacturers were out in force (or so it would appear) as *JAS* was busy marketing its own hardcard, which is designed to protect executables from tampering. As the number of viruses continues to skyrocket, producing a virus scanner from scratch is becoming increasingly difficult, and soon (if not already) it will be almost impossible for new manufacturers to enter the virus-specific detection race. It is therefore likely that 1993 will see an increasing number of generic detection techniques. Who knows, maybe 1994 will be officially designated 'The Year of the Hard Card'!

Dinner Date

The high point of *any* conference is the conference dinner. This year the delegates were lucky enough to be addressed by Representative Ed Markey. Markey seems to be keen to set up meaningful computer-related laws.

When commenting on computer viruses, Markey explained that the committee took the problem seriously. 'Yesterday's hearing on toll fraud, invasion of privacy, and viruses made members of the Telecommunications Subcommittee well aware that real protections and programs need to be developed to protect people in Cyberspace. Because the network is a *human* creation, it will embody all the eccentricities, judgement, reason, sense and dreams we consist of ourselves, along with out flaws, weaknesses, and prejudices.'

But would any of this have any effect? Markey believes so. 'This year I will be fighting to pass legislation to protect the privacy of consumers in Cyberspace. I will be fighting to ensure access to advanced communications technologies for all Americans, regardless of social strata, regardless of whether they live in urban or rural areas... let's get the word out and educate users as to proper ethical conduct so that the electronic highway is safe for all its travellers.'

PRODUCT REVIEW 1

Mark Hamilton

VirusNet - What's in a Name?

Computer journalists are, by and large, a cynical breed of individuals who have a healthy mistrust of all marketing hype. There is nothing that delights us more than to uncover the 'real' truth and then pass that information on to the reader. *VirusNet* is a product which I therefore took great delight in reviewing, as it is not what it seems, making this as much a salutary tale as a review of the package.

Interesting Origins

When I first received the review copy of *VirusNet* from *Safetynet Inc*, I had absolutely no idea that it was a commercial version of a well-known shareware anti-virus package. However, my suspicions were aroused when I read the list of included files in the manual, and a screen-shot in some accompanying advertising material looked all-too familiar. A quick telephone call confirmed my suspicions: the product is in fact a 'designer label' version of *Frisk Software's F-Prot*, and like most designer labels, the vast majority of the price tag is paying for the name.

Virus Bulletin frequently exhorts its readers not to rely on any one single anti-virus package, rather to choose two (or more) packages which use markedly different technologies. In this case we have a piece of software which does not show its parentage, and it is completely possible that a user could decide to use *VirusNet* along with *F-Prot* - after all, *F-Prot* is excellent value for money! When they realise the similarity between the two products aren't the poor unfortunates who bought *VirusNet* going to feel hoodwinked?

Nowhere in the *VirusNet* documentation is there any mention of *Frisk Software* (the authors) and you have to look extremely hard within the program to find even a mention of the author. The name is there as a Copyright Holder, buried under 'Updates', but even then, it appears after *Safetynet's* own Copyright.

The version we were sent for review, 2.08A, is for all practical purposes exactly the same as the version of *F-Prot* that is currently in the file libraries of both *CompuServe* and *CIX*. I personally reviewed *F-Prot* for *Virus Bulletin* in December 1991 (pp. 21-23) shortly after 2.01 was released. I shall make reference to that earlier review throughout and the eagle-eyed among you will notice that the screen-shots included in this review bear a slight resemblance to those I used some eighteen months ago.

VirusNet was delivered on a write-enabled 3.5-inch, 720 kilobyte diskette which was accompanied by a 48 page, saddle-stitched manual. The first seven pages provide a simplistic definition of a computer virus: 'A program that modifies other programs by placing a copy of itself inside them' - a definition that is not by any means complete and, on its own, not even entirely accurate. However, this is followed by some practical advice on what to do if disaster strikes: 'DON'T PANIC!' and an exhortation to protect yourself with regular backups. The manual continues with installation instructions and usage details of the package.

Unlike many other software packages, neither *VirusNet* or *F-Prot* needs to be installed on a hard disk; they can quite happily be run from a floppy. Alternatively, one can either use the 'Configure' menu option to copy the files to the hard drive or simply copy the files from the DOS prompt.

