ISSN 0956-9979

JULY 1995

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

Editor: **Ian Whalley**

Assistant Editor: **Megan Palfrey**

Technical Editor: **Jakub Kaminski**

Consulting Editors:

Richard Ford, NCSA, USA
Edward Wilding, Network Security, UK

IN THIS ISSUE:

- **Scanning the results.** This edition sees the second major DOS scanner review of 1995. How did the competition measure up? See pp.14-20 for some surprises.

- **Little.Red - brought to book!** Former *VB* editor Richard Ford (now at the *NCSA* in Pennsylvania, USA) writes from a different perspective. His virus analysis begins on p.12.

- **Stang on life.** Self-avowed 'turtle lunatic' David Stang (no longer at the *NCSA* in Pennsylvania, USA) gives his views on a lifetime of service. Turn to p.6 for an Insight.

CONTENTS

VIRUS BULLETIN **©1995 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)1235 555139. /95/$0.00+2.50** No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

EDITORIAL

Revenge of the Trojans

In recent weeks, we have seen the appearance of two new Trojan horses, only one of which I have been able to confirm. The second, although unsubstantiated, has been mentioned several times in messages on various *Compu$erve* fora.

In the field of what is sometimes known as ‘malware’, Trojans are perhaps the simplest creatures. Designed to cause some form of damage to your data, they can be produced with little effort and less skill. This fact is attested to by the first of the Trojans, which is a dummy version of *PKZIP* (the almost universally-used compression utility from *PKWare*), and is to be found in files called PKZ300B.EXE. The second Trojan reported masquerades as version 2.22 of *McAfee’s VirusScan*, and is said to have very similar effects to the *PKZIP* Trojan - specifically, deleting files and directories on the drive being scanned.

> *“ it is easier to write a Trojan than to write a virus ”*

Needless to say, if you come across a file called PKZ300B.EXE, avoid it like the plague. If the file is run, it unpacks itself into five more files, one of which is called PKZINST.EXE, and is a Trojan. When run, it executes two DOS commands: ‘format C: > NULL’, followed by ‘deltree /Y C: > NULL’. Anyone with experience of DOS can see the flaws. First, to throw away the output of a command, the device to which it must be redirected is NUL, not NULL. Second, the format command waits for the user to type ‘Y’, then return, before actually formatting the disk.

If the user wonders, as is often the case, why nothing appears to be happening, and presses return a few times in an attempt to prod the program into life, the format command will fail, and execution will proceed to the deltree, which will work. The command’s output will be redirected to a file called NULL in the current directory, but at this point the user will not be overly concerned with such trivia; more with retrieving his missing files.

Earlier in this editorial, I used the word ‘plague’, which perhaps is not the best term to use in the context of Trojan horses. Plague implies infection and contagion: a Trojan horse can carry out neither of these processes unaided. The mechanism by which such a program spreads is, of course, the area in which it differs from a virus: whereas a virus works to spread itself, a Trojan has no built-in means of spreading; for that, it relies on human intervention. The only area which requires a degree of creativity is deciding how to encourage users to get hold of and use the beastie.

The way in which such a program spreads varies from Trojan to Trojan. In the case of the two mentioned above, users are encouraged to spread the programs by the fact that they appear to be new versions of utilities which the users already have.

It is a mysterious urge of the computer user always to have the latest version of something, be it a simple utility like *List* (a freeware file viewer) or *TDE* (*Thomson-Davis Editor*, a freeware text editor), or a vastly complex multi-component application such as *Microsoft Word*. Even if that user is perfectly happy with his current version of the software, it is seemingly impossible to resist the compulsion to replace it with a new version. It is upon this that the Trojan relies to seduce people to download it, and then to use it.

The other technique is that of making the Trojan appear to be a new package which tempts the user to run it. Two categories are commonly used: something pretty (e.g. a new screensaver), and something useful (e.g. a software utility to change your 386 to a 486…).

These two incidents serve to reinforce the usual advice - only get updates and new software from reliable, secure sources; and treat all incoming software with suitable suspicion. The appearance of new Trojans such as these is a cause for concern. Even the best scanner is no defence against a Trojan (or, indeed, a virus) which it has not seen before. The increased connectivity in the electronic world today makes the passing of such programs all the easier, which in turn makes it all the more likely that your organisation will be exposed to one.

NEWS

Cruncher ‘In the Wild’ in Russia

Anti-virus researcher Eugene Kaspersky reports from Russia that the Cruncher virus, one of the most difficult-to-detect and complex infectors around at the moment, has been found ‘in the wild’ in Russia.

The incident began at a bank in Siberia. On 8 June 1995, when the bank’s *NetWare* servers started operations for the day, workstation monitors displayed the message ‘DIVIDE OVERFLOW’, and then locked.

On analysis of system files, technicians discovered a change in the length of an EXE file: in fact, its length had decreased from 55KB to 44KB. The lengths of the *NetWare* files LOGIN.EXE, MAP.EXE and LOGOUT.EXE had been more than halved, and the end of each contained the string:

```
  *** CRUNCHER
```

IT staff at the bank worked until late into the night to assess the extent of the damage, searching each and every workstation for the virus’ signature. By midnight they had found more than one hundred infected files; however, as Cruncher uses self-compression, not all files were found. The next morning, the virus was in the bank’s network once again.

The bank spent considerable time and effort in disassembling the virus, writing a TSR vaccine, and looking for reliable anti-virus software.

Cruncher reached the bank via what has become an almost clichéd route: an employee downloaded a DOC (*MS Word*) file from a local ftp site. The file ostensibly contained an ‘automatic compression utility’, a phrase which almost perfectly describes Cruncher. The employee then executed the file on his PC, and logged in to the network… and the circle began ▮

VB 95

The *Fifth Annual Virus Bulletin Conference* will be held at the *Park Plaza Hotel* in Boston, Massachusetts, USA, from 20-22 September 1995. This will be the first time this highly successful gathering will have been held in the US.

Experts will address a wide range of issues, including the susceptibility of *Windows NT* and *Windows 95* to virus infection, viruses on the *Internet*, and virus control in a corporate environment. The two-and-a-half day conference will consist of technical as well as non-technical streams, and will also feature an exhibition by security soft- and hardware vendors.

The fee for the event is £595 (US$895), less a £50 discount for *VB* subscribers. Information is available from Conference Manager Petra Duffield on Tel +44 1235 555139 ▮

Virus Prevalence Table - May 1995

Virus	Incidents	(%) Reports
Form	32	18.3%
Parity_Boot	21	12.0%
AntiCMOS	20	11.4%
AntiEXE.A	15	8.6%
Monkey.B	10	5.7%
JackRipper	9	5.1%
Junkie	8	4.6%
NYB	7	4.0%
Monkey.A	6	3.4%
Sampo	5	2.9%
Telefonica.A	5	2.9%
Viresc	4	2.3%
BUPT	3	1.7%
DA_Boys	2	1.1%
Natas	2	1.1%
Neuroquila	2	1.1%
She_Has	2	1.1%
Tequila	2	1.1%
V-Sign	2	1.1%
Other *	18	10.3%
Total	175	100%

* The Prevalence Table includes one report of each of the following viruses: Angelina, Arara-1375, Arianna-3375, Cascade, Die_Hard.2, EXE_Bug.A, Goldbug, Keypress, Leandro, LZR, Markt, NED-09, NoInt, RPS2, Screaming_Fist, Stealth_Boot, Tai-pan, Vacsina.

