ISSN 0956-9979 DECEMBER 1993

# VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL



## IN THIS ISSUE:

• **Yankee Virus hits *Sizewell B.*** Was the recent virus outbreak at *Nuclear Electric's* newest reactor site a real threat, or nothing more than media hype?

• **Neuron's Neuroses.** A 'fan' of *McAfee Associates*, who signs himself 'Neuron', has released a new collection of viruses devised to evade the *McAfee SCAN*. Does the file Part_1.Zip signal the beginning of the end for virus scanners?

• **Worm/Virus/Trojan.** ARJ-Virus could develop into a serious threat to various methods of virus protection: what is the best way to proceed?

## CONTENTS

# EDITORIAL

## Hype-Powered Reporting

The residents of Suffolk will doubtless have enjoyed a delightful sense of security following this month's furore over the virus outbreak at the *Sizewell B* nuclear power station. Imagine living in the shadow of this micro-processor controlled behemoth, only to discover that its very mind is riddled with computer viruses. Could such an event lead to a low-level format of East Anglia?

The public's perception of computer viruses is sketchy at best. When this is combined with a general fear of all things nuclear, the possibilities for a good story are endless. However, the important question is whether there was a real risk to safety. In this case, the answer is definitely no. So why has *Nuclear Electric* been castigated over a typical outbreak of a typical virus? The answer lies in the highly emotive nature of the issues involved (nobody actually explained why a virus on an office PC was worthy of national coverage) and in the public fascination with the various elements of which the story was made up.

*"The stigma which is seemingly associated with virus infection has no place in a healthy computing culture."*

The machines in question at *Sizewell* were not in any way responsible for the safety of the plant, its workers or the public. For such machines, the type of precautions taken were adequate: companies like *Nuclear Electric* do not use PCs for safety-critical functions. The important parts of the *Sizewell* system are armed to the teeth with backup systems, hardware overrides, safety trips and the like. Should *Nuclear Electric* have to install the electronic equivalent of prowling Dobermans, barbed wire fences and armed security guards to defend their non-critical systems in order to make the public *feel* safer? One would certainly hope not.

There should be no corporate stigma in a couple of machines becoming infected with a computer virus. If infected media were shipped out of a company, or lives endangered, the public would have a right to know. However, the fact that a handful of machines happen to be infected with the Yankee virus is hardly a national security issue. In the case of *Sizewell's* Yankee outbreak, the virus was discovered shortly after the machine had become infected - had the virus existed on the network for several months without detection, it is possible that the concern displayed might have been justified.

A little learning can be a dangerous thing. Although everyone is aware of the fact that computer viruses can spread from one PC to another, the popular misconception persists that viruses can jump platforms, with mainframes becoming infected by their less resilient cousins, the PCs. This is not the case, nor is it likely to become so.

The entire Sizewell virus outbreak has served as a reminder of the limitations of the IBM PC: it is not, and was never designed to provide, a secure working environment. For those applications which need to run with a very high degree of reliability, it is not the appropriate machine. The more security is added to a computer, the less usable it becomes - a fact which is particularly true for the DOS-based IBM PC. If misleading press coverage leads to the development of a security-paranoid culture, the result will be less efficient use of computers, making the end product more expensive to produce, be it sausages or nuclear power.

The entire computer virus issue is something which desperately needs good media coverage, based upon *fact*. Public humiliation of companies whose machines become infected does nothing but harm. The hysterical 'viruses invading our computers' style of reporting has planted seeds of distrust in computing which will grow to block out new and possibly useful thoughts and ideas.

The stigma which is seemingly associated with virus infection has no place in a healthy computing culture. If the wave of negative publicity generated by the *Sizewell* virus 'calamity' prevents companies coming forward and discussing the true scale of the virus problem, the price of using 'sensationalistic' journalism will have been a high one. The virus issue should not be swept under the carpet in the hope that it will go away. If the PC virus problem is not publicised in the right way, it will get worse - and the entire suppurating mass will have to be removed piecemeal. By making companies afraid of the brief sting of the antiseptic, the Press is endangering the entire limb.

VIRUS BULLETIN **©1993 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel +44 (0)235 555139. /90/$0.00+2.50**

# NEWS

## Getting away with IT

November 25th saw the launch of a new joint initiative between the *Metropolitan Police*, *IBM (UK)* and *PC Plus*. With the snappy catchphrase of 'Don't let them get away with IT', the sponsors of the venture hope to make the job of the computer criminal much more difficult.

The scheme was launched with a morning of presentations at IBM's South Bank offices. The speakers included Nick Temple (Chief Executive, *IBM (UK)*), Dave Veness (Deputy Assistant Commissioner, *Metropolitan Police Service*), Inspector John Austin (*Computer Crime Unit*), and Mark Drew (also from *IBM (UK)*).

The campaign is designed to help the users help themselves by protecting their own systems. Good computing practice was strongly advocated, with the usual pleas for regular backups, the judicious use of write-protect tabs on disks, and the scanning of incoming disks. The task of educating the user can sometimes be a difficult one: just by following these three simple steps, much of the damage caused by computer viruses could be eliminated.

Dave Pullin, *IBM's* Software Business Director, underlined how to utilise the best defence against computer viruses: the backup. 'As with so many things in life, we often don't appreciate the value of data until it is gone,' cautioned Pullin - a statement which anyone who has had first hand experience of the Michelangelo virus will know well.

However, the aims of the schemes go far beyond mere virus prevention. It is hoped that all aspects of computer crime can be combatted by relatively simple measures, though such preventative medicine has proven difficult to sell.

During the closing session, the most interesting point was raised: that of resources. It is no secret that computer crime requires many resources for its investigations. With the *CCU* consisting of only a handful of overworked officers, would the *Metropolitan Police* make any further resources available to investigate computer crime?

In reply, Inspector John Austin of the *CCU* said that it had sufficient resources at this time. However, when quizzed after the press conference, he admitted that in an ideal world, more resources would greatly help, and that the *CCU* had to fight for its budget, just like other specialist units in *New Scotland Yard*. How high on Scotland Yard's list of priorities is computer crime?

This worry, coupled with the impending loss of one of the *CCU's* most experienced officers, DC Noel Bonczoszek, is a cause for concern. The transferral is simply part of standard police staff rotations. Although Bonczoszek will be replaced by a new officer, the loss of his expertise will make the *CCU's* tough job even harder ∎

**Virus Prevalence Table - October 1993**

| Virus | Incidents | (%) Reports |
|---|---|---|
| Form | 18 | 36.7% |
| New Zealand II | 5 | 10.2% |
| Spanish Telecom | 5 | 10.2% |
| V-Sign | 4 | 8.2% |
| Cascade | 2 | 4.1% |
| NoInt | 2 | 4.1% |
| Parity Boot | 2 | 4.1% |
| 1575 | 1 | 2.0% |
| Brunswick | 1 | 2.0% |
| Eddie | 1 | 2.0% |
| Even Beeper | 1 | 2.0% |
| Exebug-1 | 1 | 2.0% |
| Helloween | 1 | 2.0% |
| Monkey | 1 | 2.0% |
| Necropolis | 1 | 2.0% |
| Tequila | 1 | 2.0% |
| V2P6 | 1 | 2.0% |
| Vacsina | 1 | 2.0% |
| Total | 49 | 100.0% |

## ITSEC Revisited

Four and a half months after the first meeting on the government's *ITSEC* product evaluation scheme, discussion of how best to certify anti-virus software still grinds on.

The second meeting of the *Anti-Virus Working Group* was held in London on November 3rd. The main objective of this group is to forge closer ties between the government and the private sector, and the aim of the day was agreement on recording virus prevalence and statistics gathering (the least controversial part of the master plan).

Discussion raged for the better part of the morning as to the best methods for recording and reporting incidences of virus outbreaks - it was eventually decided that an incident recording form, a draft of which was tabled, would be an effective way of achieving both. Many of those present already had some form of incident logging system, and so it was felt that the suggested system would not incur major changes in the current practice.

Delegates all agreed that information on attacks should be reported to the *Central Computer and Telecommunications Agency*, and that victims should be encouraged to report the incidents to the *Computer Crime Unit* at *New Scotland Yard*. The *CCTA* agreed to collate the data gathered, due to the commercially sensitive nature of the information ∎

## Viruses In the Wild

In a new cooperative effort led by *Symantec's* Joe Wells, a list of viruses known to be in the wild is being compiled. Current contributors to this list are Alan Solomon (*S&S International*), Dave Chess (*IBM*), Eugene Kaspersky (*KAMI*), Fridrik Skulason (*Frisk International*), Glenn Jordan (*Datawatch*), Joe Wells (*Symantec*), Paul Ducklin (*CSIR*), Padgett Peterson, Roger Riordan (*CYBEC*), Vesselin Bontchev (*University of Hamburg*), Wolfgang Stiller (*Stiller Research*), and Yuval Rakavi (*BRM*).

Rather than attempting to measure virus prevalence, the list is designed to show exactly which viruses are actually spreading. In order for a sample to be added to this list, an infected file or disk has to be received and verified by one of the members compiling statistics.

The following is a list of viruses confirmed to be in the wild, and should be of use to anyone interested in the epidemiology of computer viruses:

| CARO NAME | ALIAS |
|---|---|
| Barrotes.A | Barrotos |
| Butterfly | |
| Cascade.1701.A | 1701 |
| Cascade.1704.A | 1704 |
| Changsha | Centry |
| Chinese Fish | Fish Boot |
| Dark_Avenger.1800.A | Eddie |
| Dark_Avenger.2100.SI.A | V2100 |
| Datalock.920 | V920 |
| Den_Zuko.A | Den Zuk |
| Dir-II.A | Creeping Death |
| Disk_Killer.A | Ogre |
| Even_Beeper | |
| EXE_Bug.A | CMOS |
| EXE_Bug.C | |
| Fichv.2_1 | 905 |
| Filler | |
| Flip.2153.A | Omicron |
| Flip.2343 | Omicron |
| Form | |
| Frodo.Frodo.A | 4096, 100 Year |
| Green Caterpillar | Find, 1591 |
| Helloween | |
| Jerusalem.1244 | 1244 |
| Jerusalem.1808.Standard | 1808 |
| Jerusalem.Anticad.4096 | Invader |
| Jerusalem.Fu_Manchu | |
| Jerusalem.Mummy.2_1 | |
| Jerusalem.Zerotime.Austr | Slow |
| Joshi.A | |
| Kampana.3700:Boot | Telecom, Drug |
| Keypress.1232.A | Turku, Twins |
| Liberty | Mystic, Magic |
| Maltese Amoeba | Irish |
| Music_Bug | |
| Necros | Gnose, Irish3 |
| No_Frills.Dudley | Oi Dudley |
| No_Frills.No_Frills | |
| Nomenklatura | Nomen |
| November_17th.855.A | V855 |
| NPox.963.A | Evil Genius |
| Parity_Boot.B | |
| Ping_Pong.B | Italian |
| Print_Screen | PrnScn |
| Quit.A | 555, Dutch |
| Quox | |
| Screaming_Fist.696 | 696 |
| Stealth.BSTB | |
| Stoned.16 | Brunswick |
| Stoned.Azusa | Hong Kong |
| Stoned.Empire.Monkey | |
| Stoned.June_4th | Bloody! |
| Stoned.Manitoba | Manitoba |
| Stoned.Michelangelo | March 6 |
| Stoned.NoINT | Stoned 3 |
| Stoned.NOP | |
| Stoned.Standard.B | New Zealand |
| Stoned.Swedish_Disaster | |
| Stardot.789 | 805 |
| SVC.3103 | SVC 5.0 |
| Tequila | |
| Tremor | |
| V-Sign | Cansu, Sigalit |
| Vacsina.TP-05 | RCE-1206 |
| Vacsina.TP-16 | RCE-1339 |
| Vienna.648.Reboot | DOS-62 |
| WXYC | |
| Yale | Alameda |
| Yankee Doodle.TP-39 | RCE-2772 |
| Yankee Doodle.TP-44.A | RCE-2885 |
| Yankee Doodle.XPEH.4928 | Micropox |
| Yeke.1076 | |

The following viruses have only been seen by one member of the cooperative:

| CARO NAME | ALIAS |
|---|---|
| 10_Past_3.748 | |
| Brain | |
| Cascade.1701.G | 1701 |
| Coffeeshop:MtE_090 | |
| Darth_Vader.3.A | |
| Datalock.828 | |
| DosHunter | |
| Emmie.3097 | |
| EXE_Engine | |
| Flame | |
| Ginger | Gingerbread |
| Hafenstrasse | Hafen |
| Involuntary.A | Invol |
| Jerusalem.1808.CT | Capt Trips |
| Jerusalem.1808.Null | |
| Jerusalem.Carfield | |
| Jerusalem.Montezuma | |
| Jerusalem.Mummy.1_2 | |
| Jerusalem.Sunday.A | Sunday |
| Jerusalem.Sunday.II | Sunday 2 |
| Joshi.B | |
| Little Brother.307 | |
| Lyceum.1788 | |
| Murphy.Smack.1841 | Smack |
| NJH-LBC | Korea Boot |
| Ontario.1024 | SBC, 1024 |
| Parity_Boot.A | |
| Sat_Bug | Satan Bug |
| Sleepwalker | |
| Stinkfoot | |
| Stoned.Bunny.A | |
| Stoned.Empire.In_Love | |
| Stoned.Empire.Int_10 | |
| Stoned.W-Boot | |
| Swiss_Boot | |
| Swiss_Phoenix | |
| Syslock.Syslock.A | |
| Voronezh.1600 | RCE-1600 |

# IBM PC VIRUSES (UPDATE)

The following is a list of updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 25th November 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)
- **N** Not memory-resident
- **M** Infects Master Boot Sector (Track 0, Head 0, Sector 1)
- **R** Memory-resident after infection
- **P** Companion virus
- **L** Link virus

**Barrotes.1303**

**CER:** An encrypted, 1303 byte variant of the Barrotes virus, which activates on September 23rd.

```
Barrotes.1303   5F57 83C7 07B9 F904 2E80 2D?? 47E2 F9E9 DAFE
```

**Blinky**

**CR:** A 1302 byte virus, probably written by the same author as Pinky.

```
Blinky          8A26 0901 B9C2 04BE 0C01 8BFE FCAC 32C4 AAE2 FAC3 0E07 0E1F
```

**Checksum.1253**

**CER:** Very similar to the 1233 byte variant, but 20 bytes longer.

```
Checksum.1253   832E 0300 5083 2E02 0050 0BC9 740B 508C C040 8EC0 B449 CD21
```

**Clonewar.546**

**P:** A 546 byte long variant of this family of companion viruses.

```
Clonewar.546    93B9 2202 BA00 01B4 40CD 21B4 3ECD 21BA 5702 B903 00B8 0143
```

**Finnish.709.C**

**CR:** This variant was recently reported 'in the wild' in Finland. It is not significantly different from the original virus (which was first named F-709), and is detected by the same pattern.