Excellent Detection

VN is the same program as *F-Prot* and, in point of fact, has changed very little since I last looked at it in detail. The interface is the same, its scan speed seems a fraction slower (although it is detecting many more viruses and trojans than it did then) but its options remain the same. In 1991, I criticised *F-Prot* for not allowing the user to be able to tailor the standard list of executable files which it examines when an 'executable files only' scan is initiated. This restriction remains, and it only deems files whose extension is APP, COM, EXE, OVL and OVR as being executable-type files.

Unfortunately, life is never easy and there are considerably more 'standard' executable extensions following the rise in popularity of *OS/2*, *Windows*, and *Visual Basic*. Extensions such as DLL and VBX could well be the target of attack by those viruses that routinely infect overlay files, for example. Fortunately, VN (and *F-Prot*) allows you to scan either every file on your PC or just those files with a particular extension.

In terms of its detective capabilities, I can find no fault with this product. It found all those in the 'Common', 'Standard' and 'MtE' Test Sets and an impressive 1,038 out of 1,059 in my unofficial test-set. Its scan speed is also impressively fast (see the summary table for full details). While writing this review, I simultaneously ran some detection tests on my PC in the background - an easy task under *OS/2* 2.1. It scanned all 4,251 files which occupy 86.9 megabytes in just under six minutes - pretty nippy, under the circumstances.

Heuristic scanning is an imprecise science - maybe 'black art' is a better way of describing it. It is not something in which you should put too much faith or reliance, as the technology is still rather immature. *F-Prot* and VN offer this facility and, in December 1991, the former was prone to flagging completely innocuous files. Since that time, *Frisk Software* has dramatically improved the search algorithms and the number of false alarms has been vastly reduced.

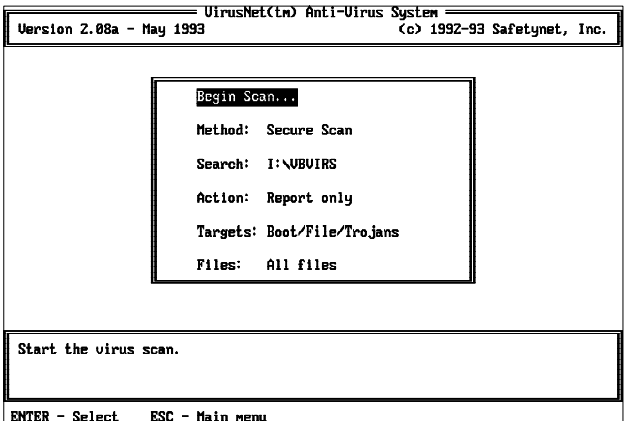
It was not without fault though: for example, it did not like a program I have which saves and restores Boot Sectors [*Some may argue that this is in fact a correct result, given the action of the utility. Ed.*] and it informed me that one of my TSR programs was an invalid program. This is still much better than the majority of other heuristic scanners on the market, and the developers deserve credit for this.

Resident Protection

VIRSTOP is a TSR that scans files as they are loaded into memory and prevents the execution of any that are infected with a virus. It also scans executables that are copied or otherwise accessed, and checks the boot sector of any floppy disk inserted into a diskette drive, the first time that disk is accessed. In 1991, I complained that I could not get VIRSTOP to work with DOS 5 or *Digital Research's* equivalent. I am happy to report that this has been fixed, and VIRSTOP now works exactly as advertised.

VIRSTOP uses exactly the same virus signature database files as VN and has identical detection properties. In addition, it performs some rudimentary analysis of program files and will warn if any perform virus-like activity such as self-modification or relocation to another part of memory. In addition, it warns if the executable has been compressed with (for example) PKLITE, DIET or LZEXE, since VIRSTOP is unable to determine if that executable was infected prior to it being compressed.

I am not a particular fan of TSRs, particularly for virus detection, because there is too high a risk that the anti-virus TSR will conflict with other TSRs or programs. However



A directory is specified and a secure scan commences. In this instance the scanner is instructed simply to report the presence of any viruses, although *VirusNet* can be configured to disinfect files.