Ready… Steady… 95!

Anti-virus software vendors are readying themselves for the imminent release of the new operating system, *Windows 95*. Both *ESaSS* and *Symantec* are currently testing Beta versions of anti-virus products - *ThunderBYTE* and *Norton Anti-Virus* respectively. *McAfee Associates* states that it will have a *Windows 95*-specific anti-virus product shipping within 60 days of the release of the operating system.

In addition, *Symantec* is beta-testing a version of the widely used add-on, *Norton Utilities*, for the new operating system, which should provide many similar features to the DOS version of the best-selling package.

It is assumed that other vendors also are working on new versions of their products. Indeed, an unexpected side-effect of the repeated postponements to the release date of *Windows 95* is that more manufacturers should have products ready either before or shortly after the OS first ships ▮

IBM PC VIRUSES (UPDATE)

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

Type Codes

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

3APA3A.C

RD: Historically the first variant of the kernel infector from Russia; also found in the wild. The signature for detection is slightly different in infected files [*see VB, November 1994, pp.9-10*].

```
3APA3A.C         0EE8 0000 5E83 EE04 5650 5351 521E 06B4 04CD 1A80 FE06 7208
```

Antipode 2

CR: Encrypted, appending, 1007-byte new variant of Antipode virus with the hidden text: ‘TBDRIVER COMcom TBSCAN.EXE PROT.EXE [Antipode 2]’.

```
Antipode 2       E802 00EB 14BE 3000 03F2 8BFE 81EF 4001 B9BF 0331 3C46 E2FB
```

Bane

ER: Encrypted 256-byte virus with stealth capabilities. It inserts its code in an unused space in file headers. Contains the text ‘[Bane]’.

```
Bane             500E 0E1F 07BF 2400 E804 00E9 1600 ??8A 260E 00BE 2400 B9DC
```

CodeJournal

CER: Polymorphic virus about 4700 bytes long. Uses the ViCE v0.5 polymorphic engine (‘V’ is the FBH ASCII code), and contains the text: ‘CodeJournal by Virogen [NuKE]’ (V==FBH). The template given detects the virus in memory.

```
CodeJournal      9C3D AB63 7504 33F6 9DCF 2E80 3E72 0401 743E 2EC6 0672 0401
```

DarkKiller.693

CEN: Based on the PS-MPC structure, this is an encrypted, 693-byte direct, fast infector, which contains the text: ‘This is <DK> Virus Written By Dark Killer’.

```
DK.693           BE?? ??B9 4D01 CD12 2E81 34?? ??83 C603 4E83 F600 BFFF FFAA
```

DarkKiller2.269

CN: A 269-byte, prepending, direct, fast infector. It contains the plain-text message: ‘I’m <DK2>, Written By Dark Killer’. The only encrypted code is the original part of an infected program.

```
DK2.269          BB0D 028B 0EB0 01F6 1743 E2FB BE0D 02BF 0001 578B 0EB2 01F3
```

DK.Clouds.588

CN: A 588-byte, direct, fast infector. It contains the plain-text message: ‘This is [Clouds] Virus, Written By Dark Killer’.

```
DK.Clouds.588    B43F B904 008D 96EC 02CD 213E 81BE EC02 CCE9 7506 8D86 AA01
```

DK.Clouds.657

CN: A 657-byte, direct, fast infector, containing the plain-text message: ‘This is [Clouds II] Virus, Written By Dark Killer’.

```
DK.Clouds.657    B43F B904 008D 9624 03CD 213E 81BE 2403 90E9 7506 8D86 C501
```

DK.Clouds.718

CN: A 718-byte, direct, fast infector. It contains the plain-text message: ‘This is [Clouds v3.1] Virus, Written By Dark Killer’.

```
DK.Clouds.718    B43F B904 008D 9628 03CC 3E81 BE28 0390 E975 03EB 4290 B802
```

Dos-Idle

CER: An encrypted 692-byte virus containing the text: ‘[DOS Idle]’.

```
Dos-Idle         E80F 0044 49B9 4A01 2E81 35?? ??47 47E2 F7C3 5D81 ED03 001E
```

Emmie.2823.B

CR: Minor polymorphic variant, encrypted, with stealth capabilities, 2823 bytes long. Contains the messages: ‘It’ll tire you too much’, ‘My name is Emmie, I am Eddie’s sister’. Same as signature for Emmie.2823.A, with the exception that two bytes have been swapped.

```
Emmie.2803.B     B8?? ??E8 0000 FC5D 8D76 0EB9 0205 3104 4646 EB00
```

Eternity

EN: Encrypted, appending, 565-byte, direct infector containing the text: ‘[ETERNITY!] (c) ’93 The Unforgiven/Immortal Riot’.

```
Eternity.565     E800 005D 83ED 03E8 1500 EB27 90E8 0F00 B440 B935 028B D5CD
```

Five_u

CR: 314-byte appending virus. Installs itself in the interrupt vector table. It marks infected files with the signature ‘5u’ at offset 3. All infected programs end with the plain-text message: ‘5uThe G World Fuck’.

```
Five_u           2E88 6504 0E1F 57C3 3D75 3575 04B8 4444 CF80 FC4B 7403 E984
```

Hydra_II.B

ER: A polymorphic, 1665-byte virus, which contains the message: ‘This is Hydra v1.1. Don’t panic, I will not destroy your data’. The template below detects it in memory.

```
Hydra_II.B          7516 81FB 7310 7510 81F9 DAFE 750A BB59 48B9 5244 B800 01CF
```

Jerusalem.Rulis

CER: Encrypted, 1639-byte variant with the string ‘Rulis’ attached at the end of infected files. It does not infect COMMAND.COM. It can be found in memory with the Barcelona template.

```
Jerusalem.Rulis     B4FF CD21 3D52 4F75 12B4 EEFC BF00 01BE 6206 2E8B 8D12 0003
```

Jerusalem.2000

CER: Another variant, this time about 2000 bytes long. This also does not infect COMMAND.COM.

```
Jerusalem.2000      FCB4 B0CD 2180 FCB0 7316 80FC 0B72 11B4 B1BE D007 BF00 0103
```

Morbid

CR: A 461-byte encrypted virus which contains the message: ‘The MORBID(OS) virus V2.00’.

```
Morbid.461          8A07 32C2 8807 438D 8690 023B D875 F1C3 E8E3 FFB4 40BB 0500
```

Neither

CN: A 591-byte virus which prepends part of its code and appends the rest. Contains the plain-text messages: ‘I love you P, always will’, ‘neither here, nor there’. The time stamp of infected files is set to the illegal value FAF0 Hex, which DOS translates to 7:23pm (19:23).

```
Neither             5AB9 0A01 B440 CD21 B945 0129 CA89 D6B4 40CD 2152 33C9 8A0E
```

Page.1221

CER: Polymorphic, 1221-byte virus containing the text: ‘[NUKE’95] by pAgE!’. It is identical to the VLAD virus [*see VB, June 1995, p.5*] apart from the signature of a different author.

