ISSN 0956-9979 SEPTEMBER 1994

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **KAOS reigns.** A virus has been released on the *Internet*: how great are the risks? For an analysis of the virus, see p.8; for an analysis of the risks, turn to p.6.

• **Comparatively speaking.** *VB* has always advised against the use of virus removal software. However, it has become an integral part of many anti-virus software packages. How effective is this technique? See p.11.

• **Fire, fire!** *Norman Data Defense Systems* has released a server-based anti-virus package: how does it compare to the DOS version of this software, *Norman Virus Control*? Product Review 2 has all the answers.

CONTENTS

EDITORIAL

To Detect or Not to Detect…

When is not a virus a virus... or, put more simply, are there occasions when a virus scanner should label a file as infected, when it is not? The answer to this question seems obvious: virus scanners should detect files which are infected by a fully-functioning virus. However, is this what one actually wants a 'real world' scanner to do?

Consider the following experiment. An innocuous file is encrypted using the Mutation Engine. The file is then virus-checked by a number of different packages. Each scanner will give one of three results. Firstly, the scanner might pass the file as clean. Secondly, the scanner might alert the user to the presence of the Mutation Engine code. Finally, the scanner may display a message informing the user that the file is infected with a virus. Clearly, the most useful statement is the second one. However, almost all scanners fall into either the first or the last category - the second statement, though factually accurate, is not a great deal of use to the user: can he run the file or not? The only way to find out is to analyse it, a task which will be beyond the scope of the casual browser.

The experiment can be made still more complicated. Imagine a polymorphic virus which sometimes does not carry out its infection process correctly, adding a decryption routine to 'infected' files, but failing to add the encrypted virus code. Such a file will not replicate, and is therefore not a virus. However, there is a powerful argument that the user should be alerted to its presence.

The current trend in the industry is a move towards precise or exact virus identification. In many ways, this is highly beneficial, as disinfection can be carried out more accurately, and the user is left with a better picture of what potential effects the virus may have had on his system. With the advent of widely available polymorphic engines, the only way to identify exactly which virus a file is infected with is to decrypt the file and examine the code 'hidden' by the encryption. If this code is identified as a virus, then the file is deemed to be infected. If it is not, the file is clean. This process is slow on infected files, but very quick on clean ones.

" Surely a user would like to know that a particular program is wrapped in an MtE decryption routine? "

All well and good... except when the program encounters a file which has a valid decryption routine attached to it, but random code encrypted within it. In such cases, some scanners will not label the file as infected. But is this action unnecessarily pedantic? Surely a user would like to know that a particular program is wrapped in an MtE decryption routine? This is critical in the case of the user who is attempting to clean up a system after an attack by a polymorphic virus. Here, the anti-virus software should identify all those files which contain the virus, or parts of it: that is, those files which have been altered.

Vendors will quite justifiably point out that change detection is exactly what a checksummer does. However, the use of checksummers is hardly widespread, and one feels that there is something unnecessarily pernickety about the failure of a scanner to alert the user to fragments of viruses left scattered across the disk. If files half-infected by a botched virus cannot be detected by the scanner, then it is time for a change of emphasis: the scanner and the checksummer should be combined in a way which is transparent to the user. The checksum information can then be used during a clean-up operation, to identify those executables which have been altered by the virus. It should be noted that during clean-up, whether an altered program is a virus or a dysfunctional attempt at infection is immaterial to the user - all that matters is getting the machine operational as quickly as possible. Some products already use this two-pronged attack, but few seem to make the marriage of the two techniques as harmonious as it could be.

Until the next generation of products is installed upon computers worldwide, the main line of defence against virus attack is the humble scanner. Here the question of 'to detect or not to detect' is still unanswered: some scanners can identify 'half-infected' files, and some cannot. So when is it acceptable to call something a virus when it is not? The current industry consensus on the matter is rather undecided, leaving those unlucky enough to be caught without a backup of their system blundering around in the dark. Anybody care for a light?

VIRUS BULLETIN ©1994 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)1235 555139. /94/$0.00+2.50

NEWS

Honecker: The Last Laugh

The latest computer virus story to hit newspaper headlines worldwide is the 'Honecker virus'. The virus triggers on 13 August (the anniversary of Erich Honecker's construction of the Berlin Wall), and displays a caricature of the late East German leader, complete with spectacles, on the screen.