VIRSTOP appears to operate flawlessly, although it does require some 15K with its signatures loaded into memory. This can be reduced to just over 3K if one elects to keep these on disk - however this adds to processing time as they have to be reloaded each time the TSR activates.

VIRSTOP contains a number of command line options, only one of which is documented in the manual (that to minimise the TSR size by keeping signatures on disk). Other options disable the memory scan when VIRSTOP is initially run, disable the boot sector checking and disable the checking of files during copy operations. It is a pity that *Safetynet* did not see fit to document these useful options. [*VB is told that these will appear in the next update of the manual. Ed.*]

Bits and Pieces

DO-ONCE allows the scanner to be run during bootup or network login at predetermined intervals - either once per day, once per week on a particular day of the week or on a particular day of the month. It is designed to be placed in a batch file, or in the network login script, to ensure that the software is run at set times or at the first opportunity thereafter. It works - what more can I say!

The authors have also provided a workstation deployment program called VNDEPLOY which copies the software from the network server to each user's local hard drive when that user logs in. I was unable to test this program, but assuming it operates as advertised, both DO-ONCE and VNDEPLOY add some useful additional functionality to the standard *F-Prot* package. Neither of these programs is loaded by VN's built-in installation process, although *SafetyNet* assures me that this will be updated in later versions.

Conclusion

I have absolutely no gripes regarding the technical aspects of this software, but I do question the sense of the commercial marketing of software that is almost identical to that available as shareware. *Safetynet* should make *VirusNet's* parentage much clearer to potential and existing users - although this is unlikely, as this would surely lose them some clients. It is not that *VirusNet* offers poor value for money *in itself*, it is that *F-Prot*, as a shareware product, is such tremendous value that it is very hard to compete with. *VirusNet* does give you more - but not a lot.

Personally, I cannot, in all honesty, recommend that anyone purchases *VirusNet*, unless one suffers from an overwhelming desire to own a saddle-stitched manual, some technical support, and two extra utilities. I can, however, wholeheartedly recommend that you download (from *CompuServe* or *CIX*) essentially the same product, *F-Prot*, which will give you very high quality virus detection at a bargain price.

VirusNet

Scanning Speed

Hard Disk:

Turbo Mode (950.2 Kbytes/sec)	17 secs
Secure Mode (496.0 Kbytes/sec)	60 secs

Floppy Disk:

Turbo Mode (28.2 Kbytes/sec)	11secs
Secure Mode (26.3 Kbytes/sec)	20 secs

Scanner Accuracy

'*VB* Standard' Test-set[1]	Turbo	364/364
	Secure	364/364
'InThe Wild' Test-set[2]	Turbo	116/116
	Secure	116/116
'MtE' Test-set[3]	Turbo	1536/1536
	Secure	1536/1536

Technical Details

Product: *VirusNet*

Version: 2.08A

Serial Number: Not stated.

Author: *SafetyNet Inc.*, 55 Bleeker Street, Millburn, NJ 07041-1414, USA.

Telephone: +1 201 467 1024 or 1-800 851 0188 (USA Only)

Fax: +1 201 467 1611

Price: $70 per PC.

Test Hardware: All tests were conducted on an *Apricot Qi486* running at 25Mhz and equipped with 16MB RAM and 330MB hard drive. *VET* was tested against the hard drive of this machine, containing 1,645 files (29,758,648 bytes) of which 421 were executable (16,153,402 bytes) and the average file size was 38,370 bytes. The floppy disk test was done on a disk containing 10 files of which 6 (310,401 bytes) were executable.

For details of the test-sets used please refer to:

[1] Standard test-set: *Virus Bulletin* - May 1992 (p.23)

[2] 'In The Wild' test-set: *Virus Bulletin* - January 1993 (p.12)

[3] 'MtE' test-set: *Virus Bulletin* - January 1993 (p.12)

PRODUCT REVIEW 2

Keith Jackson

RingFence: Keeping Tabs on your Disks

This month's review discusses a product called *RingFence*, produced by *S&S International* (of *Anti-Virus Toolkit* fame). It is very different from the myriad scanners and checksummers that are discussed constantly in reviews, in that it attempts to prevent viruses from having any effect by forming a barrier to prevent their introduction. To quote the developers: '*RingFence* is a terminate and stay resident program (TSR) which monitors floppy disk activity'. As such it is similar to other disk authorisation products such as *DiskNet* (*Reflex Magnetics*) and *D-Fence* (*Sophos Ltd.*)