**Halloechen.B**

**CER:** Almost identical to the original. Detected with the Halloechen pattern.

**Helloween.1384**

**CER:** A new, 1384 byte variant, detected with the Helloween search pattern.

**Mirror.B**

**ER:** 924 bytes long, just like the original, and with the same effect. Detected with the Mirror pattern.

**Never Mind**

**CR:** An encrypted, 838 byte virus. Awaiting analysis.

```
Never Mind      BB?? ??8B F3BF ???? B923 03B2 ??8A 0400 0530 1546 4781 FE
```

**No Frills.835**

**CER:** Similar to the 843 byte variant, but not fully analysed.

```
No Frills.835   3D32 5475 04B8 0710 CF80 FC4B 7418 80FC 3D74 1380 FC43 740E
```

**Nygus**

**CN:** The following three variants of the Nygus virus are much smaller than those reported earlier, and somewhat different (for example, these samples are non-resident). However, they are obviously related, and these three just seem to be earlier versions.

```
Nygus.163       B440 CD21 B002 E82B 00B1 A3BA 0501 B440 E82A 00B4 3ECD 21B4
Nygus.227       B800 40CD 21B0 02E8 3200 B1E3 BA05 01B4 40E8 3100 B43E CD21
Nygus.295       B440 CD21 B002 E841 00B9 2701 BA05 01B4 40E8 4C00 B43E CD21
```

**Osiris**

**CN:** This 299 byte virus activates on September 30th, where it has a 10% chance of displaying the message, 'Osiris Presents / The Trish Virus . Luv and Hugs OSiRiS'.

```
Osiris B939 00BE 0000 8A94 EF01 80F2 C646 B402 CD21 E2F2 B44E 33C9
```

**PC-flu.763**

**CR:** This 763 byte variant is quite similar to the 802 byte one. It is detected with the original PC-flu pattern. Not fully analysed.

**Pinky**

**P:** An encrypted, 952 byte companion virus, which contains the message 'The Pac-Man PINKY Ghost is watching (Can you find Inky?)'.

```
Pinky           8A26 0701 B958 03BE 0A01 8BFE FCAC 02C4 AAE2 FAC3 8A26 0701
```

**Pit**

**CN:** A simple, 492 byte virus that does not appear to do much but replicate.

```
Pit             438A 2780 FCE9 7403 B400 C383 C303 8A27 80FC 1274 03B4 00C3
```

**Pixel.300**

**CN:** A minor variant, detected with the Pixel.277 pattern.

**Pixel.847.Advert.C**

**CN:** A very minor variant, detected with the Amstrad pattern.

**Predator**

**C(E)R:** Five encrypted viruses are now known in this family. The 1072, 1137, 1148 and 1195 byte viruses only infect COM files, but the 2448 byte variant also infects EXE files.

```
Predator.1072   BA0C 02B1 ??FA 8BEC BC?? ??58 F7D0 D3C8 50EB 01?? 4C4C 4A75
Predator.1137   BA2E 02B1 ??FA 8BEC BC?? ??58 F7D0 D3C8 50EB 01?? 4C4C 4A75
Predator.1148   BA33 02B1 ??FA 8BEC BC?? ??58 D3C8 50EB 01?? 4C4C 4A75 F4
Predator.1195   BA4A 02B1 ??FA 8BEC BC?? ??58 D3C8 50EB 01?? 4C4C 4A75 F4
Predator.2448   0E1F BF?? ??B8 ???? B9BD 0449 7808 ???? ???? 4F4F EBF5
```

**Quadratic.1283**

**CER:** A polymorphic, 1283 byte virus which contains the string 'Quadratic Equation II'.

**Traveler Jack**

**EN:** Three new variants of this virus have now been discovered, 854, 979 and 982 bytes long. They are all encrypted, and the decryption loops have been modified so that no single search pattern can detect them all. The 979 byte virus is detected by certain virus scanners as a variant of the Flower virus, and examination revealed that the Flower virus should be re-classified as Traveler_Jack.883.

```
TravJack.854    8CC8 8EC0 2E8C 1E88 038E D880 3E02 0090 7416 BB36 008A 1602
TravJack.979    8CC8 8EC0 8ED8 803E 3100 0074 258A 1631 00BB 3700 8A07 32C2
TravJack.982    0E0E 5807 2E8C 1E0A 0450 1F8A 1631 00BB 3700 803E 3100 0074
```

**Trivial**

**C(E)N:** Several new viruses which belong to the 'Trivial' family are now known. The search patterns given below are shorter than normal, because the pattern would otherwise contain far too much of the actual virus code.

```
Trivial.26.B    2A2E 2A00 5656 B44E 5A41 CD21 83EA 62
Trivial.27.C    B43C CD21 93B4 405A CD21 C32A 2E2A 00
Trivial.28.C    2A2E 2A00 5656 91B4 4E5A CD21 83EA 62
Trivial.29      CD21 93B4 40B1 1D5A CD21 C32A 2E2A 00
Trivial.30.F    CD21 93B4 40B1 1E5A CD21 C32A 2E2A 00
Trivial.40.A    B440 B128 BA00 01CD 21B4 3ECD 21CD 202A 2E43
Trivial.40.B    BA00 0193 B440 CD21 B44C CD21 2A2E 636F 6D00
Trivial.40.D    40B1 2856 5ACD 21B4 3ECD 21B4 4FEB E1C3
Trivial.40.E    2A2E 3F3F 3F00 86F0 B43D B29E CD21 93B4 40BA
Trivial.40.F    0001 B440 CD21 B43E CD21 B44F EBE1 2A2E 2A00
Trivial.42.D    40B1 2ABA 0001 93CD 21B4 3ECD 21B4 4FEB DFC3
Trivial.42.E    40B1 2ABA 0001 CD21 B43E CD21 B44F EBE0 C32A
Trivial.43      40B1 2B56 5ACD 21B4 3ECD 21B4 4FEB E1C3
Trivial.44.D    40B1 2CBA 0001 CD21 B43E CD21 B44F EBE0 C32A
Trivial.45.D    40BA 0001 CD21 B43E CD21 B44F EBE1 C32A 2E43
Trivial.40.C    2A2E 434F 4D00 86F0 B43D B29E CD21 93B4 40BA
Trivial.44.C    8BD8 B440 CD21 B43E CD21 CD20 2A2E 636F 6D00
Trivial.102     B900 00BA 5301 CD21 720B B966 00BA 0001 93B4
```

In addition, several new search strings are included below to detect the new viruses in the PART_1.ZIP archive. [*See page 9. Ed.*]

```
Carioca.B       01FC F3A4 B800 01FF E02E 8B1E 0301 81C3 7C05 53B1 04D3 EB43
DA.2100.DI.B    D3E8 8CD1 4003 C18C D949 8EC1 BF02 00BA 0C01 8B0D 2BCA 3BC8
DataCr.1168.B   3601 014E 4E4E 8BC6 3D00 0075 03E9 FE00 8DBC DB04 BB00 01B9
DataCr.1280.B   3601 0183 EE03 8BC6 9090 9075 03E9 0201 8DBC EC04 BB00 01B9
Hymn.B          FF64 F500 07E8 0000 5E83 C6B4 FC2E 81BC 4207 4D5A 740E FA8B
Kemerovo.E      0400 89C7 B904 00A4 E2FD 525F 29D3 81EB C100 899D BB00 29C9
Wisconsin.B     8B0E 0601 8A04 34FF 8804 46E2 F7B4 1ABA 3901 CD21 E8B1 FDE8
Fu Manchu.D     B4E1 FCCD 2180 FCE1 7316 80FC 0472 11B4 DDBF 0001 BE20 0803
Fumble.867.E    5351 521E 0656 0E1F E800 005E 83C6 DCFF 4C16 837C 1603 7505
WW.217.D        BF00 0181 C6D2 01A4 A490 90A4 5EB4 4EBA C901 03D6 B9FF FFCD
PSQR.B          B80F FFFC CD21 3D01 0174 3B06 B8F1 35CD 218C C007 3D34 1274
Vienna.623.B    FC8B F2BF 0001 83C6 0A90 9090 A5A4 8BF2 B430 CD21 3C00 7503
Vienna.623.C    FC8B F2BF 0001 83C6 0AA5 9090 90A4 8BF2 B430 CD21 3C00 7503
MG.3.C          C43E 0600 49B0 EAF2 AE26 C43D 83EF DFEA 3902 0000 061F 8B75
YD.1049.B       EB10 1E5A 83C2 102E 0316 2000 522E FF36 1E00 061E 5053 B800
ACad.3012.C     B840 4BCD 213C 7890 7512 B841 4BBF 0001 BEC4 0B03 F72E 8B8D
ACad.Mozart.B   0F00 901F C31E 0633 C050 1FA1 1304 B106 D3E0 8ED8 33F6 8B44
Syslock.D       8AE1 8AC1 3306 1400 3104 4646 E2F1 5E59 58C3
Scott's Val.B   E800 005E 5690 5B90 81C6 3200 B912 082E 8034 ??46 E2F9
Perfume.BlankB  FCBF 0000 F3A4 81EC 0004 BFBA 0006 57CB 0E1F 8E06 5F00 8B36
Quiet.B         BB00 0153 5052 1E1E B800 008E D8BB 4000 A113 04F7 E32D 6708
Phoenix.800.C   B981 0151 31D2 AD33 D0E2 FB59 3115 4747 E2FA
```

# INSIGHT

## Sizewell B: Fact or Fiction?

Anybody who keeps an eye on UK newspapers will have noticed that in the last month, computer viruses have hit the headlines once again. The cause of this wave of media publicity was the infection of computers at the *Sizewell B* nuclear power station. The story, with perceived danger to the public, nuclear power, and computer viruses, had all the elements necessary to be highly newsworthy, and much of the portrayal bordered on the hysterical. The key question was whether a virus could compromise safety at the plant.

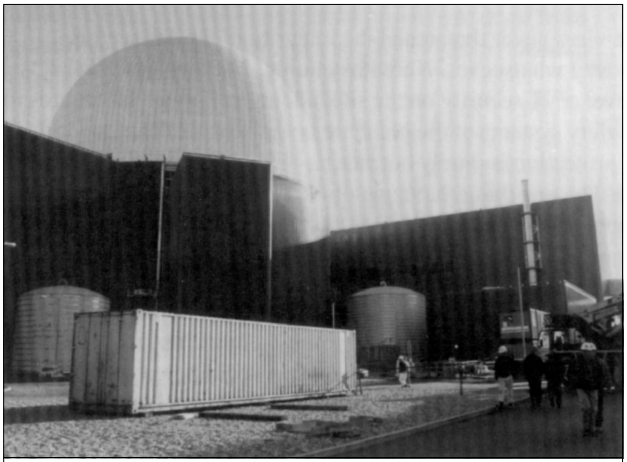
### Power to the People

As one drives up the A12 from London it soon becomes obvious that a large project is underway at *Sizewell* - the signs for the '*Sizewell B* construction traffic' start before Ipswich, and lead the traveller down increasingly small roads until he eventually arrives at *Nuclear Electric's* newest reactor site. The plant is situated on the east coast of England, near the sleepy town of Leiston: at first glimpse one has no idea of the size of the project. A number of power lines converge on the station from the surrounding area, and the white dome of the containment building stands out from the flat Suffolk countryside.

Upon my arrival at the plant, I was directed to my parking place beneath one of the towering pylons which was humming and crackling above me, and the true scale of the project began to dawn: at *Sizewell*, B clearly stands for big!

### Check your Disks Here

When anyone enters the site they have to pass through a security checkpoint. Here, the visitor is asked if he is carrying any computer media, and if so, the disks are checked for viruses. Somewhat dog-eared posters adorn the doors of the security checkpoint, reminding users that 'All computers must be checked' and appealing to everyone to 'B Safe' - the system has clearly been in place for some time, rather than just put up after the recent virus attack.



*Sizewell B*'s containment building, just one of the many different safety features built in to the reactor

The machines which became infected with the Yankee virus were not part of the controversial Primary Protection System, but of the construction team's *Local Area Networks* (*LAN*). 'Let me explain the different systems we have here,' said Dave Hollick, Site Manager. 'There are the construction computers, and split off from them are the computers which actually control the site. The construction computer systems are linked into a *LAN* running *OS/2*. Another 120 dumb terminals link into the *Nuclear Electric* mainframe system based off-site. So the virus never affected the control systems. What we have here is basically a standard office system, and it was this which became infected.'

'The 29th of June was the date it happened. We had a full investigation of the incident, and all members of the team were re-inducted. We then got some press coverage locally in the *East Anglia Daily Times*, and thought that was the end of it,' explained Hollick. 'The virus infected the *LAN* and we found out on the day it became infected - even if the trigger hadn't been so obvious, we would have found out the next day when people logged on to the system.'

The site policy is very strict. Every incoming disk should be checked by security at the door, although with a maximum of 5,000 people working on-site at any one time, this can be a gargantuan task. 'Each of the construction computers is checked for viruses when anyone logs on to the network, and since the Yankee outbreak, we have installed a new tool, *PC Guard*, so that it is impossible to run unauthorised software from floppy disks,' Hollick adds. 'We have three different virus scanners: *Dr Solomon's Anti-Virus ToolKit, Central Point Anti-Virus* and *Norton Anti-Virus*. Computer security is something which we take very seriously.'