Rainbow

CERDM: Multi-partite virus, six sectors long, with stealth capabilities. The length of infected files increases by 2351 bytes. The virus contains the plain-text message: ‘HiAnMiT - roy g biv’. A system with an infected MBS cannot be booted from a clean system floppy if the machine is running any DOS version of 5.0 or higher.

```
Rainbow (boot)      BB00 7C8E D38B E38E C3B8 0502 B9?? ??BA ???? CD13 9AA5 8300
Rainbow (file)      E800 005E 83EE 03B8 AD1B CD13 3DED DE75 450E 1F81 C664 0781
```

Ratboy

CN: A 269-byte direct infector which infects one file at a time. Contains plain-text message: ‘RATBOY’.

```
Ratboy              B440 CD21 B802 4233 D233 C9CD 21B4 40B9 0D01 8D96 0401 CD21
```

Ratboy.Ihater

CN: A 513-byte encrypted direct infector with a destructive payload. Triggers on Sundays in the second half of the month. Contains the text: ‘RaTBoY HaTeS YoU!!!’ and ‘Ihater-u-all’.

```
Ratboy.Ihater       E803 00EB 2890 3E8B 8651 018D B653 01B9 DA00 3104 4646 E2FA
```

Ratboy.Killer

CN: A 545-byte encrypted direct infector which targets anti-virus software. It contains the message: ‘My Vx, Invircible Killer’.

```
Ratboy.Killer       E803 00EB 2890 3E8B 8641 018D B643 01B9 F200 3104 4646 E2FA
```

Ratboy.Love

CN: A 306-byte encrypted direct infector which infects three files at a time. It contains the message: ‘To My Wife, Love Ratboy’.

```
Ratboy.Love         E801 00C3 3E8B 860C 018D B63F 01B9 7C00 3104 4646 E2FA C38A
```

Ratboy.Mrs

CEN: A 671-byte encrypted direct infector which infects COM files and corrupts EXE files by overwriting the first 3243 bytes. The normally-encrypted message can be seen in all destroyed EXE programs: ‘RaTBoY Loves Mrs. RatBoY!!!’.

```
Ratboy.Mrs          E803 00EB 2890 3E8B 8651 018D B653 01B9 2901 3104 4646 E2FA
```

Ratboy.OW.A

CN: A trivial 50-byte virus. Overwrites first file in current directory, and contains the text: ‘OW-Ratboy’.

```
Ratboy.OW.A         B44E B900 00BA 2301 CD21 B802 3DBA 9E00 CD21 93B4 40B9 3200
```

Ratboy.OW.B

CN: A trivial 80-byte virus which overwrites one file at a time.

```
Ratboy.OW.B         93B4 3FB9 0400 BA50 01CD 21B4 3ECD 2180 3E53 0172 7504 B44F
```

Replicator

ER: A 651-byte virus which infects files in a similar manner to a direct infector after changing drive or directory. Contains the text ‘[Replicator]’.

```
Replicator          E800 005D 81ED 0300 1E06 B804 63CD 213B C374 518C C048 8ED8
```

Sarampo

CER: A 1371-byte long virus which triggers on 25 April, 12 October, and 25 December; overwriting the video memory with random code. It displays this message, which can be seen in the virus’ body: ‘Do you like this Screen Saver? I hope so. Created by Sarampo virus’.

```
Sarampo             BA6F 03B8 2125 CD21 B42A CD21 81FA 1904 740F 81FA 190C 7409
```

Trivial.OW.26

CEN: A simple 26-byte, overwriting, direct infector which targets the first file in the current directory. The new length of an infected file is always set to 158 bytes.

```
Trivial.OW.26       2A2E 2A00 32C9 B44E 8BD1 CD21 BA9E 00B4 3CB7 40CD 2193 87CA
```

U-life

EN: Encrypted; 1455 bytes long. Based on the Ear virus, it is a fast direct infector, and contains a destructive payload. One of about a dozen messages which can be displayed at random reads: ‘Wait for the departure of you life’.

```
U-life              8B96 9801 8E9E 9A01 CD21 5B0E 1FC3 8DB6 BB01 8134 ???? 4646
```

INSIGHT

David Stang: Turtles, Trains and Trojan Horses

David Stang, by his own admission an adventurer, began his professional life in computer security with *NCSA* (US-based *National Computer Security Association*), which he created. The organisation started out of his basement, and he called himself Director of Research - that, he felt, would imply that other people were working there. In fact, for the first year or so he was the only employee, losing money 'hand over fist'.

Stang and the *NCSA*

'In the very beginning,' he admitted, 'I knew nothing about viruses - no-one did. One of the things we did was answer the phone: "if you've got a problem, call; we'll take care of you". "We'll" meant "I will". So we announced the number, and the phone rang: somebody had a question about a virus.'

Indeed, most early queries concerned viruses. At the time, Stang's 'network' consisted of one computer, which made him extremely careful when he worked on a virus. He was the *NCSA's* resident virus expert; explaining viruses, helping people get rid of them, offering suggestions as to which anti-virus product would be the best in a given situation, providing contacts for further assistance, etc.

'In those days, we didn't charge - our attitude was that you might appreciate the information and want to join *NCSA*. Membership was US$45 a year. We got up to about 1000 members, which wasn't making us rich. I did virus seminars to help subsidise the income.'

Moving On

Once the success of the *NCSA* was assured, and it was earning enough money to support him, Stang turned the running of the organisation over to two friends, Bob Bales and Paul Gates, and formed another group, called *ICSA* (*International Computer Security Association*).

'The *NCSA* was a good model; it seemed to do a good service for people. So I thought, let's start *NCSAs* in other countries.' Stang began with Thailand, Singapore, and Malaysia: it worked like a franchise, apart from the fact that Stang was not paid for the licences.

'Basically, I was going broke, so we published a journal called *Virus News and Reviews* which, in the US, was meant to compete with *Virus Bulletin*. After twelve issues, it died. Sylvia [*Moon, still a colleague, now at Norman Data Defense Systems with Stang*] and I did most of the stories, all layout and shipment, all subscription management - we must have had at least half a dozen subscribers!'

Defending the Users

Moon and Stang were then contacted by the Norwegian company *Arcen* (later to be known as *Norman Data Defense Systems*), which produced anti-virus software: 'They called us up and asked if we would like to run their North American operation,' reminisced Stang.

'We thought about it for ten or fifteen seconds, and said "Sure". We wanted to stay in this line of work, doing viruses; we wanted to be a vendor, but to serve the users. My goal was to be able to tell the truth. [*sic. Ed*]

'There's a part of everyone who wants to do some good in life. If you're going to be a computer geek, it's hard to come up with moral justification, and in the world of fighting viruses, there is a morality - not a holier-than-thou kind of morality, but the smile on somebody's face when you clean their machine and don't destroy everything, and they can sit down and do their work, and everything's back to normal. I like that - I like users, and solving their problems. And I like the sense that I'm doing something good.'