Functionality

RingFence protects floppy disks by testing that all floppy disks used by a *RingFence*-protected system have previously been appropriately marked. If a non-*RingFence* floppy disk is detected, the operating system is prevented from accessing the information stored on the floppy disk. If a *RingFence*-protected floppy disk is used on a non-*RingFence*-protected computer, then it will be accepted as valid when it is returned to the *RingFence*-protected system only if the information held on the floppy disk has not been altered.

Remember that a virus can only be introduced by altering the content of the floppy disk. If any alteration has been made then the floppy disk must be re-validated (this can be automatically combined with virus scanning) before it can be used on the *RingFence* protected system once again.

A prerequisite for this is that *RingFence* provides boot protection (the hard disk is not accessible after a floppy disk boot). As most PCs boot from floppy disk if one is present in the floppy disk drive, it is impossible to prevent any software security system from being bypassed by a floppy disk boot, unless after such a boot, the hard drive is inaccessible.

RingFence is designed in just such a manner, and claims that 'even most low-level programs will be unable to access the hard disk'. It certainly confounded my efforts.

When a *RingFence* protected PC is booted from hard disk in the normal manner, a small TSR program (<4 Kbytes) is executed which can recognise *RingFence*'d disks. The process of marking the floppy disk is called validation. Encryption of the floppy disk is offered as an extra feature which helps to ensure that information held on *RingFence* floppy disks is not available to non-company machines.

The documentation that comes with *RingFence* is quite short (27 pages of A5), but it is easy to use and well indexed. There are a few surprising omissions (see below), but these can no doubt be remedied quite easily; overall my main criticism is that technical details are almost completely missing from the manual.

DoubleSpace Disaster!

The installation and de-installation sections are particularly clearly written, and occupy more than a quarter of the manual. This made testing usage of *RingFence* very straightforward, though I must confess to one problem which was entirely of my own making.

Included with the *RingFence* manual was a single sheet of paper explaining in very large print that *RingFence* is not 'currently' recommended for use with the *DoubleSpace* compression option which comes with *MS-DOS 6* - if you want to try this you are recommended to back up all the data on the target disk. This was like a red rag to a bull - *DoubleSpace* was installed on my test computer anyway (remember the review in the May 93 issue of *VB*?), so casting caution to the wind I decided that I would try it anyway. This was a Mistake [*With a capital M... Ed.*]

I can confirm that *RingFence* does not live happily with *DoubleSpace*, as after following the installation instructions very carefully, and using *RingFence* for just a few minutes, *MS-DOS* produced the ominous message 'Bad or missing command interpreter'. This message was not in error; my hard disk had 'disappeared', booting from the hard disk proved impossible, and messages about the contents of CMOS memory being corrupt were also displayed.

I'd like to state clearly that none of the above was 'caused' by *RingFence* performing erroneously. I was warned not to use the product with *DoubleSpace*, and the disagreeable results were entirely my own fault.

Installation

Given that one does not exhibit the stupidity demonstrated so clearly by my actions described above, installation and use of *RingFence* is very easy. The INSTALL program requires confirmation that a backup of the hard disk has been made, and entry of a password (8 alphanumeric characters) specific to this *RingFence* installation. A formatted floppy disk must be provided which is used to create a recovery disk. Once the questions are answered, the actual installation process itself takes about 2 seconds. After installation, one executable program (RF.COM, a hidden file), and two data files (RF001.SCR and RF002.SCR) are installed in the root directory of the hard disk. The AUTOEXEC.BAT file is altered to execute RF.COM as early as possible.