With so many different people using the site, it was probable that sooner or later, a computer would be infected by a virus. In this eventuality, would there be any threat to the safety of the plant? 'Absolutely not!' exclaimed Len Green, Press Officer. 'The safety systems of the plant aren't run on PCs. If you are using mission critical software, you have to ensure that computer corruption cannot make things unsafe.'

### Fail Safe

The easiest way to minimise the effect of computer error is having a large number of backup systems. The computers which actually control the *Sizewell* plant have the ability to shut the reactor down completely - was Green certain that they were not susceptible to virus infection? 'Yes. The

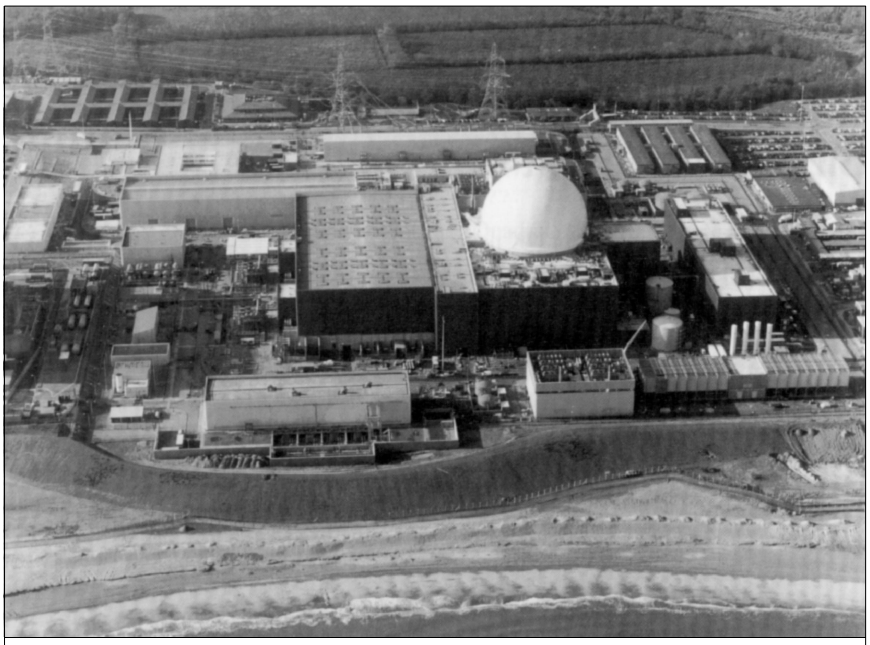
software itself is blown onto PROMs, and then that's that. An operator cannot add new code to the system. The most that can be done is that calibrations can be changed - something that is necessary in a system, however it is controlled.'

To anyone designing failure sensitive systems, the following precautions will be very familiar. The different parts of the system work on the principle of multiple layers of defence. The reactor itself is controlled and monitored by a dedicated system know as *WISCO* (*Westinghouse System for Centralised Operation*). This system is backed up by the reactor protection systems, the Primary Protection System (PPS) and the Secondary Protection System (SPS). It is the PPS which seems to have caused the most controversy. These protection systems would be used to shut the reactor down in the event of an emergency. How has *Nuclear Electric* made certain they are safe?

The PPS consists of over 100,000 lines of computer code. Although the system cannot possibly be infected by a computer virus (it is stored only on read-only memory), there is always the possibility of bugs. 'Let us assume for a minute that the Primary Protection System completely malfunctions,' explains Green. 'Imagine a fault develops and the system ups the power instead of shutting it down. At this point the SPS cuts in. That doesn't rely on computers at all, and cannot be overridden by an operator. Every safety critical feature of the plant is backed up: we don't rely on any one system alone for safety.'

### Media Attention

Given that safety at the plant was never compromised, how does Green feel about the way in which the story was portrayed? 'The frustration is that there are plenty of people who understand computer systems, who don't understand the way in which nuclear power works. These people don't know about the multiple fail-safes which we have.'



Hollick: 'We have three different virus scanners: *Dr Solomon's Anti-Virus ToolKit*, *Central Point Anti-Virus* and *Norton Anti-Virus*. Computer security is something which we take very seriously.'

'I'm still receiving calls from all over the place about this virus outbreak. I had a call from German television this morning - and the whole thing is a non-story!' With perfect timing the telephone rings… it is another call from the press. 'Things have been taken out of context, and the way in which it has been portrayed just has not been reasonable. I *understand* people wanting to know more - I want people to know more - but the system has not had a fair hearing. It makes my blood boil!'

From the half day spent at *Sizewell*, it certainly seems that *Nuclear Electric* takes the threat of viruses seriously, and is taking the right steps to prevent them spreading. 'What's the story? I carry this thing around,' Green holds up his laptop computer, which is covered in copious amounts of 'Virus Checked' stickers. 'I'm getting stickers at every location to show this computer has been virus-checked - look at it, it's covered. We take computer security very seriously here. We've already dismissed an agency engineer for using unauthorised software. I know that if I cut across established procedures, my job is on the line! That's been demonstrated.'

### The Last Word

It is clear that the Yankee virus never threatened the integrity of the *Sizewell B* computer systems in any way whatsoever. Notwithstanding, *Nuclear Electric* decided to increase the level of IT security on the site, adding still more safeguards to the office system. If the safety systems of the plant are completely isolated, does this mean that the extra virus protection is purely cosmetic - that is, security for security's sake? 'No, that's not true. The one thing that none of us in the nuclear industry can ever forget is that it is impossible to be *too* safe,' explains Hollick. 'Anything which makes the tools we use more reliable is always a good thing.'

Obviously there are lessons to be learned here for anyone responsible for running a mission-critical system. Firstly, if public alarm will result from a virus infection, this factor should be included in any risk assessment, and when deciding on security procedures. Secondly, the fact that *Nuclear Electric* made no effort to suppress the story acts in their favour: nothing looks worse than a bungled cover-up. Even in the nuclear industry, viruses are only a business problem. Having visited *Sizewell*, and seen their stringent security policies, it can be firmly stated that the *Sizewell B* 'incident' should be viewed in its true light: fiction, all too loosely based on fact.

# VIRUS UPDATE

## Part_1.Zip

*Fridrik Skulason*

The list of new viruses in the September 1993 edition of *Virus Bulletin* included two samples which had clearly been modified in order to avoid detection by one or more virus scanners. In each case, the modifications were minimal, but in the middle of the *Virus Bulletin* search pattern.

Both these viruses (BOMB.EXE and YONDER.COM) were uploaded anonymously in July, accompanied by a note from somebody who claimed to be located in the Netherlands, and signed with the alias 'Neuron'.

Variants such as these appear with monotonous regularity, and no special attention was given to the two samples… until they were sent to the Technical Editor of *Virus Bulletin*, as a part of a much larger collection. This collection is a 755,978 byte file named PART_1.ZIP, containing 266 files. A few of these files turned out to be duplicated elsewhere within the collection, while others were non-working or damaged viruses. However, the majority were new variants of known viruses.

### McAfee Targeted

Researchers quickly noticed several interesting features of the collection. It is very different from the typical virus collections which are obtained, directly or indirectly, from the 'underground'. Typically, such compilations contain a large number of 'garbage' files which are not viruses (for example, Trojan programs or completely harmless files). However, this one was unusually clean.

Not one of the viruses was detected by that version of *SCAN* from *McAfee Associates* current at the time, but other anti-virus programs fared significantly better, detecting 50%-95% of the viruses. The only conclusion possible to draw from this is that the viruses were specifically modified to avoid detection by *SCAN*.

The most likely scenario seems to be that the person(s) responsible obtained a virus collection somewhere, and either decrypted the search strings used by *SCAN*, or used an existing list of *McAfee's* search patterns. The viruses were then analysed one by one, and minor modifications made to the relevant part of the code. It should be added that the current version of *SCAN* (version 109) has been updated to deal with this collection, and it identifies 236 out of the 266 files as infected.

The name of the file (PART_1.ZIP) was worrying, as it implied that this was only part of a larger collection. So far, nothing more has been received, but researchers are concerned that one day 5000 new variants might be sent in!

### New Extensions

The relative cleanliness of the collection was not its only unusual feature. The extension of each file had been 'reversed': all files that were structurally EXE files had a COM extension, and vice versa. One can only speculate why this was done, but it may have been in order to defeat a primitive scanner to which the virus author had access.

The choice of parent viruses was also intriguing. Every virus in the collection was old - there was not a single virus family written in the past two years. This was not all: quite a few of the samples in the collection were classified as 'B' variants of the original virus, meaning that no other variants had been reported before. These viruses were either generally unavailable to the virus writing community, or were unpopular for some reason.

The names of the samples appear to have been selected at random, instead of indicating the family to which the virus belongs, or any text messages contained within them. In fact, one researcher commented that many of the names were quite good, and might be used later when a name for a new virus was needed. If readers ever see a virus called Boson, Discus, Saffron or Turtle (to name a few), this is where the name originated!

The changes made may have been carried out automatically by computer, or manually. Typical alterations are:

- Swapping two instructions
- Replacing an instruction with a different binary form (several instructions, such as XOR RW, RW, have two different forms)
- Replacing an instruction with a series of instructions having the same effect (for example, replacing ADD BX, 3 with three INC BX instructions)
- Replacing an ADD instruction with a SUB (or vice versa). This would typically involve substituting something like ADD AX, 100 with SUB AX, -100.

One cannot help but wonder why the author expended so much effort creating this collection, and then just uploaded it to a virus researcher, instead of spreading the viruses or uploading them to virus exchange BBSs. The fact that he seems to have targeted one particular product might indicate a particular dislike for that product or its producer.

The following list of viruses is printed so that any confusion about the correct names and identities of the viruses can be avoided. The name of the *VB* pattern which will detect the virus is given, along with the sample name (as shipped by the virus author), and the correct name of the variant. All viruses not detected by existing *VB* patterns have been added to this month's *Virus Bulletin* list of known PC viruses.