This is followed by a rendition of the national anthem of the former German Democratic Republic, and a message announcing the destruction of programs 'by order of the Council of Ministers of the German Democratic Republic'. The next message reads: 'Honni's last revenge - I'll be back'. It then deletes the AUTOEXEC.BAT file.

Analysis of the 'virus' shows that the program should probably be regarded as a Trojan, as it is incapable of spreading onto floppy disk without actually being copied by the user. The virus is written in a high-level language, and creates a 52480-byte file called DOSINFO.EXE in several sub-directories of the fixed disk. The Trojan then adds code to the start of batch files on the disk to ensure that the program is executed. The program was distributed in an X-rated file, uploaded to German BBSs ▌

Prism Reaches Out

The *NCSA* (*National Computer Security Association*) has announced the launch of a new program, *Prism*. Services provided by the program include access to on-line help, telephone help-desk support, the 'Underground Research Laboratory', the Virus Research Centre, magazines and newsletters, and national seminars and conferences with internationally-recognised experts in attendance. Members will also be warned by Email of any potential virus attack, and have the use of the product information service.

The program is a logical solution to a resource problem: many corporate IT Managers suffer from an overload of unscreened information, and have to allocate considerable resources to filtering it. *Prism* is designed to carry out this filtering first, supplying information from a wide range of sources, without swamping the company in trivia.

Its member-elected Advisory Council helps to determine the program's direction, assist in establishing special interest groups, provide input to educational programs, and recommend new or amended member services.

Prism membership is offered to government and business organisations of all sizes through a multilevel pricing schedule, calculated according to the revenue of the company concerned. Membership starts at US$4,500.00. Further details are available from the *NCSA*, tel. +1 717 258 1816, fax +1 717 243 8642. The *NCSA* can also be reached on *CompuServe* as 75300,2557@compuserve.com ▌

Virus Prevalence Table - July 1994

Virus	Incidents	(%) Reports
Form	15	34.1%
Spanish_Telecom	6	13.6%
CMOS4	3	6.8%
Flip	2	4.5%
Green_Caterpillar	2	4.5%
Green_Caterpillar.B	2	4.5%
JackRipper	2	4.5%
Smeg.Pathogen	2	4.5%
Cascade	1	2.3%
Eddie_2	1	2.3%
Joshi	1	2.3%
New_Zealand	1	2.3%
New_Zealand.I	1	2.3%
NoInt	1	2.3%
Parity Boot	1	2.3%
Stoned.O	1	2.3%
Taiwan.2900.d	1	2.3%
V-Sign	1	2.3%
Total	44	100.0%

ARCV Case Wrapped Up

The case of *ARCV*, the UK-based virus-writing group *Association for Really Cruel Viruses*, has finally reached its conclusion (see *Virus Bulletin*, November 92, p.3). DC Noel Bonczoszek, at the time an officer of *New Scotland Yard's Computer Crime Unit* recently issued a statement saying that the President, Secretary and two couriers of the group had been identified, arrested, and were subsequently given a police caution. A fifth person was cautioned on another matter, while another arrested at the time was released with no further action taken. However, no victims of viruses written by *ARCV* were identified.

The statement goes on to thank the anti-virus community for their assistance. Commenting on the case, Bonczoszek (now attached to *Marylebone CID*) said, 'a potentially serious problem was nipped in the bud.'

The arrest and cautioning of members of *ARCV* is likely to be met with a mixed response from those in the IT industry. Although the virus-writing group was stopped in its tracks, the lack of convictions will be a source of irritation to some. Part of the reason for members of *ARCV* not being taken to court is believed to be the dearth of reports of their viruses in the wild, highlighting the need for those affected by viruses to report the attack to the appropriate authorities. The *CCU* can be contacted on Tel. 0171 230 1177 ▌

VIRUS BULLETIN ©1994 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)1235 555139. /94/$0.00+2.50

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 20 August 1994. Each entry consists of the virus name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or a dedicated scanner which contains a user-updatable pattern library.

Type Codes

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

ARCV.Christmas.678
CR: This virus appears to be based on the same source code as the 670-byte ARCV.Christmas virus. Detected with the Ice-9 pattern.

Barrotes.1310.F
CER: Detected with the Barrotes pattern, as is the 1310.G variant.

Bupt.1220.C
CER: Detected with the Bupt (Traveller) pattern.