> *"in 1989 it was possible that a scanner detected all known viruses, but today that's unimaginable"*

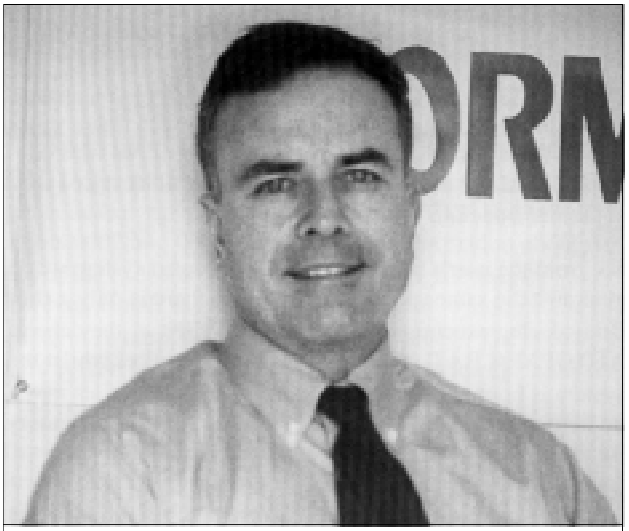
Since his first contact with *Norman*, Stang has worked closely with the company - even before the official agreement was signed, he helped to find them new partners in the form of a company in Malaysia. Today, *Norman's* programming partners include experts in Australia (*Robust*), the Netherlands (*ESaSS*), and the US (*Countermeasures* and *Communications Arts & Sciences*).

'We put the user first: we use the best code we can find and integrate it. Every product has to have a local look and feel, so we have manuals written here in America, but in Norway they are edited and produced by Norwegians. In Germany, the Germans touch them; and in Malaysia the same thing applies. Everybody makes it look local - that's important.'

A Personal View

Stang revels in discovering people in other countries who think alike. He has been in the business longer than most: 'I'll be fifty in August,' he commented, 'and that makes me one of the oldest living anti-virus guys who's still at it.

'I don't think there's anything harder than a job like mine - I'm supposed to be a manager and president, a techie, a translator from technical to practical. I have to keep pace



Stang started independently in computer security with the *NCSA*. He landed eventually at *Norman Data Defense Systems*, bringing his experience and knowledge to the vendor side.

with technology; not to mention the hundreds of virus authors who don't publish their work on a quarterly basis. To know every virus a caller might ask about, and to know enough that I am useful - that's hard. If nobody else can do it, Lord, let me try. I'm having fun with the challenge.'

Stang does not see himself as a virus expert - he regards himself as having become known in the anti-virus industry purely through historical accident: 'The amount of stuff you could know to really merit the title "expert" is unfathomable,' he said. 'The amount I know is such a small fraction of that amount, so when I compare what I know with what I wish I knew, I feel very humble. And I really wish that our industry could somehow be more effective at producing greater amounts of expertise.

'If you a look at virus authors and vendors, and compare their numbers,' he continued, 'you discover there are more virus authors than vendors; that virus authors share knowledge with each other and vendors generally do not. The result is that, from the user's standpoint, things have got worse. In 1989 it was possible for a scanner to detect all known viruses - today it's unimaginable. Vendors have been losing ground. The worldwide growth in infections is proof.'

Who is the Expert?

Stang believes that, despite the fact that there are many organisations which give virus support, and many people who style themselves experts, there are very few real experts about; people to whom a person can go with a specific question and receive a specific answer.

The anti-virus industry, said Stang, has concentric layers. There exists, in his view, a large layer of expertise beyond the well-known names, which might include the technical support person in a large organisation - that layer is vast, and very uneven. Many of these people have huge collections, and are very knowledgeable; but there are others who are not: 'They get on the *Internet* and swap mail, and you can read the content of their messages - it's clear they are not experts, and it's also clear that they think they are.'

Outside the Profession

Stang does have interests outside viruses and work: 'Turtles. I like turtles. I'm a turtle lunatic. In fact, I suppose I like tortoises - we Americans are a bit sloppy about what is a turtle, and what is a tortoise. And I like reptiles and dinosaurs. At work, we have three tortoises - I've learned a lot about myself through watching them. Tortoises are patient and persistent creatures, qualities which I admire.

'I like tortoises. And I like model trains. We should do a survey - many *Virus Bulletin* readers had model trains as children [*choo-choo. Ed.*]; more than would be expected by statistics. They're different that way - even now many of them still have model trains.'

Stang feels that, with model trains, there is understanding, prediction, and control - things which people in the computing world appreciate.

'Think of the virus,' he said. 'You run your machine, you know what happens when you hit Return. The virus changes that. Under this circumstance, this happens, and under this circumstance this other thing happens. Once you understand this virus, you can predict what happens, and you regain control. So we are again masters, but this time of a more complex beast.

'There's another thing, which is the fascination with near-life forms. Model trains are alive - see, look, this one's going around this corner, he's going through that tunnel, he's coming over here, he's making this turn, he's going this way, he's slowing down, he has a light. It must have a heart, and there's a caboose, with people and stuff in it. There is this lifelike thing to model trains…'

The Right Way

Stang enjoys his work - in America, he said, successful people work hard; and they work hard because they enjoy it: 'You can't put in an 80-hour week unless you find some pleasure in it. I like to program, and I like to do seminars. I sometimes like to write, but I don't like ever re-telling a story. I like creating knowledge when I write.'

Despite the fact that he is an extrovert in public, he sees his true nature as shy - he would rather program and be alone than be with people: 'That's where I began life. The rest was cultivated. I've wound up with a wonderful bunch of work partners - the Norwegians and the Malaysians, Dutch and Australians; and of course the folks from my own office - they are what I have wanted to build our company to be.'

He recalls the time that Alan Solomon [*of S&S International*] asked him whom he would serve: 'Forcing me to answer that helped me to make the right decision. Users are first in my book; I mean to take care of them as best I can.'

VIRUS ANALYSIS 1

EM: System BootUp Virus

Eugene Kaspersky
KAMI Associates

Sun, sea, summer beaches… and computer viruses. Have they anything in common? Unfortunately, yes. Computers can now be found in every place where there is electricity and room for the processor box and keyboard.

After a long, tiring journey to a favourite holiday hotel, the discovery that its reception is shut because of a computer virus could not be a pleasant one. Such an incident happened in Sochy, a well-known resort town on the Russian Black Sea (150 km of beaches…). A sample of EM, the virus concerned, was recently sent to me from there. It may well be that this virus spoilt someone's summer holiday.

Inside the Virus

EM is 1303 bytes long, and encrypted with a simple XOR instruction which uses a randomly-selected key. It utilises i386 instructions: 32-bit data access is very handy while searching for the four-byte text strings 'path' and 'PATH'. The virus uses the double-word register (EAX) when scanning the data for PATH= lines.