| Sample name | VB pattern | Name |
|---|---|---|
| 007.EXE | Burger | Burger.560.AF |
| ACAPULCO.EXE | 8-tunes | Eight tunes.B |
| ACID.EXE | Jerusalem-US | Jerusalem.1808.A-204.B |
| AIX.EXE | Amstrad | Pixel.847.Advert.B |
| AI_OKO.EXE | Vacsina | Vacsina.TP.5.B |
| ALASKA.EXE | *Syslock.D | SysLock.Syslock.D |
| ALMA_ATA.EXE | Jerusalem-1 | Jerusalem.1808.Anarkia.E |
| ALPHA.EXE | 1067 | Headcrash.B |
| APPOLO.EXE | Datacrime2 | DataCrime II.1514.C |
| ATARI.EXE | Oropax | Oropax.B |
| ATHENA.EXE | *Vienna.623.B | Vienna.623.B |
| BAKU.EXE | Taiwan-c | Taiwan.677.B |
| BANZAI!.EXE | Jerusalem-1 | Jerusalem.1808.Frere.D |
| BARBARA.EXE | 707 | USSR-707.B |
| BELINDA.EXE | Voronezh | Voronezh.1600.B |
| BENSON.EXE | *Queit.B | Stupid.919.Queit.B |
| BISTRO.EXE | JoJo | Cascade.1701.Jojo.C |
| BOMB.EXE | Eddie-2.d | Eddie-2.D |
| BOMBAY.EXE | Solano | Jerusalem.Solano.Dyslexia.B |
| BONNY.EXE | *Carioca.B | Carioca.B |
| BOSON.EXE | *DA.2100.DI.B | Dark Avenger.2100.DI.B |
| BOSTON.EXE | Yankee | Yankee Doodle.TP.44.D |
| BRAD.EXE | Violator | Vienna.Choinka.B |
| BRAZIL.EXE | Sat 14 | Saturday 14th.B |
| BROKEN.EXE | Burger | Burger.560.AB |
| BRONCO.COM | Icelandic_(2) | Icelandic.Saratoga.B |
| BULLDOG.EXE | W13 | Vienna.W-13.507.D |
| BUNKER.COM | Vcomm | Vcomm.637.C |
| BURLEY.EXE | Jerusalem-US | Jerusalem.1808.sUMsDos.AA |
| BURP.COM | Bestwish | Best Wishes.1024.C |
| BUTTER.EXE | Taiwan-c | Taiwan.708.B |
| CARTER.COM | Yankee | Yankee Doodle.TP.44.E |
| CIAO.EXE | 5120 | Vbasic.E |
| CLINTON.EXE | Lovechild | Lovechild.488.B |
| COACH.EXE | Jerusalem-US | Jerusalem.1808.A-204.C |
| COLLIDER.EXE | Dark Avenger | Dark Avenger.2000.Traveler.D |
| COLT.EXE | Virdem | Virdem.1336.German.B |
| CONDOM.EXE | Jerusalem-1 | Jerusalem.1808.Frere.H |
| COPENE.EXE | Doteater | Doteater.C |
| CRISIS.COM | Jerusalem-1 | Jerusalem.1808.sUMsDos.AB |
| CUT.EXE | Testvirus B | Testvirus-B.B |
| DAME.EXE | *Wisconsin.B | Wisconsin.B |
| DEACON.EXE | 2480 | Crew.2480.B |
| DELTA.EXE | Taiwan-2 | Taiwan.743.B |
| DILDO.COM | Wolfman | Wolfman.B |
| DISCUS.EXE | Violator | Vienna.627.B |
| EPHRAIM.EXE | Spanish | Traceback.2930.B |
| FEY.EXE | Ambulance | Ambulance.E |
| FILTH.EXE | Frodo | Frodo.H |
| FONDLE.EXE | Justice | Justice.B |
| FORD.EXE | *Scott's Val.B | Jerusalem.Zerotime.Scott's Valley.B |
| FORTUNE.EXE | Jerusalem-US | Jerusalem.1808.Blank.C |
| FUCK.EXE | Sunday | Jerusalem.Sunday.I |
| GABRIEL.EXE | Vacsina | Vacsina.TP.16.B |
| GAMMA.COM | *ACad.3012.C | Jerusalem.AntiCad.3012.C |
| GAMMA-7.EXE | Christmas-Japan | Japanese_Christmas.600.E |
| GATES.COM | Suriv_2.01 | Suriv 2.C |
| GENESIS.EXE | Halloechen | Halloechen.C |
| GETTY.EXE | *Kemerovo.E | Kemerovo.E |
| GET_LOST.EXE | 405 | Burger.405.C |
| GILLIGAN.EXE | *DataCr.1280.B | DataCrime.1280.B |
| GINGER.COM | MIX1 | Icelandic.MIX-1.F |
| GIZMO.EXE | Frodo | Frodo.F |
| GLUON.EXE | Jerusalem-US | Jerusalem.Groen Links.D |
| GOAT.EXE | ACAD-2576 | Jerusalem.AntiCad.2900.Plastique.D |
| GOON.COM | Jerusalem-1 | Jerusalem.1808.sUMsDos.AC |
| GOYA.EXE | Guppy | Guppy.D |
| GRASS.EXE | Black Monday | Black Monday.1055.G |
| GREECE.EXE | Slow | Jerusalem.Zerotime.Australian.C |
| GRISANTI.EXE | Amstrad | Pixel.847.Near_End.B |
| GUARDIAN.EXE | Sunday | Jerusalem.Sunday.H |
| HARBOR.EXE | Burger | Burger.560.AC |
| HEDGES.COM | Suriv_2.01 | Suriv 2.G |
| HENDRIX.COM | Suriv_2.01 | Suriv 2.D |
| HITMAN.EXE | Jerusalem-1 | Jerusalem.1808.Frere.E |
| HOLSTEIN.COM | Jerusalem-1 | Jerusalem.1808.sUMsDos.AD |
| HONGKONG.EXE | Armagedon | Armagedon.1079.D |
| HUMP.EXE | Shake | Shake.B |
| HUNGER.EXE | Traceback | Traceback.3066.B |
| IAN.EXE | *Hymn.B | Hymn.Hymn.B |
| IDAHO.EXE | Dr. Q | Vienna.648.AA |
| ILIAD.EXE | Jerusalem-US | Jerusalem.1808.sUMsDos.AE |
| INGRID.EXE | 2144 | Hymn.2144.B |
| ISIS.EXE | Taiwan-c | Taiwan.708.B |
| JACKSON.COM | Voronezh | Voronezh.1600.C |
| JEDDAH.EXE | Testvirus B | Testvirus-B.C |
| JOYGIRL.EXE | Interceptor | Vienna.Choinka.C |
| KENNEDY.EXE | Burger | Burger.560.F |
| KENTUCKY.EXE | Plastique1 | Jerusalem.AntiCad.2900.Plastique.C |
| KHEFRAI.EXE | *Fumble.867.E | Fumble.867.E |
| KICK.EXE | Sylvia | Sylvia.1332.E |
| KINKY.EXE | Burger | Burger.560.AE |
| KISS.EXE | Jerusalem-US | Jerusalem.1808.sUMsDos.AB |
| LA_BAMBA.COM | Black Monday | Black Monday.1055.E |
| LEPTON.COM | Icelandic_(1) | Icelandic.1.B |
| LICK.EXE | 1024PrScr | Zherkov.1023.B |
| LONDON.EXE | Diskjeb | Tenbyte.Diskjeb.B |
| LUCKY.EXE | SVC | SVC.1689.D |
| LUSTY.EXE | Jerusalem-US | Jerusalem.1808.Null.B |
| MARYLAND.COM | SVC | SVC.1689.B |
| MAYBE.EXE | Number of E | No. of the Beast.AA |
| MCAFEE.EXE | South Africa | Friday the 13th.540.C |
| MEPHISTO.COM | 2144 | Hymn.2144.C |
| MEXICO.EXE | Diskjeb | Tenbyte.Valert.B |
| MILLION.EXE | Pixel-277 | Pixel.277.B |
| MINISTER.EXE | W13 | Vienna.W-13.534.H |
| MISFIT.COM | Yankee | Yankee Doodle.TP.44.F |
| MOON.EXE | Dbase | Dbase.E |
| MOORE.EXE | Black Monday | Black Monday.1055.F |
| MUCK.EXE | Jerusalem-1 | Jerusalem.1808.Frere.F |
| MUD.EXE | Plastique1 | Jerusalem.AntiCad.2900.Plastique.B |
| MULE.EXE | Frodo | Frodo.G |
| NO.EXE | Bestwish | Best Wishes.1024.D |
| NOTHING.COM | Voronezh | Voronezh.1600.D |
| NUCLEAR.EXE | Yankee | Yankee Doodle.TP.44.G |
| NUT.EXE | *PSQR.B | Jerusalem.PSQR.B |
| NUTMEG.COM | SVC | SVC.1689.C |
| OF_COURS.EXE | Taiwan-c | Taiwan.708.B |
| OMEGA.EXE | 711 | Thirteen minutes.B |
| OORT.EXE | *Fu Manchu.D | Jerusalem.Fu Manchu.D |
| OREO.EXE | Destructor | Destructor.B |
| ORION.COM | Suriv_2.01 | Suriv 2.E |
| OSIRIS.EXE | MGTU | MGTU.273.B |
| OSLO.EXE | *DataCr.1168.B | DataCrime.1168.B |
| PASTOR.EXE | Vacsina | Vacsina.Joker.B |
| PEARL.EXE | *Phoenix.800.C | Phoenix.800.C |
| PEGASUS.EXE | Diskjeb | Tenbyte.Diskjeb.C |
| PENGO.EXE | Suriv_3.00 | Jerusalem.sURIV 3.B |
| PEPPER.EXE | Yankee | Yankee Doodle.TP.46.B |
| PERHAPS.EXE | VFSI | VFSI.B |
| PHOTON.COM | MIX1 | Icelandic.MIX-1.G |
| PILGRIM.EXE | *Perfume.BlankB | Perfume.765.Blank.B |
| PISS.EXE | Devil's Dance | Devil's Dance.D |
| PLAYBOY.EXE | Sunday | Jerusalem.Sunday.J |
| PLEIADES.EXE | MG | MG.2.D |
| PLEXUS.EXE | Attention | Attention.C |
| POSSIBLY.EXE | Oropax | Oropax.C |
| PRAVDA.EXE | VP | VP.C |

| Sample name | VB pattern | Name |
|---|---|---|
| PRICK.EXE | Victor | Victor.B |
| PULPIT.EXE | Burger | Burger.560.AD |
| PUSSY.COM | Vcomm | Vcomm.637.D |
| RHO.COM | Suriv_2.01 | Suriv 2.F |
| ROGER.EXE | Taiwan-2 | Taiwan.743.B |
| ROT.EXE | Subliminal | Jerusalem.Solano.Subliminal.B |
| SAFFRON.EXE | *YD.1049.B | Yankee Doodle.1049.B |
| SALEM.COM | SVC | SVC.1689.D |
| SALSA.EXE | GhostBalls | Vienna.648.AB |
| SALT.COM | Black Monday | Black Monday.1055.H |
| SANDY.EXE | *WW.217.D | WW.217.D |
| SCAM.EXE | Zero_Bug | Zero Bug.B |
| SCARE.EXE | Jerusalem-1 | Jerusalem.1808.sUMsDos.AG |
| SET.COM | December_24th | Icelandic.December 24th.B |
| SHANGHAI.EXE | Liberty | Liberty.E |
| SHARK.EXE | W13 | Vienna.W-13.534.J |
| SHIT.EXE | W13 | Vienna.W-13.534.I |
| SIGMA.EXE | Jerusalem-US | Jerusalem.1808.sUMsDos.AH |
| SIN.EXE | Parity | Parity.B |
| SIRIUS.COM | Jerusalem-1 | Jerusalem.1808.sUMsDos.AI |
| SLASH.EXE | 191 | Danish tiny.163.B |
| SMILE.EXE | GhostBalls | Vienna.648.AC |
| SMURF.EXE | Nina | Nina.C |
| SNAKE.EXE | Russian mirror | Russian mirror.B |
| SOHO.EXE | Kennedy | Danish tiny.Kennedy.B |
| SONAR.EXE | Datalock | Datalock.920.K |
| SQUID.EXE | Violator | Vienna.648.AD |
| STAB.EXE | Jerusalem-US | Jerusalem.1808.Blank.B |
| STALLION.EXE | Murphy_1 | Murphy.1277.B |
| STRIKE.EXE | 440 | No Bock.B |
| ST_PETER.EXE | W13 | Vienna.W-13.537 |
| SUCK.COM | Alabama | Alabama.C |
| SUSHI.EXE | Jerusalem-1 | Jerusalem.1808.Frere.G |
| TANGO.EXE | 707 | USSR-707.C |
| TASHKENT.EXE | Doteater | Doteater.E |
| TERRIER.EXE | MGTU | MGTU.273.C |
| THE_CULT.COM | Jerusalem-1 | Jerusalem.1808.sUMsDos.AJ |
| THE_THE.COM | Icelandic_(2) | Icelandic.Saratoga.C |
| THUNDER.EXE | *MG.3.C | MG.3.C |
| TONGA.EXE | *ACad.Mozart.B | Jerusalem.AntiCad.4096.Mozart.B |
| TRUST_ME.EXE | Dark Avenger | Dark Avenger.1800.F |
| TURTLE.EXE | Yankee | Yankee Doodle.TP.44.H |
| UTRECHT.EXE | GhostBalls | Vienna.GhostBalls.C |
| UZI.EXE | Burger | Burger.560.AA |
| VAGELOS.EXE | MLTI | Red Diavolyata.830.B |
| VEGEMITE.EXE | Devil's Dance | Devil's Dance.C |
| VENICE.EXE | Voron-370 | Voronezh.600.B |
| VERITAS.EXE | W13 | Vienna.W-13.507.E |
| WHORE.EXE | 492 | SI-492.C |
| WIDGET.EXE | 417 | F-you.417.B |
| WINDSOR.EXE | *Vienna.623.C | Vienna.623.C |
| WINSTON.EXE | 516 | Leapfrog.B |
| X-17.EXE | Zero Hunt | Zero Hunter.415.C |
| XXX.EXE | Voronezh | Voronezh.1600.E |
| YAHOO.EXE | Westwood | Jerusalem.Westwood.B |
| YELLOW.EXE | Diskjeb | Tenbyte.Valert.C |
| YONDER.COM | Cookie.b | SysLock.Cookie.B |
| ZAP.EXE | Taiwan-c | Taiwan.752.B |
| ZEUS.EXE | Jerusalem-1 | Jerusalem.1808.Anarkia.D |
| ZIMBABWE.EXE | -no pattern- | Flip.2343.B |
| ZULU.EXE | Yankee | Yankee Doodle.TP.39.B |

A '*' in front of the name of a search string indicates this is a new search string, first published this month.

The second group includes viruses which either did not replicate in testing, or have not yet been classified. Some of those samples are clearly damaged, and are incapable of replicating under normal circumstances.

| Sample name | VB pattern | Variant of... |
|---|---|---|
| 38-24-37.EXE | GhostBalls | Vienna.648 |
| ABRAHAM.EXE | | Vienna |
| AMWAY.EXE | Interceptor | Vienna |
| BAHRAIN.EXE | Agiplan | Month 4-6 |
| BENNY.COM | Jerusalem-1 | Jerusalem.1808 |
| BULL.COM | Icelandic_(1) | Icelandic.1 |
| CERTAIN.COM | Icelandic_(3) | Icelandic.2 |
| CHOLERA.COM | Icelandic_(3) | Icelandic (2) |
| CROTCH.EXE | | SysLock |
| DANIEL.EXE | | Int13 |
| DANZIG.EXE | Number of F | No. of the Beast |
| DICK.EXE | Zero Hunt | Zero Hunter |
| DINGO.EXE | Dr. Q | Vienna.648 |
| DISNEY.EXE | Anthrax | Anthrax |
| DOLPHIN.EXE | 1600 | Happy New Year.1600 |
| DONKEY.EXE | Dark Avenger | Dark Avenger.1800.G |
| DONNA.EXE | Sunday | Jerusalem.Sunday |
| EROTICA.EXE | Vienna-5 | Vienna.VHP.348 |
| EXPLODE.EXE | Jerusalem-US | Jerusalem.1808 |
| FELINE.EXE | Do_nothing | Stupid.583 |
| FLEMMING.EXE | Bebe | Bebe.1004 |
| GADGET.EXE | Sverdlov | Dark Avenger |
| GAY.EXE | Intercepter | Vienna |
| GEYSER.EXE | | Frodo |
| GINSENG.EXE | | Vienna |
| HADRON.EXE | | Phoenix.Proud |
| HIT.EXE | Number of | No. of the Beast |
| HONGKONG.COM | December_24th | Icelandic.December_24th |
| INTRO-1.EXE | GhostBalls | Vienna.648 |
| ISTANBUL.EXE | | Flash.688 |
| JENNY.EXE | Plastique1 | Jerusalem.AntiCad.3012 |
| JIHAD.EXE | | Vienna.435 |
| KAISER.EXE | Jerusalem-US | Jerusalem.1808 |
| KEY_WEST.EXE | | Vienna |
| KILL.EXE | | Hymn.Hymn |
| LLAMA.EXE | Lehigh | Lehigh |
| MALARIA.EXE | 1600 | Happy New Year.1600 |
| MARY_LOU.EXE | Kylie | Jerusalem.Kylie |
| MELON.EXE | Burger | Burger.560 |
| NICOTIN.EXE | Suriv_1.01 | Suriv 1 |
| NIXON.EXE | Burger | Burger.560 |
| NURSE.EXE | Suriv_1.01 | Suriv 1 |
| PARTICLE.EXE | | Vienna.644 |
| PEANUT.EXE | Vienna-5 | Vienna |
| PEROT.EXE | Datacrime2 | DataCrime II |
| QUARK.EXE | 696 | On 64 |
| RAPE.EXE | Burger | Burger.382 |
| RISUTORA.EXE | Jerusalem-US | Jerusalem.1808 |
| SACK.EXE | | Amoeba |
| SAND.EXE | | Vienna |
| SNOW.EXE | South Africa | Friday the 13th.416 |
| SUPER.EXE | MLTI | Red Diavolyata |
| SYPHLIS.COM | Vcomm | Vcomm.637 |
| TOTO.EXE | | Suomi |
| TURBO.EXE | | Number 1.AIDS.A |
| XYZ.EXE | Crazy Eddie | Crazy Eddie |

In addition, a few viruses were represented by several samples:

| Sample same | Identical to... |
|---|---|
| COLGATE.EXE | COACH.EXE |
| DISCOVER.EXE | DELTA.EXE |
| FOXTROT.EXE | DELTA.EXE |
| GRETHE.EXE | DELTA.EXE |
| KATYA.EXE | ISIS.EXE |
| LASER.EXE | HIT.EXE |
| LINGAM.EXE | HIT.EXE |
| MAESTRO.EXE | LLAMA.EXE |
| MURDER.EXE | COLLIDER.EXE |
| Q345.EXE | JIHAD.EXE |
| YES.EXE | OF_COURS.EXE |

**Editor's Note:** Any reader with any further information about the author of this virus collection should contact The Editor, *Virus Bulletin*, or *New Scotland Yard's Computer Crime Unit*. Tel. +44 (0)71 230 1177.