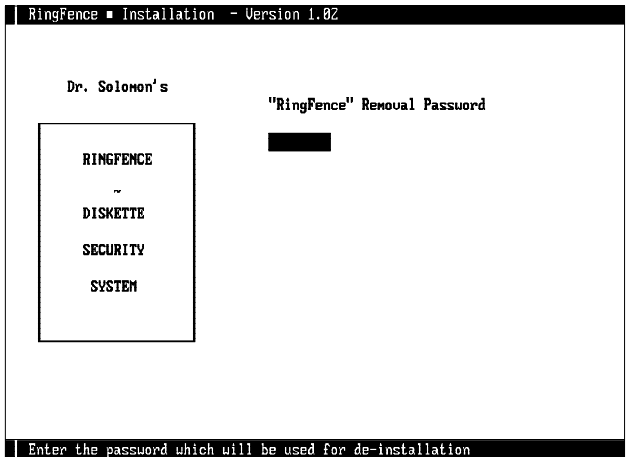
Note that these files are installed in the root directory, and there is no option available to store them elsewhere. I have a personal aversion to programs that clutter up the root directory in this way. Rather curiously, after *RingFence* had been used with *DoubleSpace*, a third hidden file appeared (RF003.SCR) before things went awry. I have no idea what this signifies as the documentation does not explain what these data files are used for.

The Lord of The Ring...

RingFence must be controlled by a supervisor, whose job it is to validate floppy disks, install/de-install *RingFence*, and introduce encryption when necessary. All these functions are available to the supervisor through a program called RFMASTER which is stored on the *RingFence* master floppy disk. With the cussedness typical of computer software, this program could not be executed once *RingFence* had been installed on my PC, as it is not a validated disk. Therefore RFMASTER could not be accessed/executed. Typical.

The only solution that I could find to this problem was to deinstall *RingFence*, copy the RFMASTER program to the hard disk, reinstall *RingFence*, and execute RFMASTER from hard disk. This problem is not explained in the documentation, or if it is I cannot find it, and I would contend that a non-technically minded supervisor might well be stumped by it for some time.

Once I crossed the Rubicon and obtained control of things with RFMASTER, it proved very simple to use - I wish that all security systems had such straightforward front-end software. *RingFence* seems to carry out its stated operations as described in the documentation: every time that a problem was detected, it produced a warbling noise as a warning, and displayed a message stating 'RingFence alert, Foreign Disk' in the middle of the screen. For some operations this warning appeared several times.



RingFence could be just the thing for your site if you have a large number of PCs, and provides a first-line of defence against viruses.

Note that it is vital that the supervisor checks the content of each floppy disk before it is validated. This means checking that the files on the disk are actually allowed to be introduced on to the PCs controlled by *RingFence*, and more importantly that no viruses are present. If a virus is present when *RingFence* validates a disk, then it can spread in exactly the same manner as if *RingFence* were not in use.

The *Dr Solomon's Anti-Virus Toolkit* (from the same developer as *RingFence*) is recommended, and can perform this task satisfactorily. A supervisor should also take into account that files can be introduced to a PC by a variety of routes (e.g. a modem), and if a network is present then this modem could be attached to any computer on the network. This reinforces the fact that *RingFence* is only really one component of a 'solution' to improvements in security.

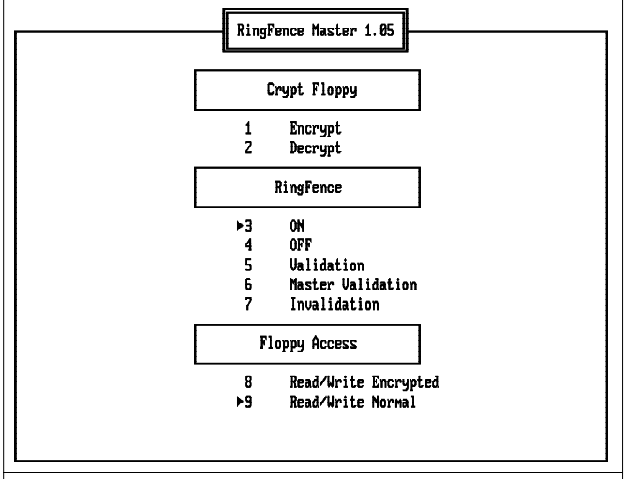
The manual discusses at length the problem of installing software packages which arrive on permanently write-protected floppy disks. The only workable solution is to turn *RingFence* off temporarily, boot using the *RingFence* master floppy disk and a clean DOS disk, then make copies of the new software on *RingFence* validated disks. This may seem an onerous task, but it is exactly this sort of control which the software is designed to provide!