The virus has two forms: either a 1303-byte file called EM.COM (whence the name derives) which is a COM file containing the virus, or an EXE file appender. No text is visible in the virus code, but several strings, used by the virus during creation of the COM file and while searching for EXE files, appear after decryption:

```
path
PATH
em.com c:\ autoexec.bat c:\*.* *.exe
```

EM receives control when an infected file is executed. The decryption loop restores the virus code to its original form, and passes control to one of two infection routines. The first modifies AUTOEXEC.BAT and creates EM.COM, and the second seeks EXE files to infect. After the chosen infection routine is performed, the virus passes control back to the host program (when executed from an infected EXE file) or to DOS (when executed from EM.COM).

If the virus is executed from EM.COM, it passes control to the routine which searches for and infects EXE files; if it executes from an infected EXE file, the virus creates the file EM.COM. To detect the type of the host file (EM.COM or an EXE file), the virus uses a flag, which it sets/resets during EXE file infection or during creation of the COM file.

The infected EXE file, when executed on a PC, will only create the file EM.COM, and does not search for other files. When DOS processes AUTOEXEC.BAT at load-time, EM.COM is executed. This searches for any EXE files on the C: drive and infects them, although not all at once (see EXE File Infection below).

So, from the user's point of view, the infected system works at almost the same speed as a clean one: there is only a short delay during execution of infected EXE files, as they check AUTOEXEC.BAT. The virus also hits other files whilst the system is booting, so the delay to find and infect files may be written off as a 'lazy' driver installing itself into memory.

AUTOEXEC.BAT as Target

During execution of an infected EXE file, the virus opens the file AUTOEXEC.BAT in the root directory of drive C, reads its contents into memory, and scans it for the line containing the string 'PATH' or 'path'. If found, the virus searches for the end of that line, and checks the next line for the string 'em'.

If it contains only this string, the virus exits the infection routine and passes control to the host EXE file. If the string is not found, the virus moves the rest of AUTOEXEC.BAT four bytes down, and inserts the string 'em', followed by the bytes 0D0Ah (which correspond to a carriage return and a line feed) into the gap. The relevant lines in the file AUTOEXEC.BAT before and after alteration are as follows:

```
before alteration       after alteration
@echo off               @echo off
…                       …
PATH=C\DOS;C:\…         PATH=C\DOS;C:\…
PROMPT=…                em
…                       PROMPT=…
win                     …
                        win
```

EM does not modify AUTOEXEC.BAT intentionally if there are no 'PATH' or 'path' strings; neither does it alter such a file correctly - it leaves four random bytes at the end of that file. Where this is the case, the virus cannot infect other files on the computer. Thus, one way to protect a PC against this particular virus is to change the PATH string in AUTOEXEC.BAT to 'Path', 'patH', or something similar.

After modifying AUTOEXEC.BAT, the virus encrypts itself and writes its code to the file EM.COM in the root directory of drive C. The system is now infected: when next rebooted, EM.COM will be executed, and EXE files on the C: drive will begin to be infected.

EXE File Infection

During system bootup, the virus, in the form of EM.COM, is executed from the file AUTOEXEC.BAT. The virus code is the same as that of infected EXE files, but another execution path is followed. First, the virus checks the system date. On the 28th of any month it calls the trigger routine. On other days, it scans the directory tree of drive C, infecting not more than ten EXE files. The virus will do this on each reboot, until all EXE files on drive C are infected.

EM's technique for detecting already-infected EXE files is quite unusual: there are two words in the EXE header, at offsets 0002 and 0004, which contain information about the size of the executable module (DOS EXE file length, however, equals the length of the executable module and the length of overlay data, plus relocation tables). The second word (offset 0004 in the EXE header) contains the number of 512-byte pages occupied by the executable module.

The first word (offset 0002) contains the number of bytes in the last page. The value of the first word must be within the limits 0000-01FFh; however, the EXE file executes with no apparent problem if that field contains data above 01FFh. I only tested this under *MS-DOS*, so cannot tell whether it would apply to other operating systems.

The virus uses this first word in which to save its ID-marker. During infection, the virus stores the file's date stamp in the EXE header at offset 0002. While searching for files, the virus compares the word with the file's date stamp, and does not infect if they are equal.

> *“the infected EXE file … will only create the file EM.COM, and does not search for other files”*

The other parts of the EXE infection routine are quite standard: it reads the file header; increases file length, making it a multiple of 16 bytes (the size of a paragraph); encrypts and writes itself at the end of the file; and modifies and saves the EXE header with new initial register values.

EM does not check overlay data in EXE files. As a result, it corrupts files with an overlay block (as do Jerusalem viruses), as well as the new format EXE files (as used by *Windows* and *OS/2*). This is a gross error, especially in the year of *Windows 95*! The virus does not perform such standard viral tricks as Int 24h hooking, but there is a reason for this - the virus only infects files on the C: drive; no other drives are accessed during searching and infection.

Trigger: Bye-bye Files!

As has already been seen, the 28th day of any month is the *dies iræ* for owners of infected computers. This is when the virus calls the trigger routine, and brings a mountain of unexpected work for the rest of the day - data restoration.

The virus scans the subdirectory tree of the C: drive using absolute disk access calls (Int 25h), collects the addresses of subdirectories, and then corrupts entry names. It overwrites the first character of the names of all objects found (files, directory names, volume labels) with the space character. As a result, file and directory names have a space at the beginning, and cannot be accessed using a standard DOS command. Despite this, data is not corrupted. DOS will not find such files, because it does not support the space character in file names, but the DIR command displays the fact that all the files are there, albeit with ‘shortened’ names.

There is a way to rescue files without using special disk editing utilities, using ‘features’ of the *MS-DOS* command RENAME (this may not work under systems other than *MS-DOS*). If we create the file A.A and type the command:

```
RENAME A.A A?B?C?D.E?F
```

the result is a file with a very strange name, A B C D.E F, with spaces between the letters. Using this ‘trick’, it is possible to restore files with corrupted names. The command

```
RENAME ?ilename.dat filename.dat
```

is used to restore the name to its original form. Such a command would not be able to restore directory names: here, it is necessary to use a disk editor utility.

Conclusion

It is perhaps unlikely that EM will spread far in the wild, given the infrequency of its trigger. It does, however, display an interesting technique in replication, attempting to hide within the sometimes lengthy system boot process.

EM

Aliases: None known.

Type: Encrypted, non-polymorphic file infector.

Infection: EXE files: parasitic, modifies AUTOEXEC.BAT, creates the file EM.COM. The virus will infect files only on the C: drive.

Self-recognition in AUTOEXEC.BAT:

Virus checks line following the ‘PATH=...’ line for the ‘em’ string.

Self-recognition in EXE Files:

Compares file's date stamp with EXE header field at offset 0002 (number of bytes in last 512-byte page of program).

Hex Pattern in Files:

```
8CC8 8CD3 8BD4 8ED0 BCFE FF53
522D ???? 50BB ???? 538C DA8C
```

Intercepts: None.

Trigger: Corrupts system sectors containing file directory entries.

Removal: Under clean system conditions, identify and replace infected files. Delete the line ‘EM’ in the file AUTOEXEC.BAT.

VIRUS ANALYSIS 2

Sticky: What’s in a Name?