# VIRUS ANALYSIS 1

## The Monkey Virus

Monkey is a new boot sector virus, reported to be at large in Europe. Two samples were sent for analysis, differing in both content and the location of various sections of code. However, they are undoubtedly variants of the same virus, presumably written by the same individual. Monkey has no trigger routine, but can cause serious damage, due to the method of operation. Its name is contained at the end of the code in both samples, hidden by a simple encryption routine.

### Installation and Operation

This virus infects the Master Boot Sector of fixed disks when they are booted from an infected diskette. Processing begins by initialising the various code parameters needed. A request for available memory size is issued to the BIOS: one Kbyte is removed from the top of RAM, and the original system Int 13h vector is collected into the virus code. The virus' Int 13h interception routine is then hooked into the system, and a segment address is calculated, relocating the virus code to the top of memory. Next, the MBS of the first fixed drive is read into memory. Should signs of infection be found, the virus identifies where the original MBS is stored, reads it into memory, decrypts it, and returns control to the MBS, enabling booting to continue. If the fixed disk MBS is clean, the virus infects it, storing an encrypted copy of the original.

'Encryption' is rather a grandiose description: in both versions, each byte in the sector is simply XOR-ed with a value of 2Eh. This may be an attempt to make disinfection more difficult, but will present no difficulty to a good detection/disinfection program.

Once hooked, the virus intercepts requests to the disk access services. The infection routine is only called during 25% of read requests, making it slightly more difficult for the virus to replicate. Requests for read access to sector 1 or 2, head 0 on fixed disks or head 1 on floppies are routed through a routine which completes the request and examines the sector to see whether it is infected. If it is, the original MBS is collected and decrypted before returning to the calling routine. Requests for write access to the same sectors are treated slightly differently: a request to write to sector 1 or 2 of head 0 on a fixed disk is changed to a disk reset command, preventing virus code from being overwritten.

### Infection

Before attempting to infect the fixed disk, two checks are made. The first check is simply to prevent an attempt to infect an already infected disk. The second is more interesting: the virus appears to look for a specific type of boot sector (which may be part of an anti-virus package) and modifies its operations accordingly.

This first test is made by searching for the value 9219h at offset 01FAh in the MBS. If this is found, the infection routine is aborted. Should the first flag value not be found, the second is examined (see below). If it is not present, the virus writes a copy of its code to the MBS, and encrypts the existing MBS before writing it to an alternative sector (though always on Track 0). The position of this sector varies for different media:

| | Head | Sector |
|---|---|---|
| 360k floppy | 1 | 3 |
| 720k floppy | 1 | 5 |
| 1.2M floppy | 1 | 14 |
| 1.44M floppy | 1 | 14 |
| Fixed Disk | 0 | 3 |

On floppy disks, these positions represent the final sector of the root directory, and infection by the virus will destroy any file entries stored there.

The function of the second flag is more interesting. If the MBS contains the value 6150h at offset 0119h, the virus treats the second sector of the disk as if it were the MBS, writing the virus code to this sector.

The flag value of 6150h can be interpreted as the ASCII letters 'Pa': this may be part of the word 'Partition' which often appears in MBS code. This check appears to be an attempt to bypass a boot protection mechanism. If such a system is encountered, it is likely that infection will be unsuccessful, as the virus contains a serious bug which causes the machine to hang.

### Monkey

Aliases: None known.

Type: Master Boot Infector.

Infection: Fixed and floppy disks.

Self-recognition on Disk:
Value 9219h at offset 01FAh.

Self-recognition in Memory:
None.

Hex Pattern: (on Master Boot Sector or in memory)

```
83F9 0373 3A3A 3475 3680 FC02
740E 80FC 0375 2C80 FA80 7227
```

Intercepts: Int 13h Read and Write requests.

Trigger: None found.

Removal: Disinfection possible using specially written software.

# VIRUS ANALYSIS 2

## ARJ: a Place in the Archives!

*Eugene Kaspersky, Vadim Bogdanov*

The main thrust of most virus writers' work is the development of existing infection techniques. Optimization of virus code and the creation of elaborate new polymorphic algorithms are but a few of the ways in which the computer underground attempts to thwart scanner developers. Most new developments in the field are simply extensions of a well-known idea. For example, virus code might be inserted into the free space in an EXE header, rather than appended.

Now and then, however, virus writers come up with a completely new idea. When this happens, anti-virus software manufacturers must decide whether or not to modify their products to deal with a new infection strategy. The ARJ-Virus represents one such turning point for the industry: it is capable of infecting files *inside* ARJ archives.

### Compress and Save

ARJ-Virus is, in fact, more akin to a worm than to a standard DOS virus. It is 5000 bytes long, and adds code to compressed ARJ files. These compressed files, when unarchived and executed, infect other archives. One would assume that the task of adding code to these compressed files would be extremely complex, which in turn would make the virus very large. However, ARJ-Virus was sent to me complete with a copy of its C source code. This is approximately two hundred lines in length. How is it possible to do so much in such a short program?

When an infected file is executed, it searches in the current and in all the parent directories for any files which have the extension ARJ. If an ARJ file is found, the virus creates a temporary file and the extension COM. The filename is generated by randomly choosing four letters from the range A to V. The choice is restricted because the upper limit for letters used by the virus is 0Fh: thus, the virus has a range of fifteen letters from which to choose. Examples of typical filenames generated by this routine are BHPL.COM, NLJJ.COM, and OKPD.COM.

Once such a file is created, the virus copies itself into it, and appends a random number of 'garbage' bytes. These Trojan files range in length from about 5K (the length of the virus code) to 64K, the maximum allowable size of a COM file.

The virus then needs to add this file to the host archive. It does this in the easiest manner possible… by executing the archiving file, ARJ.EXE! This program allows users to compress and store one or more files (including subdirectories) in one or several archive [*Colloquially known as Arjive. Ed.*] files in compressed format. ARJ is one of the most popular archivers, like *PkWare's PKZIP*.

ARJ.EXE is designed to be called from the command line, and therefore has a raft of commands and switches which can be set when it is executed. One of these, the 'a' switch, tells the program to add particular files to a named ARJ file.

The virus uses this option to infect the host ARJ file, executing the following command line:

```
c:\command.com /c arj a <arj-file> <filename>.com
```

where <arj-file> is the name of the archive file about to be infected, and <filename> is the four bytes-long, randomly selected name described above. The '/c' switch causes COMMAND.COM to execute a program, and to exit immediately upon execution.

> *"This new virus … presents a new idea which could be developed into a real threat to certain approaches to virus protection"*

On execution of this command, the archiver ARJ.EXE compresses and adds this Trojan program to the archive file. The virus then deletes the temporary file and searches for the next ARJ file. If there are no other archive files in the current directory, the virus will jump to the parent directory. Should the current directory be the disk root directory, the virus returns to DOS.

### The Manual Virus

The virus described above is, under certain circumstances, capable of spreading. The most important requirement necessary for ARJ-Virus to work is the presence of the ARJ.EXE archive utility. The virus author has assumed that where ARJ files exist, so should the archiver.

Moreover, ARJ-Virus assumes that the archiving program is specified somewhere on the path. Though this would seem to limit the spread of the virus, it is likely that it is capable of replicating on a number of machines: if ARJ files are stored on a machine, it is probable that the archiving program is also present. Traditionally, this file would be located in a directory specified on the path, such as \BIN.

Another factor which limits the spread of this virus is its requirement that the user execute the Trojan file contained in the archive. Examining the situation from a psychological point of view, it does seem probable that the file will be executed. When the file is unpacked and examined, its contents will be seen to contain an extra executable file with a strange name. What is it, and will the user give in to the instinctive urge to execute the file to see what it does?

One of several different things will happen at this point. Firstly, the user may simply ignore the file, or not even notice its existence. Should this be the case, the virus will not spread. Secondly, the user may examine the file, and try to gauge its function. In the absence of further information, the most likely result will be file execution. By relying on the user to help spread the virus, the author has made it difficult to use traditional methods for virus detection.

As an experiment, I decided to ask ten people who work with computers every day what they would do if they unpacked an archive file and found that it contained an unknown COM file.

About half of my test-set replied immediately, 'Execute it!'. The others suggested that the contents of the file should be examined, and that if no further information came to light, it should be executed anyway! Not one of those whom I questioned suggested that the file should be scanned with anti-virus software before execution.

It is possible, after having read this description, that the reader will assume that this virus is less likely to spread than most common viruses. This, however, is not the case: most viruses rely on users executing infected files or leaving an infected disk in the disk drive - with care and attention it is possible to prevent almost all virus attacks.

### The Features

One of the more unusual traits of the ARJ-Virus is its ability to infect the same file many times over. The virus, by its very nature, cannot easily determine whether an archive file is already infected. Checking the contents of an archive file for the presence of the virus is quite a task, given that the Trojan COM file will be of variable size and name. This does not matter: ARJ-Virus has the look of a demonstration that a new idea works, not that of a finished product.

Although it contains no intentionally destructive code, the virus can still damage executable files under certain circumstances. Sometimes the filename chosen by the virus is the same as a file already present within the archive. In this case, the virus overwrites the file already within the archive with the newly created Trojan file.

The virus attempts to hide its presence by hooking Int 10h, the video interrupt. When the archive program is called, the virus simply installs its own Int 10h, which consists of an IRET instruction - i.e., all calls to the screen are ignored. If all goes well, and no errors are encountered, the infection process will be transparent to the user.

Unfortunately, if either DOS or ARJ.EXE displays an error message during this process, things go awry. In the case of the virus attempting to infect a write-protected disk in the A: drive, the infection process will cause DOS to attempt to display the familiar 'Write-protect' error message and wait for a keystroke. The user will see only a blank screen, making it look as if the computer has crashed.

### The Problems?

This virus raises new issues for anti-virus software developers. One problem pertains to behaviour blockers: how can the monitor intercept the legitimate request to add a file to the archive? I see no easy answer. Should the TSR display a warning about an ARJ file opening, or when COM files are opened or executed? This cannot be a good idea. A behaviour monitor would normally detect this virus when a new COM file is created. However, this is such a common occurrence that most users would ignore the warning.

The virus itself, once unpacked, is relatively easy to detect (it even contains the internal text string '*.arj .. 0000.com /c arj a c:\command.com'). However, searching for the virus in infected ARJ files is much more difficult.

How important is it that scanners should be able to detect archives infected with ARJ-Virus? How many different popular archive standards are in use in the IBM PC world? In order to add this function to anti-virus software, a great deal of development time, money, and EXE code bytes are required - a bill which would eventually be laid at the feet of the user. Scanners are already bulging from the steady influx of new viruses, and making them aware of many different compressed file formats will slow them down still further.

ARJ-Virus is quite primitive, and not a great security threat to PCs. However, it presents a new idea which could be developed into a real threat to certain approaches to virus protection. The idea of virus encryption introduced in Cascade grew up to be the MtE and TPE . Let's be ready.

## ARJ-Virus

| | |
|---|---|
| Aliases: | None known. |
| Type: | Non-resident Worm. |
| Infection: | Creates Trojan COM files inside archives compressed using ARJ. |
| Self-recognition on Files: | None. |
| Self-recognition in Memory: | None necessary. |
| Hex Pattern: | `558B EC83 C4EE E883 03B8 B614`<br>`50E8 3E0B 50E8 450B 83C4 04B8`<br>The pattern in infected archive files depends on the version of ARJ archiver stored on the host machine. |
| Intercepts: | Int 10h to disable the screen output. |
| Trigger: | None. |
| Removal: | Delete Trojan COM files from disk and within archives. |

FEATURE

# The Real Virus Problem

*Jim Bates*

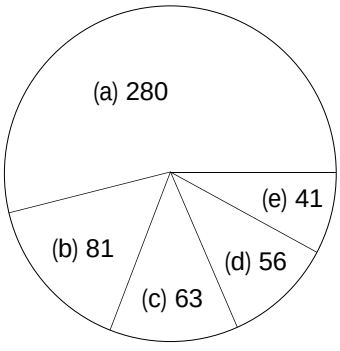
There has always been a pressing need for reliable information concerning computer virus activity in the real world: only by accurate assessment of the problem can an effective defence be created. Thanks mainly to the marketing efforts of the anti-virus industry around the world, the true extent of the problem has been efficiently concealed beneath a ragbag of pseudo-scientific projections, surveys, reports, forecasts and speculations. Here I present the findings of a recent survey of UK computer programmers, conducted without any input from the software vendors.

### Vital Statistics

The infamous Tippett Prediction appeared to forecast virus infections of galactic proportions by the end of this century. Since then, most of the information concerning virus prevalence has either been unabashed hyperbole and exaggeration designed primarily to frighten users into buying a particular anti-virus package, or simply gathered in such a way as to invalidate the statistics.

One of the biggest problems in this area is that, following the grossly overestimated predictions about Michelangelo prevalence, predictions from within the industry are seen to be self-serving at best. Many anti-virus companies experienced record sales in the scanning frenzy which preceded 'Michelangelo Day' in 1992, and ever since, the public has been understandably wary of industry-generated figures.