Disk Encryption

The supervisor can encrypt the entire contents of a floppy disk to ensure that information on the floppy disk cannot be accessed outside the *RingFence* protected area. I carried out tests to ensure that all of the floppy disk was encrypted, and as far as I could tell this seems to be true.

No matter what number of files were present on the disk, RFMASTER always took somewhere between 2 minutes 7 seconds and 2 minutes 15 seconds to encrypt (or decrypt) a 3.5 inch 720 Kbyte floppy disk. When individual files were copied from an encrypted floppy disk, the time to copy the files across never increased by more than 5% when encryption was introduced - indeed a nearly full disk showed an overhead of just 2.8%.

These results show that the manual is quite correct to state that there is only 'a small performance penalty' when encryption is introduced. I would also contend that it also shows that the encryption is not very complex - it cannot be if such a small overhead is introduced. There are no details provided in the documentation of what encryption system is used, a lamentable omission which needs rectification before the encryption can be fairly judged.



Unlike many examples of security software, the *Ringfence's* user interface is easy to learn and use.

Jumping Over The Fence

How does *RingFence* work? My answer to this question is very simple - I don't know precisely. My tests show that changes have been made to the Master Boot Sector of the hard disk, and the boot sector of validated floppy disks. In addition, any changes to the FAT of floppy diskettes is picked up, *but* changes to the contents of the disk on a sector by sector level are not. This means that certain viruses may be able to sail through the *RingFence* protection. If encryption is used, this should be prevented, but this point is not brought out in the manual - indeed no technical information is brought out in the manual.

It is possible that the lack of technical detail in the manual is a conscious decision in an attempt to make the system harder to circumvent, and this is a familiar dilemma for vendors of security software. However, it does make the product harder to review, and I feel that the information given is just too sparse - users do need to know what is being done.

Version 1.0?

I am not quite sure which version of *RingFence* was tested for this review, as the banner displayed by *RingFence* at boot time stated that version 1.04 was in use, yet the floppy disk label stated quite clearly that it contained version 1.00. Somebody has got their wires crossed during manufacture.

Also, I am not sure why de-installation has to be carried out as a two-stage process. First the DEINSTALL program has to be executed, then a reboot is required, then DEINSTALL must be executed again to complete the process. All this is described in the *RingFence* manual, but nothing attempts to explain why a two-stage process should be necessary.

Whenever I tried to format a floppy disk, DOS always returned the rather puzzling error 'Invalid parameter', and refused to allow the formatting process to proceed. This should happen if encryption is switched on, but I found it happened under all circumstances on my test computer. The mere fact that the manual bothers to state that floppy disk formatting has been tested to work correctly with *RingFence* using versions of *MS-DOS* from 3.30 to 5.0, *Windows*, *Norton Desktop* and *Norton Commander* shows that the developers know that there is a problem lurking in the background as far as floppy disk formatting is concerned.

Causing strange DOS errors to be reported was not confined to the FORMAT program. For instance, any attempt to use the CHKDSK program on a non-validated floppy disk when *RingFence* is operational produces the curious error message 'cannot CHKDSK a SUBSTed or ASSIGNed drive', and if one tries to copy multiple files to the same disk the error message 'File allocation table bad, drive A' is displayed. The documentation fails to discuss this problem.

Conclusions

Overall I like the implementation of *RingFence*. The problems that I encountered were either self-inflicted, or, with the exception of being unable to format floppy disks under any circumstances, relatively minor. My only gripe with the system is its practicality. It is natural for me to reach for a floppy disk many times per day, indeed I have a store which currently contains nearly 1000 floppy disks. In my humble opinion, usage of *RingFence* requires that it is either used on *all* floppy disks or not at all. Am I going to install *RingFence* and validate all my disks? I think not.

Having said that, I can think of several sites where *RingFence* would be a complete answer to the problems that they have had for many months now. I think it is horses for courses - if you have a site which does not use a lot of floppy disks then *RingFence* could prove very useful indeed.

Technical Details

Product: *RingFence*

Developer and Vendor: *S&S international*, Berkley Court, Mill Street, Berkhampstead, Herts. HP4 2HB, Tel: +44 (442) 877877, Fax: +44 (442) 877882.