Burger.441.B
CN: Detected with the Burger pattern and the Virdem and Virdem-fam patterns. The Virdem patterns also match the Burger.382.C virus.

Cascade.1701.T
CR: Detected with the Cascade(1) pattern, as are the 1701.U and 1701.V variants. The 1701.Q variant requires a new searchstring, as the decryption loop has been modified.

```
Cascade.1701.Q      018B D9EB 0446 4943 418D B74D 01BC 8206 8134 F066 464C 75F8
```

Chaos.I
CER: Detected with the Chaos (formerly Spyer) search pattern.

Chaos_Year
CER: An unremarkable 1837-byte virus.

```
Chaos_Year          3D00 4B75 06E8 F102 E97B 0080 FC3D 7506 E84D 04E9 7000 80FC
```

Chris
CR: This 463-byte virus contains the text '<CHRIS of S.i.t.>'.

```
Chris               80FC 4B74 052E FF2E FC01 061E 5557 5652 5153 5090 901E 5231
```

Chromo
CN: A 406-byte virus containing the text '[Chromosome Glitch] v1.0 Copyright (c) 1993 Memory Lapse'. This virus may perhaps be reclassified as a member of a family which contains several other viruses by the same author.

```
Chromo              CCC6 8699 0200 C686 9A02 00EB 00B4 4EB9 FF01 8D96 5602 CC3D
```

Cobra
CN: A 400-byte virus which prepends itself to the files it infects. It contains the text '~Cobra Cou~'.

```
Cobra               A19A 00A3 1B01 E80C 00B4 4FCD 213D 1200 7402 EBE0 C3B9 0500
```

Cybertech.552
CN: This virus uses variable encryption, and no searchstring is possible. The following text is present within the decrypted code: 'Mourners of a dying world. Too late to reconcile. Into Everlasting Fire. Can't you see it's Satan's world.'

Cybertech.1066
CN: Another encrypted variant, but the decryption loop is constant. There is also a 1228-byte variant by the same author.

```
Cybertech.1066      E800 005D 83ED 0750 8DBE 1B00 89FE B913 04AC 34?? AAE2 FA
Cybertech.1228      E800 005D 83ED 0750 8DBE 1B00 89FE B9B5 04AC 34?? AAE2 FA
```

Dicker
CR: A 400-byte virus which prepends itself to infected files.

```
Dicker              80FC 9075 03BB 9900 3D00 4B74 052E FF2E 3401 9C50 5351 5206
```

FeelBad
CN: This 1124-byte virus probably originated in the Netherlands. Its name derives from the text 'we feel bad about Ritzen'.

```
FeelBad             B840 008E D8BB 6C00 8A07 1F24 033C 0375 06BB 7804 E801 00C3
```

Fifo
CR: A 300-byte virus containing the text 'FIFO'.

```
Fifo                80FC 3674 03E9 D300 5053 5152 1E06 55B4 19CD 2150 FECA 7804
```

Filehider.1057
CR: Detected with the Filehider (789) pattern. Similar to a 1067-byte variant reported in July 1993.

Fission
CER: A 517-byte virus containing the text '[Binary Fission] v 1.0 [ML/PS]'. The text indicates that it is written by the same person who wrote Chromo.

```
Fission             3D00 3D74 283D 013D 7423 3D02 3D74 1E3D 0043 7419 3D01 4374
```

Flip **CER:** Three variants of Flip (2153.E, 2153.G and 2365) have not been mentioned before. Like other Flip viruses, they cannot be detected with a single, simple searchstring, but programs capable of detecting earlier Flip variants should also be able to detect these.

Flue **CN:** A variable-size virus (1179 bytes long or more) using variable encryption.

Friday_the_13th.416.C **CN:** An unremarkable minor variant of this virus, requiring a new searchstring.

```
Friday_13th.416.C   FF36 0201 FF36 0401 B43F B903 00BA 0201 CD21 725F A00E 0138
```

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

IVP **CN, CEN:** Three new IVP-generated viruses are known: 260 (CN), April (1676) and Mandela.943.

Jerusalem.Sunday.L **CER:** An unremarkable 1636-byte variant, detected with the Jeru-1735 pattern. The same pattern will also detect the new 2064-byte Jerusalem.Tarapa.C virus.

Keypress.1232. **CER:** Detected with the Keypress pattern.