Academic discussion of the pros and cons of rare and exotic virus techniques, coupled with the magpie collection complex displayed by vendors and researchers intent upon playing the numbers game, may be very stimulating. Such counting, however, bears little direct relevance to the problems faced by computer users. Similarly irresponsible attitudes to virus writers themselves encourage a whole group of prospective 'researchers' to think it perfectly acceptable to write viruses for 'research purposes' and then pass them on to others, to swell their collections.




**Breakdown of virus type:** (a) Never had a virus. (b) Had a boot sector virus. (c) Unsure of virus type (d) Had a parasitic virus (e) Had both boot sector and parasitic viruses.

Those researchers genuinely concerned with helping users have had to rely upon verified reports of virus infections coming in through their own channels, as well as upon occasional statistics produced by other trusted organisations such as the Police. Until now, this is all they have had to enable them to evaluate the extent of the problem. We may, however, be seeing the beginning of a new trend, with the publication of the results of a survey conducted by the *Institution of Analysts and Programmers* (*IAP*). This organisation is dedicated to the promotion of excellence amongst computer professionals, and their survey represents the first truly independent attempt which I have seen to evaluate the real extent of the virus problem.

### Setting the Scene

Several fascinating revelations from the results of the survey confirm the reliability of the approach adopted by responsible researchers in the UK. First, existing figures seem to indicate that under 2% of known viruses are actually at large and causing problems for real computer users. Second, it appears that there is a slight preponderance of boot sector over parasitic viruses, despite the fact that parasitic types form the vast majority of most collections. Finally, it is thought that most of the real problems arise from a handful of aged viruses (old, that is, when compared to the age of the virus problem).

The *IAP* survey consisted of a simple questionnaire sent out to around 2,500 members (mainly in the UK) and 521 (circa 20%) were returned. I understand that this is a better than average response to such things. The figures which follow include approximate percentages, in order to give an idea of just where changes are occurring in this field.

### In the Wild

Of those replying, 280 (54%) reported no virus incidents. When asked how long ago the infection occurred, the remaining 241 were split 166 to 75 (69% to 31%) - the larger group indicating infection within the past year.

The survey then went on to determine which types of virus had been noted. Here, 81 (34%) definitely identified boot sector viruses only, 56 (23%) said parasitic viruses only, 41 (17%) experienced both types, and the remaining 63 (26%) did not know what type of virus had infected their computer. There were eight different boot sector viruses and 14

different parasitic varieties reported, so even if the 63 people who were unsure of the type all had different viruses (extremely unlikely), well under 100 different viruses would have been reported at large. This seems to confirm the current suggestion of approximately 40 to 45 common viruses causing almost all real-world problems.

A further breakdown of the virus types indicated that just five viruses accounted for around 93% of all boot sector infections (Form 38%, New Zealand 31%, Michelangelo 9%, Tequila 8%, Spanish Telecom 8%) whilst another four viruses caused around 65% of parasitic infections (Cascade 26%, Jerusalem 17%, Yankee Doodle 11%, Dark Avenger 11%). Thus the overall picture shows that of the 234 people who were able to identify the virus, 188 (80%) had been hit by one of just nine viruses.

This again tallies with most observed data from other sources, and is a far cry from the threat of 'thousands of viruses' which some vendors claim are in the wild.

*"It would seem from this that an anti-virus policy alone is no real defence against the threat."*

**Changing Times**

The survey revealed some interesting variations on the point at which infections were noted, and additional analysis was made of this. The most common virus reported from more than one year ago was Tequila (31 instances) followed by Cascade (14 reports), New Zealand (11) and Form (10). Since there were 100 reports within this time frame, these figures also represent percentages. The results for the past year show dramatic changes. The most common virus now is Form with 41 reports (21%), followed by New Zealand with 31 (16%) and Spanish Telecom with 11 (6%).

As well as obtaining these figures for actual virus infections, users were also asked how those affected had dealt with the problem. The response showed that over 82% had used proprietary anti-virus software, while around 14% had dealt with the problem in-house. Just 3% had contacted an outside consultant for further help.

Another series of questions asked how users handled the threat of virus infection. Rather surprisingly, 41% had an anti-virus policy and had been hit, 41% had *no policy* and had been hit, 13% had no policy and had not been hit, and the remaining 5% had an anti-virus policy and had not been hit. It would seem from this that an anti-virus policy alone is no real defence against the threat. The type of anti-virus measures which users implement were analysed as follows: 10% banned incoming software, 25% had some form of quarantine arrangement, 30% maintained control of software sources and 27% conducted regular software audits.

**Helping with Enquiries**

A final question concerned the reporting of virus attacks. This contained the biggest surprise - fewer than 6% of the respondents actually reported the incident to the police!

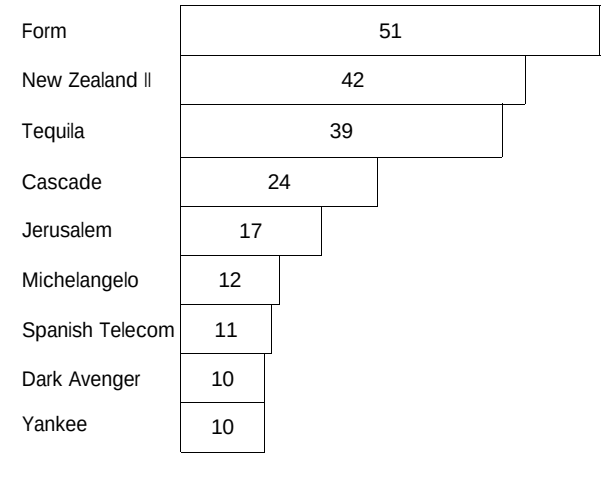
These figures certainly confirm that a virus problem does exist, since nearly half of all respondents had experienced an attack. However, the extent of the problem indicates that the level of user awareness, at least in the UK, has contained the problem within far narrower limits than those suggested by many vendors of anti-virus software.

All the viruses reported are relatively simple ones; there is a distinct absence of the more exotic types beloved of the academic researchers and virus collectors (Commander Bomber, Starship, DIR II, Tremor and so on). It seems that the presence or absence of an anti-virus policy has little effect in preventing infections. This can only be due to poor implementation and user education: a well designed virus defecne will prevent infection.

I was most disappointed to read just how few people report the problem to the police, as this has been a major source of statistical information on virus prevalence for some time now. However small their sample may have been, its usefulness is amply demonstrated by the similarity of the *IAP* survey. I would urge all users to reconsider any policy which prevents reporting virus outbreaks.

Each report is treated in the strictest confidence and provides the only possibility of bringing the perpetrators to book. If you need further information, call the *Computer Crime Unit* at *New Scotland Yard* on +44 (0)71 230 1177.

I am particularly indebted to Michael Ryan, Director General of *The Institution of Analysts and Programmers* (+44 (0)81 567 2118), for allowing me access to these figures and analyses.




**UK's 'Most unwanted' list:** The top nine viruses account for 80% of all virus outbreaks among those polled.

# PRODUCT REVIEW 1

## The ASP Integrity Toolkit

*Mark Hamilton*

The *ASP Integrity Toolkit* was first reviewed in *VB* by Dr Keith Jackson in June 1992. It was distributed by a Danish company, *Sikkerheds Radgiverne* (*SR*), who acquired sole worldwide distribution rights in January 1993. I was therefore particularly interested to see whether the product had since improved.

### The Package and Documentation

The product claims to provide an 'Integrity Shell' within which only validated programs can be executed. The *Integrity Toolkit* offers the user access control, file check-summing and verification, boot sector protection and a choice of two virus scanners (see below) to ensure that the system is virus-free before installation. This list is by no means exhaustive: the product attempts to provide a comprehensive solution to computer viruses in one package.

*Integrity Toolkit* consists of two manuals and one high-density 3.5-inch diskette. Unfortunately, not all computers can accept this media type, a point made in our previous review of the product, and one apparently ignored by the vendors. Also included was a letter, some of whose contents concerned me: 'I must stress that the installation process described in the manual must be followed to the letter.' Why is this so vital?

The manuals appear little changed since the author, Dr Fred Cohen, first penned them in 1991. One is an A5 book over 90 pages long, entitled simply, '*The ASP Integrity Toolkit*': it serves as user documentation. The other, a slim A4 booklet, contains details on how a technically proficient user might tailor the *Integrity Toolkit* to meet his needs. It is much more technical in content, and clearly not designed as a light read.

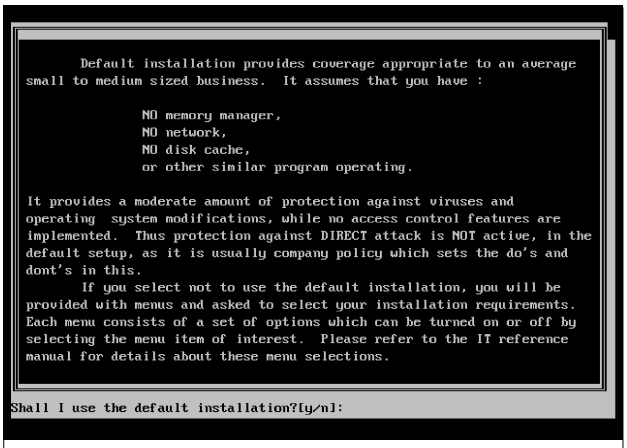
Whilst the product's underlying kernel is written in a mixture of assembly language and C, a LISP interpreter is used to configure the *Integrity Toolkit's* operation. LISP is not a popular language - certainly, in the computer departments of large corporates, skills in *Visual Basic*, *C*, *FoxPro* and the *Windows* and *OS/2 APIs* are far more common.

### Are we Compatible?

One of the caveats mentioned in the very brief installation instructions refers to setting up on a PC where a memory manager is running. Almost all PCs have some sort of memory management software; these users are referred to the System Administrator's manual, which appears later in the A5 book. Finding this section was not easy: the index and the table of contents are particularly unhelpful.

The manual stresses that the *Integrity Toolkit* should work with most memory managers, but does indicate a potential problem - if the user is running a memory manager, he is warned that installing BootLock may fail, causing the PC to lock up. The documentation states: 'If default installation fails, there is a chance you will have to use the recovery techniques listed earlier to regain access to your system.'

The BootLock component of the software actually encrypts the Partition Table - this is not a viable option for users with dual or multiple boot machines using the Boot Managers that come with *OS/2* or *Windows NT*, as it negates access under a non MS-DOS operating system. These changes take place without an explicit warning to the user, which could cause a few worrying moments. The documentation should be altered to explain this process more thoroughly.




If default installation is not used, the user must refer to the System Administrator's manual.

### Installation

Keeping in mind the warnings about compatibility, I began the installation procedure, which opens with a request for a registration number, an expiration date and a registration code. When I received the package, a note of the registration number was not included; I merely pressed carriage return and entered the registration code and expiry dates - a decision I would later regret (see below).

Menu Integrity Tool (MIT) is the program used to install the product. When executed, this proceeded to scan the hard drive with *F-Prot*, and install the components into the *ASP* directory created on drive C. This directory contained 148 files, and used almost a megabyte of disk space.

This initial scan is the only time *F-Prot* appears to be used in the *Integrity Toolkit*. I was informed by *SR* that it is not necessary to use this or any other scanner once the *Integrity Toolkit* is installed - a stance which, while factually accurate, does not reflect the way in which the product is likely to be used. If new programs are to be added to the hard drive, they

should be scanned before use. Unless one intends never to upgrade the software on the protected PC, a scanner *is* useful, though only for incoming disks.

When MIT had finished installing the program, I rebooted the PC, and the *Integrity Toolkit* immediately checked the boot sectors and executable files, comparing checksums to values stored in the database created on installation. It then displayed a 'Logged in' message. To accomplish this, it had modified my CONFIG.SYS file and, without alerting me first, had inserted the statement

```
SHELL=\ASP\ASPLOGIN.EXE \ASP\LOGARGS.ASP
```

This instruction means that the program ASPLOGIN.EXE program is executed before loading the command interpreter (usually COMMAND.COM). The original shell specification is stored in the file LOGARGS.ASP, so those machines which use a replacement COMMAND.COM should still function correctly with the *Integrity Toolkit* installed.

### The Menu Integrity Tool

The manual states that the product is normally used in one of two ways. The first of these lets the user implement features automatically installed at system bootup; the second gives more rights to System Administrators (primarily through use of the 'Menu Integrity Tool').

So far, in the words of the manual, I am a user. I decided to be the System Administrator of my own machine, which seemed to me a reasonable choice. I ran MIT, and was informed that the program had not been registered. As I had done before, I then typed carriage return to the registered number prompt once again, and re-entered the registration code and the expiry date. It had worked before, should it not work again? I was mistaken.

At that point a call to Denmark was necessary to obtain the necessary number. Once this had been done, I was able to gain access to the MIT program by providing the registration details required. In defence of *Sikkerheds Radgiverne*, installation is nearly always carried out by its own staff (indeed, there is a warning in the manual that this should be the case), so such problems should not occur. However, the unnecessary complexity would certainly put me off installing the product on new machines which I added to a system.

The *Integrity Tookit* provides protection by ensuring that only uninfected, authorised programs are allowed to execute. Each program is verified by checking its contents against a checksum. It was at this point that the vast number of options offered became confusing rather than useful.

The checksums can be either sequentially stored or hashed, the latter being faster but using more disk space. The choices range from Big/Hashed/Slow through Small/Hashed/Fast to Sequential/Trivial. The differences in the various storage methods are inadequately explained in the manual, which gives no suggestion as to which type would suit each individual user. On-screen help is also woefully lacking: the hint bar at the bottom of the screen, when selecting the option, tersely states 'No help available'. Indeed, I found reference to the previous *VB* review of this product, by Dr Keith Jackson, to be more useful than the manual provided by the manufacturer!

If none of the algorithms offered by the product are suitable for the user, it is also possible to nominate an external routine. The built-in checksumming methods should suit most users, although all the routines are proprietary, and conform neither with *ANSI* nor *ISO* standards.