Availability: IBM or compatible PCs running DOS 3.1 or above. Disk space required: 20 Kbytes, TSR memory required: 4 Kbytes. Not fully DOS 6 compatible.

Version evaluated: 1.00 (or 1.04?, see text of review)

Serial number: TK1006352

Price: £18 per PC, dropping to £7 per PC for 250+ machines.

Hardware used: *Toshiba 3100SX*, 16MHz 386 laptop, with 5 Mbytes of RAM, one 3.5 inch (1.44M) floppy disk drive, and a 120 Mbyte hard disk, running under MS-DOS v6.0 (and DOS 4.01).

END-NOTES AND NEWS

Attention anti-virus software vendors! Would you like to include your product literature in the 1993 *Virus Bulletin* conference proceedings for a nominal charge? The conference, held on 9th and 10th September in Amsterdam is set to attract 300 computer security managers from around the world, making it the most prestigious conference there is on computer viruses. For further information or for conference booking contact Petra Duffield. Tel. +44 235 531889.

The *Business Software Alliance* has announced that the **level of software piracy has declined in most European markets**. After several years of rising piracy in Europe, the level fell from 1991, when approximately 77% of software used in Europe was illegal, to 66% in 1992. According to the *BSA's* estimates, the reduction in piracy in 1992 added approximately $700 million in revenue to the European software publishing and distribution industries, as total piracy losses in Europe fell from approximately $5.3 billion in 1991 to $4.6 billion in 1992. Tel. 071 491 1974.

Sophos has announced that its NLM, ***Sweep for NetWare* has been certified by *Novell* as a 'Tested and Approved' product** under *NetWare 4.0* as well as *NetWare 3.11*. Technical Manager, Richard Jacobs, said 'This confirms our commitment to virus detection on networks of all sizes.' Tel. 0235 559933

Alistair Kelman, the barrister acting for Paul Bedworth in the Bedworth hacking trial has been critical of the approach used by New Scotland Yard's *Computer Crime Unit*, saying 'Our successful defence has caused the *Computer Crime Unit* serious problems. They might now have to start prosecuting adult computer criminals rather than choosing the soft target of children.' New Scotland Yard is understandably annoyed by Kelman's comments. 'We do not have the benefit of being in the minds of the jury so we do no know what they considered important.' said DC Noel Bonczoszek, on behalf of the *Computer Crime Unit*. 'I doubt if the sentiments expressed by Mr Kelman would be looked upon with favour by any computer professional, particularly those who have suffered the attentions of a hacker. The computer criminal is often male and aged between 16 and 23 year so we can only play the cards we are dealt. If a similar case comes to light again it will be investigated in a similar way.'

Productivity Management Group Inc, has announced the launch of *STEP*, a new interactive Compute Based Training (CBT) security program. According to Mitch Zahler, president of the company, 'when we wrote *STEP* we focused on both the content of the program and how it can assist companies in protecting themselves. With *STEP* a LAN administrator or department manager can determine which employees pose a security threat due to lack of security knowledge. We have bundled an Anti-virus program with STEP [*F-Prot. Ed.*].' *STEP 2.0* retails for $99.95 and requires no special hardware. Tel. +1 (201) 669 8667.

It has been a disappointing month for avid Dr Solomon fans who are alleged to have queued for weeks in the rain for tickets to the first *Virus News International* conference, which was postponed this month. It is understood that letters of apology have been sent to the delegate. Tel. +44 792 324000.

STOP PRESS: If any users have had experience of any of the ARCV viruses, can they please contact DC Noel Bonczoszek, at New Scotland Yard. ARCV viruses include: Friends, HiDos Boot, Joanna, Jo Exerciser, Jo V1.11, More, Nichols, Reaper Man, Scroll, Scythe, Scythe 2, Slime, Small ARCV, Small EXE, Solomon, Spawn 1, Two Minutes, X-1, 2, 3A, 3B, Zaphod, any of the ARCV series, Benoit, ARCV-XMAS or Chad. Tel. +44 71 230 1177.

VIRUS BULLETIN

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel 203 431 8720, Fax 203 431 8165