Leprosy.Busted.572 **EN:** Yet another member of this family of primitive overwriting viruses.

```
Leprosy.Busted.572  8B0E 0C02 51E8 1000 5BB9 3C02 90BA 0001 B440 CD21 E801 00C3
```

Necropolis.C **CEN:** Very similar to the other two known variants; detected with the Necropolis (1963) pattern.

PS-MPC The appearance of the following PS-MPC viruses should not be a surprise to anyone: 339.F (CN), 347.K (CN), 352.M (CN), 574.E (CEN), 578.H (CEN), Alien.733 (CER), ARCV-4.742 (CEN), Asstral (EN, 753), G2.Mudshark.312 (CN), Joshua.964 (CEN), Shiny.934 (CN), Sucker (CR, 572), Tester (CN, 302).

Trivial **CN:** There is a constant trickle of new small overwriting viruses which do nothing but replicate. Due to their small size, the patterns are shorter than normal, and should be used with care.

```
Trivial.25.B        BA9E 00CD 212A 2E2A 00B7 4087 D193 EBF3
Trivial.29.B        21BA 9E00 B802 3DCD 2193 5AB4 40CD 21C3
Trivial.30.G        218B D8B4 40B1 1EBA 0001 CD21 2A2E 2A00
Trivial.33          2193 BA00 01B4 40CD 21C3 2A2E 434F 4D00
Trivial.37          0001 CD21 B43E CD21 B44F EBE4 2A2E 2A00
Trivial.38.B        BA00 01B9 2600 CD21 CD20 2A2E 636F 6D00
Trivial.39.B        B440 CD21 B43E CD21 B44F EBE2 2A2E 2A00
Trivial.42.F        21B4 3ECD 21B4 4FEB E2CD 202A 2E63 2A00
Trivial.42.G        CD21 B43E CD21 B44F EBE1 2A2E 636F 6D00
Trivial.43.B        B43E CD21 B44F CD21 73E4 C32A 2E63 2A00
Trivial.43.C        B92B 00BA 0001 CD21 CD20 2A2E 636F 6D00
Trivial.45.E        7473 7920 7275 6C65 7321 202A 2E43 2A00
Trivial.54          EBE0 2E2E 00B4 3B5A BA28 01CD 2173 CBC3
```

Troi.E **CR:** Almost identical to the C variant. Detected with the Troi pattern.

VCL **CN, PN:** Several VCL-generated viruses have appeared recently: 609, Beepop (PN, 587), Bigtime (676), Butthole (overwriting, 493), Dumbco (3808), Genesis (741), Gif (696), Renegade (5737) and Westward (657). Most are encrypted, and should be detected as other VCL viruses: Westward is not, and is detected with the VCL.VoCo pattern.

Vienna.648.Oscar **CN:** Three 648-byte variants have been found recently, all of which contain the text '(C) OSCAR'. Variants A and C are detected with the interceptor pattern, but B requires a new pattern.

```
Vienna.648.Oscar.B  B903 008B D690 83C2 0DCD 218B 5406 8B4C 0483 E1E0 83C9 1D90
```

Vienna.778 **CN:** Detected with the Dr_Q pattern.

Vienna.Violator.707.B **CN:** Detected with the Violator pattern.

Vienna.Violator.5286.B **CN:** Detected with the Xmas_Viol pattern.

Xph.1010 **CER:** Similar to the two variants reported earlier.

```
Xph.1010            3D00 4B74 0580 FC3D 7553 2EC6 060C 0401 8BFA 4774 4280 3D00
```

YB.316 **CN:** This virus is also known as Silent Runner, as it contains the text 'Silent Runner by Nostradamus [NuKE'94]'. It is 316 bytes long, and has not been fully analysed.

```
YB.316              B802 3DCD 2193 B905 008D 9408 01B4 3FCD 2172 218B 842B 0105
```

YB.466 **CN:** This virus contains the text 'YB-1 & Handsome Dick Manitoba / Köhntark', indicating that it is by the same author as the KAOS4 virus.

```
YB.466              B802 3DCD 2172 2F93 B905 008D 9494 01B4 3FCD 2172 218B 84C1
```

YB.647 **CN:** A related virus, containing the text 'YB-2 / Köhntark'.

```
YB.647              B802 3D9C FF9C 6801 72E3 93B9 0500 8D94 5F01 B43F 9CFF 9C68
```

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.