### In Use

In its default configuration, all executable files are checked at boot time, along with key areas such as the boot sectors and the interrupt vector table. Using the Big/Hashed/Slow method, bootup time on the 25MHz 486sx notebook I used for testing this product was lengthened by ten seconds - not an unacceptable overhead.

Other overheads were similarly encouraging. Using even the slowest checksumming technique on offer, I noticed only a very slight increase in the time taken to load and execute programs. If an attempt is made to run a program not yet registered in its integrity database, the user is alerted and asked if its checksum should be determined and stored. If the answer is negative, the program is simply not run.

If the *Integrity Toolkit* detects alterations to the boot sector, the user is alerted - however, no disinfection is offered. This feature worked in my tests, although each time I had to run software from other vendors to disinfect my system before the *Integrity Toolkit* would allow me to boot from the fixed disk. An excellent result.

### Access all Areas

One of the many different features offered by *Integrity Toolkit* is access control. If the program is configured so that this is implemented, certain decisions must be made by the user. For example, three different types of access control are available; Two Type, POSet, and Milspec: adding the Two Type method (a two-password system; one with limited user rights, one with more access control, for the System Administrator) will lock the root directory of the computer. Locking the root directory, while providing a high level of security, prevents creating, editing or deleting any file or directory entry in the root.

This means that a user without Supervisory rights cannot install new software. This is an extremely useful prophylactic against a number of different IT threats, especially the use of pirated software and games.

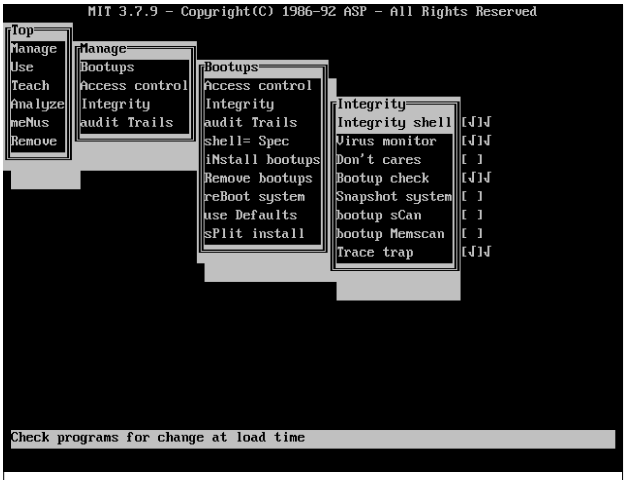
Due to the way in which PC access control is viewed, this option is likely to be used only in companies with their own dedicated IT departments. This is a shame, as there are many computing environments which could benefit from the features this option provides.

## The Scanner

As previously stated, two scanners are included with this product, Fridrik Skulason's *F-Prot*, and Fred Cohen's *SCAN*. The former is an integral part of the *Integrity Toolkit*, but used at no other time than installation, to pre-scan the hard drive. This, to my mind, is a serious error: when an attempt is made to execute a program not yet in the product's integrity database, that program should first be scanned for viruses. It would be advisable to pre-scan files before including them in the integrity database.

The *Integrity Toolkit* does not do this; consequently, it is possible to include virus-infected executables. As matters stand, it would be all too easy for the *Integrity Toolkit* to be targeted by a virus which would delete its database. For this reason, the use of a good scanner which can check incoming executables is vital.

The effects of executing infected executables can be (and are) detected, but the daunting task of identifying which file is the cause of the problem still remains - the manual does not tell the user that there is a more than adequate scanner included (*F-Prot*).




*Integrity Toolkit* is one of the most option-rich programs I have ever come across. However, the developers need to spend more time explaining the pros and cons of each choice for this to be of use.

Fred Cohen has written and documented a scanner for the product, called *SCAN*: it appears, however, to be fearfully out of date (the latest file date being December 1992), and only claims to be able to detect 'common viruses' (whatever those are). When run against the current *Virus Bulletin* In the Wild test-set, *SCAN* identified fewer than 30 out of the 126 samples as infected. Obviously *SCAN* is no longer being developed, and should be dropped from the product.

When referring to *SCAN*, the manual does affirm that 'the *Monitor* mode of operation is far more effective and less expensive than the *SCAN* mode of operation'. Unfortunately, it is nowhere explained either what *Monitor* is, nor how it might operate. It is possible that this is somehow a cryptic reference to *F-Prot*. If so, the manual needs to be updated to make this clear.

Version *2.09f* of *F-Prot* (issued September 1993) was that supplied with the *Integrity Toolkit*: it found all viruses in both the *Virus Bulletin* In The Wild and Standard test-sets. It scanned 1,393 files (58.0 Megabytes) in 78 seconds, in secure scanning mode, and on the test machine approximately 761 Kbytes/sec.

## Conclusion

This is an immensely complex product, and a complete review of its many features would fill far more space than the three pages available. It must be said that in its current incarnation, the *Integrity Toolkit* is not user-friendly. This should change when *SR* releases its new version, with CUA-compliant character-mode user interface and (one hopes) context-sensitive on-line help. As it stands, MIT does not support a mouse, and on-line help is simply a one-liner at the bottom of the screen - occasionally, even this states, tersely, 'No help available'.

Quibbles aside, the product's integrity shell is excellent and will detect executable file modifications, but users should be aware that there are still a number of programs which quite legitimately modify themselves: these cause problems with all such generic checksumming programs.

I have spent a great deal of time thinking about how to conclude this review. *ASP Integrity Toolkit* works and will, without doubt, provide an excellent way of managing a reasonably-sized IT system. However, the presentation of the package needs to be improved, and the compatibility issues solved. In its current form, the problems seem to outweigh the benefits: by design *Integrity Toolkit* is very restrictive.

*Sikkerheds Radgiverne* informs me that the product is being completely revamped, the documentation simplified and some of the more esoteric functions removed. If this is done successfully, there is no doubt that the product will be much improved, and certainly worth considering for sites whose PCs require a high level of protection.

One final note - when this product was last reviewed, some eighteen months ago, the quoted unit price was $89.00: although this has now increased by some 300%, the product itself has barely changed.

**Technical Details**

**Product:** *ASP Integrity Toolkit*

**Version Evaluated:** 3.7.9

**Vendor:** *Sikkerheds Radgiverne*, Knabrostraede 20, Copenhagen, DK-1210. Tel: +45 3332 3537 Fax: +45 3332 3547

**Serial Number:** None visible.

**Unit Price:** Dk Kr 1,895.00 (Circa UK £190 or US $290)

**Hardware Used:** *SIR 486* Sub-Note with 110 Megabyte hard drive and 4 Mb RAM, a 25 MHz 486sx processor and a single high-density floppy disk drive.

For details of the test-sets used here, refer to:

[1] Standard test-set: *VB* May 1992, page 23

[2] 'In the Wild' test-set: *VB* January 1993, page 12

# PRODUCT REVIEW 2

## Discovering PC-cillin

*Dr Keith Jackson*

*Virus Bulletin* last examined *PC-cillin* from *Trend Micro Devices* over two years ago. In that last review, Mark Hamilton was less than enamoured of the product, and dourly concluded that 'there is little, if anything, about this product to commend it'. Have things improved?

### Baubles, Bangles and Boxes...

*PC-cillin* is supplied as an 'Immunizer Box' (a small piece of hardware with 25-way D-type sockets on either end), an A5 manual, various pieces of bumph, and both 3.5-inch and 5.25-inch floppy disks. The Immunizer Box is mentioned neither in the Installation chapter nor in the index of the manual; the README file is also silent on the matter. I had to dig around elsewhere in the manual to learn that it should be attached to a parallel port.

This information is vital: PC hardware design is such that serial ports, with male sockets, and parallel ports, with female sockets, both use 25-way D-type connectors. Thus, it is possible to insert the Immunizer Box incorrectly into an RS-232 serial port. Given the higher voltages used by RS-232 signals, this may cause damage to the PC or to the Immunizer Box. As I value my test machine, I did not test the verity (or otherwise) of this!

### Documentation

Probably the biggest problems encountered with *PC-cillin* concern the manual, which appears to have been written with a different product in mind. It has not been revised for this version of the software, and even worse, no explanation has been added to the README file. The many flaws are doubly disappointing, as much of the discussion of anti-virus strategy is well written, and will make sense to most readers.

Several features touted in the manual, such as scanning files before execution, and disinfecting Mutation Engine infected files, are available from v4. However, the latest version of the software is v3.65, and the manual shows pictures of screens taken from v3.3 and v3.6. What has happened to v4? Why reference future versions? The documentationis confusing and confused, is virtually bereft of technical detail, describes 'Real Soon Now' features, and resorts to meaningless marketing nomenclature. In short, it is a mess.

The manual is prone to using silly names, and to depicting viruses with drawings resembling ink-blots [*or are they ink-blots which resemble viruses? Ed.*]. Disinfection of Mutation Engine infected files is called 'Mutie Clean' (on which the developers claim trademark), and the characteristics of their scanner are denoted by the phrase 'Deep Scan'.

I would argue that some of the claims made in the documentation are not fulfilled: *PC-cillin* purports to be the only product to disinfect MtE-infected files. Apart from the fact that this is patently untrue (several products can do this), the version of *PC-cillin* reviewed cannot even *detect* MtE-infected files, let alone disinfect them.

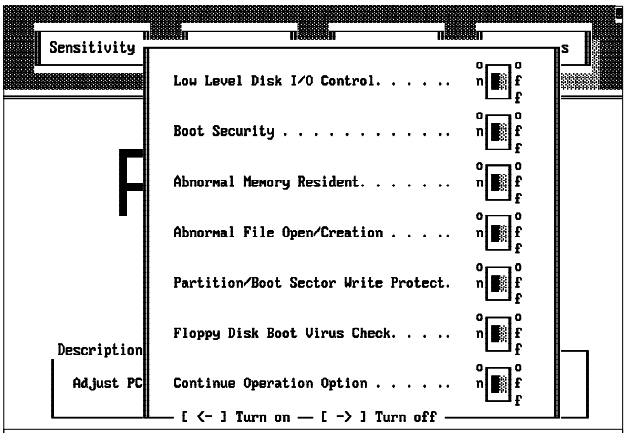
### Installation

Once the Immunizer Box was correctly installed, the manual instructed me to type PCCILLIN. This produced a 'Bad command or file name' error, as no executable file of this name existed on the floppy disk. Through a process of elimination, I eventually deduced that a file called PCC started installation. It is totally unacceptable that the name of the installation program given in the manual is incorrect.

Having completed the testing, I noticed in the extra bumph that users are advised to boot from the installation floppy disk to install. When I tried this, the naming problems described above were circumvented. This is not mentioned in the manual. The README file, which provides late additions to the manual, instructs users to boot from the floppy if upgrading, but mentions nothing about installation.

I am uncertain whether providing a boot disk for installation is a good idea or not. It requires the developer of the boot disk to solve all the problems of hardware compatibility normally tackled by *Microsoft* and the various OEM developers. What happens to compressed drives? The manual deals with none of these issues.

Finally, after booting from the installation disk, the user is asked to enter the name of the drive on which *PC-cillin* is to be installed. There is, however, no choice about the subdirectory. This is poor - it is my hard drive, and I should be allowed to put the files where I want to.




*PC-cillin* provides a number of different options in order to give enhanced protection on machines - however, lack of attention to detail lets the product down.

During installation, *PC-cillin* scans for known viruses, first in memory, then on the hard disk. Next, the boot sector and partition table information is stored in the Immunizer Box. I later found that *PC-cillin* had added lines to the start of the AUTOEXEC.BAT file, which installed a memory-resident program called Virus Trap. This occupied 14.2 Kbytes of RAM, a reasonably-sized chunk. For reasons which are beyond me, the installation program changed the date/time stamp on the CONFIG.SYS file, even though nothing in this file was altered.

The manual states that Virus Trap can be disabled after installation by removing the line which calls a program entitled PCCSTSR from the file AUTOEXEC.BAT: no such line exists. It also states that the AUTOEXEC.BAT file will be backed up, either to AUTOEXEC.$$$, or to AUTOEXEC.@@@. In fact, it is backed up to AUTOEXEC.PCC. After installation onto the hard disk is complete, the manual states that *PC-cillin* requires 110 Kbytes of disk space, and the README file says 360 Kbytes are required. Neither is correct. It occupied 449 Kbytes.

I encountered other problems during installation: for example, the manual states that the installation program will ask for a floppy disk as a Rescue Disk to 'store a copy of your hard disk partition table'. It did not. The entire installation procedure hardly inspires confidence in the product.

### Modus Operandi

With the product installed, my PC now has a small, single character, 'smiling face' (their phrase) blinking in the top right corner of the screen, which I personally find very irritating. Good anti-virus software should be completely unobtrusive. It is impossible to choose which character is displayed, but the feature can (thank goodness) be disabled.

In addition to a memory-resident anti-virus program, there is a scanner, called 'Quarantine'. When executed, *PC-cillin* scans memory before the first file scan is invoked. This takes 55 seconds. During this process, a counter zooms up to 562 Kbytes, and then clocks up very slowly to 640 Kbytes. I am not sure if this means that only the top part of memory is being scanned; such details are not in the documentation.

By default, scanning (I refuse to call it Quarantine) inspects all executable files, but this selection can be overridden. The scanner seems to accept only a single DOS wild-card expression; therefore it is possible to scan for all COM files, or for all EXE files, but not for both. If the scanner is executed from the command line, more than one file expression can then be named, but this merely invokes the scanning process twice. Not what is actually needed, I fear.

*PC-cillin* scanned the hard disk of my *Toshiba 3100SX* (see the *Technical Details* section) in 3 minutes, 1 second when scanning all files, 1 minute, 34 seconds when scanning all EXE files, and 47 seconds when scanning all COM files. In comparison, *Dr Solomon's Anti-Virus Toolkit* scanned the same hard disk in 39 seconds. *Sophos' Sweep* took 1 minute, 35 seconds in Quick mode (6 minutes, 7 seconds in full mode). The scanning speed offered by *PC-cillin* is not unreasonable, but it is by no means one of the fastest products, as claimed in the manual.