Mike Lambert
Frontier Corporation

Perhaps the most interesting thing about the virus under analysis this month is that it was found ‘in the wild’ earlier this year in Midwest USA. Its one confusing element is its relationship to the Screaming_Fist clan: this virus is linked by name to the Screaming_Fist/EMF/Enemy family (file infectors) despite the fact that it is Tequila-like, primitive, encrypted and multi-partite.

Screaming_Fist or Sticky?

While the virus is not on the current *WildList* (a regularly-updated list of viruses known to be in the wild, produced by Joe Wells), it is mentioned as common in *VSUM*. This leads one to believe that the recent sighting of this virus may not be the only one in the wild. Although *WildList* infections are verified by experts, the exact strain of many virus infections goes unverified.

F-Prot, *FindVirus*, *AVAST!*, and *SCAN* call this virus ‘Screaming_Fist.927’ (*SCAN* used to call it ‘Multi’). *AVP* uses ‘Fist.927’ (consistent with calling all Screaming_Fist/EMF/Enemy viruses ‘Fist.xxx’). *TBAV* and *AVSCAN* call it ‘Screaming_Fist_Nu_Way’. Patricia Hoffman, of *VSUM*, lists it as Sticky, and states that it is also known as Multi2. Given the gulf between viruses in the Screaming_Fist family and this infector, I shall opt for the name Sticky.

It seems that the Screaming_Fist, EMF, and Enemy viruses are all related by the text ‘Screaming Fist’, which is found in most viruses in the family. The infectors are, for the most part, functionally equivalent; usually resident COM and EXE infectors: for instance, 404, 625, 683, and 711 are Screaming_Fist; 692, 696, and 838 are Screaming_Fist.II. The ones which vary from this pattern are variants 404 and 625, which are non-resident COM infectors.

The viruses appear to have been created in the second half of 1991, and made their first appearance in the wild during 1992. All, with the exception of the variant known as Stranger, contain the text ‘Screaming Fist’.

Unique Characteristics

As stated above, Sticky itself does not resemble the Screaming_Fist series: Sticky is primitive, encrypted, and multi-partite. This is not intended to be a dissertation on all Screaming_Fist variants; there are more, albeit lesser-known, variants than those mentioned above - one of these may yield a connection. Suffice it to say that disinfecting a system infected with Sticky differs from the ‘standard’ method of disinfection used for Screaming_Fist infectors.

Sticky seems to post-date the Tequila virus by six months to a year, and is functionally equivalent to it, without the ‘bells and whistles’. In other words, when an infected file is executed on an uninfected system, Sticky acts as a dropper. When the ‘dropped-on’ system is later rebooted, the virus has become a COM/EXE infector.

A description of the virus’ life cycle shows quite clearly its similarity to Tequila:

- an infected COM or EXE file is executed on a clean system, and the virus drops into the Master Boot Sector (MBS) of the hard disk, storing the remainder of its code in sectors 2-4, Track 0, Head 0
- the infected system is later booted, and Sticky goes resident, ready to infect COM and EXE files
- the virus infects COM or EXE files when they are opened, executed, renamed or when a file attribute is changed (unless the program name is SCAN). Virulence is moderate. Virus scanners not finding the virus active in memory will cause an infection disaster (see below)
- an infected COM or EXE file is taken to another system and executed

Execution of Infected Files

The COM/EXE execution is very simple. After the virus is decrypted, it issues an ‘Are you there?’ call (Int 21h, function FEEFh). If Sticky is resident, it will answer with the value 1234h in the AX register, in which case the host program will be repaired and executed.

> *“any decent behaviour blocker which denies write access to the MBS will block the virus when infected files are executed”*

If the virus is not already resident, it reads the MBS and checks for part of its loader (18 bytes, which is the code to load the virus from sectors 2 and 3 into memory). If the MBS is not infected, it is copied to sector 4. The original MBS is patched with 44 bytes of virus loader code and rewritten to sector 1, Track 0, Head 0. The body of the virus is then written to sectors 2 and 3, following which the host program is repaired and executed.

Int 13h is used to drop the virus on the hard disk. There is no aggressive code in the virus to penetrate Int 13h; thus any decent behaviour blocker which denies write access to the MBS will block the virus when infected files are executed. If the MBS write returns an error, the infection attempt aborts.

Basic generic detection techniques are more than sufficient to find the virus' presence and identify infection attempts after the virus is loaded.

On Booting

Later, when the system is booted, the patched MBS loads the virus at the top of memory, just under the 640K limit. Standard memory protection ('subtracting memory' from available memory at 0000:0413h) is used to protect the 3K the virus needs. CHKDSK will show the lower amount of memory available. The original MBS (stored in sector 4) is loaded at 0000:7C00h and, after interrupts 13h and 1Ch are hooked, the virus transfers control there.

Interrupt 13h is only used for MBS read stealth. When a request is made to read the MBS, the virus inserts sector 4 in CX and lets the ROM code read the real MBS. The whole Int 13h handler is a mere 21 bytes long.

The simple method of using the Timer Interrupt (1Ch) to 'wait for DOS to be loaded' was an innovation when it first appeared in 1991. When the virus' Int 1Ch handler sees that DOS is loaded (Int 21h is modified), it gets the real Int 21h vector, patches its own Int 21h handler in the Interrupt Vector Table, and restores the original Timer interrupt. Tequila, which is about 3000 bytes long, did this, and much more - but Sticky is only 927 bytes long.

File Infection

Now loaded with DOS, Sticky will infect COM and EXE files on the commands Open, Exec, Change file mode, and Rename. It will append itself to a file using the standard COM and EXE infection techniques, but will not infect files named SCAN.

In COM files, the 'infected flag' is the fourth byte (which will equal the value of the second byte minus 1). In EXE files, it is the initial IP (which is set to 1). File date and time are preserved, and the R/O file attribute is overridden. A COM file must be between 300 and 62000 bytes long to be a suitable candidate for infection. 906 bytes of the virus are simply (XOR) encrypted in COM and EXE files, but not encrypted in sectors 2 and 3 on the hard disk. There is no text to read in the virus - would one not expect a 'Screaming Fist' somewhere?

There is no file stealth to hide the virus infection in the directory or file, nor is there a trigger routine. The virus uses its own Int 24h handler for error handling. Sticky merely replicates, avoiding (as stated above) any COM or EXE file called SCAN.

Conclusion

Since the virus infects on Open, a scanner which does not detect Sticky in memory will spread this virus everywhere. It is disappointing to see this happen when scanners belonging in this category are run on an infected system.

If network technical support personnel were to scan a server (many techies have write access to server files) with the scanner whilst logged on with an infected workstation, it would mean disaster, as workstations far and wide would infect every MBS when infected network executables were executed. This ought never to be allowed to happen.

The scanner can find the virus in a file; it just neglects to see if the computer you are using is infected! Only a few scanners were tested, so it is possible that several others could make the same mistake.

Why is this virus named Sticky? Perhaps Sticky is the best description of the situation in which you would find yourself if you infected your server with this virus. Remember: just because your scanner can find a virus on a clean system (the popular kind of test), that does not mean that it will help you much if you run into a virus infection, or if you scan without clean booting.