The scanner has some annoying foibles. It is possible to interrupt execution whilst scanning files on a disk, but not to interrupt the initial scan of memory. The Volume Name of the hard disk being scanned is always omitted from the appropriate field of the Report File, but curiously the Volume Serial Number is included. The scanner also insists that the Report File is written to the disk being scanned, a tactic as incomprehensible as it is annoying.

### The Viruses

I tested *PC-cillin* against the virus test samples listed in the Technical Details section below. The software claims to detect 1467 unique viruses, but the manual says there are 2600 'known viruses and strains as of June 1993'. Of the non-Mutation Engine samples, *PC-cillin* correctly detected all but five. None of the 1024 MtE samples were detected. Careful inspection of the manual discloses that disinfection of MtE samples is promised with version 4 of *PC-cillin* (remember that this review covers v3.65): perhaps MtE detection will also be included at this time. The first chapter of the manual states that '*Trend's* approach to virus protection is not compromised by the existence of today's mutation (polymorphic) viruses'. This courageous claim is wrecked by the 0% detection rate.

*"I have ploughed my way through more than 50 reviews for VB since its inception, and PC-cillin feels like a gigantic leap backwards."*

The five viruses not detected (Pitch, Power Pump, Todor, Tremor and WinVir_14) were all from the most recent addition to the test-set (a few months ago). Given that *PC-cillin* describes at some length that its scanner is 'rule-based' (their phrase), I surmise that each virus is analysed by the developers, in order to discover its method of operation, and the scanner amended as appropriate. Therefore, keeping up with the latest viruses is onerous and time-consuming.

*PC-cillin* always detected infection, but frequently (13% of the time) found a different virus from that actually present in the test sample. This may be a side-effect of using rules, rather than signatures, to detect viruses.

### The Virus Trap

The manual makes many claims about Virus Trap which, even allowing for features which will only be available from version 4 (see above), does not seem to work properly. The feature defined in the manual as 'Abnormal File Open/ Creation Detection' claims that it 'Warns of programs that

open themselves'. This is not true. Even with protection active, *Sidekick* could still edit its own executable file (SK.COM). Similarly, *Wordstar* could be used to edit WS.EXE, and my ancient address book program, which maintains names and address within its own executable image, could be updated *ad infinitum.*

Also, the feature described as 'Abnormal Memory Resident Program Detection' was happy for both *Sidekick* and *Manifest* to become memory-resident, although *Sidekick* is notorious for doing abnormal things to various interrupt vectors, and *Manifest* does a complete low-level examination of the system. What is abnormal? This adjective is never defined in the documentation.

Other features of Virus Trap include monitoring and inspection of the boot sector of both floppy and hard disks, and monitoring of 'Low Level I/O' (whatever that means: it is not explained). I tested the impact on performance imposed by Virus Trap by measuring the time taken to copy 35 smallish files (1.2 Mbytes). Without *PC-cillin*, these files could be copied under DOS in 24.4 seconds: with Virus Trap installed, it increased to 44.3 seconds. Under *Windows* these two figures were 25.0 seconds and 47.2 seconds respectively. Using either set of measurements, this corresponds to an imposed overhead of over 80%.

The scanner operated correctly under *Windows*, although it is only a program executing in a DOS box. All measured scanning times increase by about ten percent under *Windows*; a creditable performance. Virus Trap also works under *Windows*, though it needs a special program to be executed before it can make its error messages pop up.

## Problems in Reviewing

*PC-cillin* has been reviewed before by *VB* (July 1991), and the reviewer (not myself) had problems getting it to work properly. I have ploughed my way through more than 50 reviews for *VB* since its inception, and this feels like a gigantic leap backwards. First, the general standard of the documentation provided with anti-virus products over the past few years has improved dramatically. *PC-cillin's* documentation has not kept up.

Second, *PC-cillin* is dongled: although the only stated function of the 'Immunizer Box' is to store boot sector information securely, *PC-cillin* will not run without it. More sensible products include such features by writing files to floppy disk. *PC-cillin* could do the same, but chooses to use a dongle, and forces the user to attach this extra hardware to the parallel port (otherwise *PC-cillin* will not install, and Virus Trap will not execute). Still worse, unlike data stored on floppy disk, information held within the dongle cannot be securely backed up. What happens if the hardware fails?

The last review concluded that the dongle was unnecessary. I too see no reason for it, apart from the unstated purpose of copy protection. The developers seem to know this: they had a similar product called *PC Rx*, which was not dongled, reviewed by *VB* (October 1992, p.21). Re-reading the *PC Rx* review, the screens are very similar to those produced by *PC-cillin*, and the products seem to have much in common.

For testing purposes, I installed *PC-cillin* on two computers. If I cannot remember correctly on which computer the dongle was last used, what happens if I accidentally restore erroneous boot sector information? I could go on, but these questions make my point succinctly. The dongle adds no capability not achievable by 'normal' means, and can introduce problems ranging from a nuisance to something little short of a disaster.

## In Conclusion

Were the list of problems described in this review all fixed, I still would not recommend use of this product until the developers have publicly stated that their 'Immunizer Box' hardware has been ditched. In fact, had I known from the start that the 'Immunizer Box' was a dongle, I would have insisted that *VB* stick to its policy of refusing to review copy-protected software.

*PC-cillin* detects viruses well, but shows signs of being somewhat slower than other products at being updated; its detection problems were entirely with the most recently introduced virus test samples. The myriad problems with the documentation are explained earlier in the review. In my humble opinion there is no short-cut: the manual needs rewriting. Until this has been done, and *PC-cillin* has been de-dongled, I would not recommend its use under any circumstances whatsoever.

**Technical Details**

**Product:** *PC-cillin*

**Developer:** *Trend Micro Devices Inc.*, 2421 West 205th Street Suite D-100, Torrance, California 90501, USA, Tel. +1 (310) 782 8190, Fax. +1 (310) 328 5892.

**Availability:** An IBM PC, XT, AT, PS/2 or compatible with two floppy disk drives, or one floppy disk drive and a hard disk. At least 640 Kbytes of RAM and v3.0 or higher of MS-DOS are required. Even though not listed in the manual as a requirement, a parallel port is required for the dongle.

**Version evaluated:** 3.65

**Serial number:** 208628

**Price:** US $99.00

**Hardware used:** a) *Toshiba 3100SX* laptop, incorporating a 16 MHz 386 processor, 5 Mbytes of RAM, one 3.5-inch (1.4 Mbyte) floppy disk drive, a 120 Mbyte hard disk, running under *Toshiba DOS v5.0* and *Windows v3.1* b) 25MHz 486 clone, with 4 Mbytes of RAM, one 3.5-inch (720 Kbyte) floppy disk drive, one 5.25-inch (1.2 Mbyte) floppy disk drive, a 120 Mbyte hard disk, running under MS-DOS v5.0 and *Windows v3.1*.

Viruses used for testing purposes: this set of 143 unique viruses (according to the virus naming convention employed by *VB*), spread across 228 individual virus samples, is the current standard test set. A specific test is also made against 1024 viruses generated by the Mutation Engine (which are particularly difficult to detect with certainty).

For a complete listing of the viruses in the test-set used, see *Virus Bulletin* August 1993, page 19.

# CONFERENCE REPORT

## Predictable but Worthwhile

Early in November, *Virus Bulletin* fled the frosts of England for two days and went to Anaheim, California, where the *Computer Security Institute* was holding its 20th annual conference and exhibition.

Anaheim is part of the overall sprawl of southern Los Angeles, bristling with palm trees and theme parks. In the *Hilton and Towers*, opposite ageing Disneyland, several hundred delegates, speakers and exhibitors gathered for what has become the leading computer security conference in the United States.

### Agenda Details

The conference programme, with twelve simultaneous streams, was both ambitious and comprehensive, and covered a wealth of topics, including:

- Introduction to Computer Security
- The Next Step [*? Ed.*]
- LAN Security
- Management
- Awareness
- Open Systems
- Telecommunications
- Contingency Planning
- Micros and Portables
- Tools and Techniques
- Audit and Risk Assessment
- Product Specific

Only four of the 115 main conference sessions were concerned specifically with viruses. The first, from Noah Groth, of *PC Guardian*, gave an introduction to computer viruses, pointing out the importance of straightforward measures such as backups and employee awareness as aids in reducing the threat of a virus attack and limiting potential damage.

John Blackley, of *Guaranty Federal Bank*, shared his experience of creating and implementing a virus response team - this included everyday practicalities, such as choice of anti-virus software, methods of distribution, and ways of keeping it up to date.

Genevieve Burns, of *Monsanto Company*, gave an account of her strategy for developing a virus awareness campaign for a large company. Her talk covered both technical and business issues.

Finally, Dr Peter Lammer, of *Sophos*, gave a presentation on virus protection for PC LANs, in which he discussed technical aspects of virus spread and stealth behaviour in network environments, and explained the industry's move over the past 18 months to server-based scanning.

Other sessions, while not specifically virus-related, nevertheless addressed matters germane to the subject. Dan Erwin, of *Dow Chemical*, for example, gave a talk entitled 'Horror Stories and How to Use Them', applying a variety of management models to the problems of IT security.

Roger Thompson, from *Leprechaun Software*, discussing the pros and cons of anti-virus software with Hector Aguilar, of the *Deutsche Treuhandgesellschaft.*

### The Exhibitors

The anti-virus industry was represented slightly more strongly in the exhibition than in the conference; of a total of one hundred or so companies, those showing anti-virus products included *Command Software, Leprechaun, Reflex, Digital Equipment Corporation / Sophos, McAfee, Symantec* and *Trend.*

No major surprises were found here; life seems to continue much as usual. However, there appears to be more focus on server-based anti-virus software, with the apparently never-ending scanner race still the industry's bread and butter.

### Closing Thoughts

*CSI* is one of the main computer security events of the year, and for this reason alone it is worth attending. The conference itself is primarily an educational event rather than a research forum - this means that delegates who have attended before can expect a familiar programme.

This is not to say that the event was without entertainment: there was a very good cocktail bar, where copious discussion of data security issues took place each evening while Victoria Paoletti and Jerry Garvin made great music at the piano next door. Last year's *CSI* venue, the Chicago *Hilton and Towers*, was made famous this year in the film *The Fugitive*. It is up to Hollywood to decide whether *CSI's* latest venue will be afforded the same star treatment.

**SUBSCRIPTION RATES**

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

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

**Editorial enquiries, subscription enquiries, orders and payments:**

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

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

US subscriptions only:

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

Tel 203 431 8720, Fax 203 431 8165



# END NOTES AND NEWS

**Stuck for a last minute Christmas present?** The *Survivor's Guide to Computer Viruses* makes the perfect stocking filler. Both informative and highly readable, this one-shot reference book on computer viruses costs only £19.95. For further information, and for details of bulk purchase discounts, contact Victoria Lammer. Tel. +44 (0)235 555139.

*Central Point* has launched an *OS/2* version of *Central Point Anti-Virus*. The new product is designed to complement *Central Point's* recently updated *NetWare* product, providing centralised virus reporting over a network. *CPAV OS/2* costs £99 including four updates, and requires a 386 machine or higher, running *OS/2* 2.x. Tel. +44 (0)81 848 1414.

*Sneakers* type computer hacking is catching on in the United States, according to a report in *Computer Fraud and Security Bulletin*. Management companies such as *Price Waterhouse*, are being approached by clients to provide 'hacker-like' penetration services to see where the weak points to their system are. Set a thief to catch a thief, and all that…

**TSR Review Follow-up.** Commenting on *McAfee Associates'* poor performance in the recent TSR review (*VB*, September 1993 p.15), Phil Talsky, spokesman for *McAfee*, claimed that the performance of the TSR was 'not a problem' as long as users always used the scanner too. Many feel differently. David Merril, vice president of a Manhattan executive search firm, commented 'I'm supposed to feel good about that sort of protection? Who's writing anti-virus software - Beavis and Butthead?'

The *National Computer Security Association* has released its Fall catalogue, containing over 100 computer security-related items. Tel. +1 (717) 258 1816. Fax. +1 (717) 243 8642.

Yes! Dr Solomon will float! *S&S International's* buoyant Chairman has expressed his intention to take his company to a recognised stock market within two years. The company has recently been given two business excellence awards by *Commerce Business Magazine*.

An *International Symposium on Computer Crime* will be held in Beijing, China, on 25th-27th October 1994. For further information, contact Mr Jing Qian-Yuan. Tel. +86 (1) 5121667. Fax. +86 (1) 512 1667.

**Patricia Hoffman's *VSUM* ratings for October:** 1. *Command Software's F-PROT Professional* 2.09f, 95.5%, 2. *McAfee Associates Viruscan* V108, 95.0%, 3. *Sophos' Sweep* 2.53, 91.5%, 4. *Dr Solomon's AVTK* 6.55, 90.4%, 5. *Safetynet's VirusNet* 2.08a, 89.5%. **NLMS:** *McAfee NetShield* V108, 93.7%, 2. *Sophos Sweep NLM* 2.53, 91.6%, 3. *Dr Solomon's AVTK* NLM 6.54, 86.4%, 4. *Command Software's Net-Prot* 1.00s 69.2%, 5. *Cheyenne's Innoculan* 2.0/2.18g, 64.4%.

**Software piracy case lands perpetrators in prison.** According to a report in *Corporate Security Digest*, one man is in prison and another serving home detention after being convicted of manufacturing and distributing at least 25,000 copies of MS-DOS. It is believed that this is the first computer piracy case to result in a prison sentence.

The problem of *Novell NetWare* password creation has been solved by *Baseline Software's* latest product, *Password Genie*. One of the most common ways of breaking into a computer system is by guessing passwords. *Password Genie* alleviates this problem by making sure that all users employ difficult-to-guess passwords at all times. Each password must pass 43 different tests in order for it to be acceptable. The software costs $395 per server, and can be run on all versions of *NetWare* from v2.x. Tel. +1 (415) 332 7763.

*AT&T* has announced the launch of three programs designed to **enhance the security of data and communications**. The software provides encryption, authentication and secure data transmission. 'These programs offer key capabilities for anyone working on the road, from home, at remote sites or in a mobile office setting,' said Bill Franklin, business development manager for *AT&T Secure Communication System*s.