Screaming_Fist.927

Aliases: Sticky, Nu_Way, Multi2, Fist.927.

Type: Primitive multi-partite, encrypted, COM/EXE infector, MBS dropper.

Self recognition in MBS:

18 bytes starting at offset 1Ah.

Self-recognition in Memory:

'Are you there?' call, AX=0FEEFh, value 1234h returned in the AX register.

Self-recognition in Files:

COM: fourth byte is equal to the second byte minus 1. EXE: initial IP set to 1.

Hex Pattern in MBS:

```
06D3 E353 8EC3 B8B3 0050 33DB
B902 00BA 8000 B802 02CD 13CB
```

Hex Pattern in Memory:

```
83ED 04B8 EFFE CD21 3D34 1274
518B DD81 C39F 0406 0E07 061F
```

Hex pattern in COM/EXE files:

```
5D8B F556 B98A 03B3 ???? 2E30
1C46 E2F9 C3
```

Intercepts: Int 13h for MBS read/stealth, Int 21h for COM/EXE infection, Int 1Ch (temporarily) during installation to hook Int 21h.

Trigger: None known.

Removal: To remove from the MBS, boot clean, and use the command FDISK /MBR (DOS version 5 and above). To remove from files, boot clean, and replace the infected file with a clean backup copy.

VIRUS ANALYSIS 3

Little.Red: Who's Afraid of the Big Bad Wolf?

Richard Ford
NCSA

Every time I disassemble another virus I swear it will be the last one I do… ever! There is nothing more depressing than sorting through hundreds (or even thousands) of machine code instructions, in an attempt to work out what the virus author had in mind when he wrote the beast.

When a new virus appears in the wild, users and security experts alike are understandably concerned about the possible trigger mechanisms within the virus, and want an assurance that nothing devious lurks within the code.

Identification and analysis of the virus is the only way to answer this question. And that means someone like me sitting down with debugger in hand and starting to type. Viruses - love them or hate them, it would seem that for me, at least, you can't ignore them.

Standard and Non-standard

At first sight, Little.Red seems to be a fairly trivial file infector, capable of infecting COM and EXE files, appending 1465 bytes to their length. However, analysis shows that it does have a few tricks up its sleeve - certainly enough to keep me amused for a couple of hours.

Execution differs for installation from EXE and COM files, although the code is functionally similar. For the moment, let us consider how the virus operates when an infected COM file is executed.

Like most COM infecting viruses, Little.Red inserts an initial JMP (E9h) instruction at the start of infected files. Unusually, this is directed to the following block of code:

```
  MOV SI, offset COM_startup
  JMP SI
```

This transfers control to the virus' own installation routine. I have no idea what the virus author had in mind when he wrote this construction, but can see little or no point to it. The code is not polymorphic (which was my first guess) and should present no difficulty to scanners or debuggers.

The installation routine first sets up its own stack (neglecting the usual courtesy of disabling interrupts during the process), and then carries out an 'Are you there?' call (Int 21h, Function 30h). If a copy of the virus is already resident, the call returns with the BL register set to 5Bh, and the virus repairs the host program in memory, returning control to it.

The 'Are you there?' call is made in a somewhat unusual way. The code called is initially 'encrypted' in memory by being XORed with a variable key. However, the virus contains a small routine which will decrypt these sections of code (of which there are several) on the fly, execute the routine, re-encrypt it and return.

Although this routine is a nuisance when debugging, it does not cause any particular problems. It may, however, assist the virus in attempts to avoid triggering the heuristic flags of certain scanners.

In the event that the virus is not found resident, processing passes via a CALL instruction (something of which the author of this virus appears to be exceedingly fond) to the installation routine proper.

This routine simply uses the DOS call Int 21h, function 4Ah (Resize Memory Block) to shrink the current memory block by 6Dh paragraphs, and copies itself to the end of the previously allocated block, altering the MCB chain accordingly. Finally, processing passes to the newly-resident copy of the virus code via a RETURN FAR instruction.

> *"Little.Red is a fairly average virus, which does not deliberately damage any data stored on the computer"*

The final part of the installation routine hooks Int 21h and Int 1Ch, attempts to infect the file C:\COMMAND.COM, and then returns control to the host program. Little.Red was incapable of carrying out this process flawlessly on my test machine, and on booting, I was continually confronted with the somewhat unhelpful message 'Bad or missing COMMAND interpreter'.

Presumably, the virus author wrote his creation to be DOS version-specific; therefore, given that the virus' first action on execution is to infect this file, I am confident that its days are numbered, as users upgrade to other DOS versions.

Little.Red operates in a similar manner when it loads from an EXE file, with the exception that processing passes to a different location. This allows it to take account of the different requirements when preparing to repair the file's memory image.

Resident Operation

The resident operation of the virus can be broken down into three main parts: infection and partial stealth on the DOS functions Find_First and Find_Next, infection on the

Load_and_Execute subfunction, and the trigger routine. In the tradition of saving the 'best' till last, I will deal with the infection routine first.

When resident, several Int 21h subfunctions are hooked. These are:

```
AH = 11h    ;  Find_First using FCBs
AH = 12h    ;  Find_Next using FCBs
AH = 30h    ;  Get DOS version
AX = 4B00h  ;  Load_and_Execute
```

The subfunction 30h routine is simply an 'Are you there?' call, and returns the virus' ID code (5Bh) in the BL register.

Infection on the DOS Load_and_Execute function is sufficiently simple-minded to merit no further comment, save that corruption of some EXE and COM files will inevitably occur, due to several slapdash pieces of programming. The virus' actions upon Find_First and Find_Next calls are much more interesting, and represent a long-winded and somewhat novel attempt at providing stealth.

In essence, calls to the original DOS Int 21h functions 11h and 12h handlers are allowed to complete under the control of the virus. If a file is found, the virus code searches the DTA to see if the file has the extension EXE or COM. If so, its name is extracted and converted into a form suitable for digestion by the virus' infection routine (accessed by a CALL instruction).

The outcome of this operation is returned in the contents of a particular memory offset. If the target file is already infected, the virus' length is subtracted from the length of the file returned by the Int 21h call. As a quasi-stealth routine, this works well and is sufficient to hide the size increase of infected files when the user types DIR.

However, instead of being content with using this routine to hide the change in length of infected files, the virus author was unable to resist the temptation to make his 'child' a fast-infector, making it try to infect each file on a DIR command.

While this has the desired result, it also slows the machine down to a 'snail's pace' when there are more than a handful of clean EXE or COM files in a directory. This stop-go motion (reminiscent of early attempts at animation) is accompanied by much disk clanking and clicking. The unforeseen side effect rather obviates the benefit of the stealth routines the author spent so many hours hand-crafting, and provides another clue to the virus' presence.

Trigger

Of the entire virus, the trigger routine is the most elegant part: I am undecided as to whether the author copied it from elsewhere. The routine is not intentionally destructive, and will only be deeply offensive to those who enjoy music.

When the virus calls its infection routine, it checks the date. If the year value is larger than 1994, and the date is 26 December or 9 September [*The birth and death dates, respectively, of Mao-Tse Tung. Ed.*], a timer value is set within the Int 1Ch handler. When Int 1Ch is called (on a normal DOS machine, this is about 18.2 times per second), the counter, initially set to FFFFh, is decremented by one.

After approximately one hour, the virus drops into its trigger routine. This consists of playing one of two Chinese tunes: on 26 December, 'Liu Yang River' (about the river running through Hunan province, Mao's birthplace); and on 9 September, 'The East is Red'.

No intentional damage is done to data stored on the machine, though as an aside, the trigger routine did indirectly result in hardware damage: when I returned to my infected machine after lunch, an irritated colleague had disconnected the wires to the PC's speaker, in an attempt to gain some peace and quiet. That should teach me a lesson for leaving notes saying 'Please do not turn off' on my test machines…

In Summary

Little.Red is an average virus, which does not deliberately damage data stored on the computer. Its trigger routine is annoying but innocuous, and the obvious way it slows down an infected machine makes it fairly easy for a user to spot.

Little.Red

Aliases: Mao.

Type: Memory-resident non-polymorphic file infector with partial stealth capabilities.

Infection: COM/EXE files.

Self-recognition in Memory:

'Are you there?' call (Int 21h, Function 30h) returns a value of 5Bh in the BL register.

Self-recognition in Files:

EXE: Initial SP - Initial IP = 693
COM: First byte = JMP. A mathematical operation (ROR 1) is then performed on the second and third-byte word. If the result is the same as the fourth and fifth-byte word, the file is infected.

Hex Pattern:

```
8BC4 8BE6 81C4 F905 5056 81C6
C000 B905 00EB B400 5E74 040E
```

Intercepts: Infection - Int 21h; trigger - Int 1Ch.

Trigger: Two tunes played through speakers if year is later than 1994 and date is 26 December or 9 September.

Removal: Under clean system conditions, delete infected files, replacing with known clean copies.

COMPARATIVE REVIEW

Summer Scanners

It has been six months since *Virus Bulletin* last carried out a DOS Scanner Comparative Review: in that time, there has been the usual flurry of new viruses appearing in the wild, not to mention in people's collections.

Since this last comparative, vendors have had that much more time to improve their detection of such viruses as SMEG.Pathogen and SMEG.Queeg, which have been found in a limited number of incidents in the wild, and still form part of the Polymorphic test-set.

Testing Protocol

The general scheme of the comparative is the same as the January one, in that there are four test-sets - 'In the Wild', 'Boot Sector', 'Standard', and 'Polymorphic'.

For this review, the In the Wild and Standard test-sets have been extensively re-worked to reflect information both in the *VB* Virus Prevalence Tables and in Joe Wells' *WildLists* for May and June 1995. The In the Wild test-set now consists of 160 samples of file infectors which are listed in one or the other of these two lists as having been found in the wild, and the Standard test-set contains 256 samples of miscellaneous file infectors.

The Polymorphic set has been enlarged, with the addition of 100 samples of a fairly new polymorphic virus called RDA.Fighter.5871. This virus is highly polymorphic, and has only been found within the last six months; therefore, it presents a new challenge to the products being tested. The numbers of other viruses in this test-set has been increased in order that it consists of an even 5000 viruses.

The total of 5418 infected files is held on one removable *Bernoulli 90* disk, and occupies 52,273,887 bytes. For an exact breakdown of viruses used, see the end of the review.

The machine used for testing was a *Compaq Deskpro 386/20e*, with 4 MBytes of memory and a 112MByte hard disk. Scan speeds were measured on this hardware.

Scan Times

The scan times for an uninfected floppy were measured using a standard 1.4MB floppy disk which contained 49 files (18 EXE, 19 COM, 12 SYS) occupying 1,433,709 bytes. Similarly, for the scan times of an infected diskette, an identical diskette holding 1,413,319 bytes (consisting of 100 EXE and 58 COM infections of Groove or Coffeeshop) was used. A *Bernoulli 90* disk with 1,253 executable files spread across 42 directories, occupying 82,270,423 bytes, was used for false-positive and speed testing.

As far as possible, tests were carried out using the default scan options for each product. Specifying certain options on scanners was unavoidable: for example, a review such as this becomes impractical if a product asks for user interaction after finding a virus, so this was turned off. Similarly, having a product beep every time it detects a virus is guaranteed to induce insanity in the reviewer before too long, so this too was (where possible) turned off.

Each product was also asked to write a log file consisting of every file which had been scanned, and the virus found in that file (if any). This was to enable the scans requiring prohibitively long runtimes to be performed overnight, writing their results to files which were examined the next morning. Use of this option on some products does not skew the speed results against them, as the time taken to write the log file to hard disk is a tiny fraction of the time taken to check the files.

Percentages

The scoring system is as described in *Virus Bulletin*, [*February 1995, pp.12-13; article Testing Protocol: The DOS Comparative Review*]. For the Standard, the Boot Sector, and the In the Wild test-sets, unweighted percentages are calculated to represent the product's score. However, in the case of the Polymorphic test-set, it is considered advantageous if a product is able to detect all the samples of one particular virus. This is, therefore, reflected in the percentage calculations.

There is a very simple reason for this weighting. If a machine becomes infected with such a virus, and the user goes through and removes all instances of the virus reported by the scanner, then he would (justifiably) expect to have eradicated the virus.

If, however, the scanner has missed just one sample, infection will recur as soon as the infected file is executed, and the user might just as well not have bothered to do the clean-up. There is a world of difference between detecting 1049 of a set of 1050 viruses, and detecting all 1050 in that same set.

Thus, to calculate the percentage as applied to the polymorphic detection rate, 75% of the total is simply made up of the total number of viruses detected. The remaining 25% is calculated by assuming that a virus is only 'detected' if *all* of the samples in that particular sub-set were correctly identified. Thus, if a product gets 99/100, 99/100 and 100/100, it would get a score calculated as follows: 75*((99+99+100)/300)+25*(100/300) = 82.8%.

The overall percentage for each product is then calculated by combining the percentages for each of the test-sets, without further weighting.

AVAST! v7.00

In the Wild	98.1%
Boot Sector	93.3%
Standard	99.2%
Polymorphic	98.0%

This is an instance of a product whose version numbers do not change as fast as the virus information contained within. However, the files tested were dated 18 May 1995, and the virus database was listed as version 7.06.

As regards detection, this is another excellent overall result for *Alwil Software*, and places the product firmly in the big leagues. In addition, the problem experienced in the last review whereby the machine hangs in multiple floppy mode appears to have been fixed.

AVScan v2.22

In the Wild	92.5%
Boot Sector	93.3%
Standard	95.3%
Polymorphic	96.0%

H+BEDV's scanner is one of the best at polymorphic detection, and with a little improvement of the In the Wild and Standard test-set detection rates, it would be a real winner. The English language version is a great help to those of us who don't speak German…

One false positive was encountered - 'Jerusalem #10'.

AVP v2.2

In the Wild	100%
Boot Sector	93.3%
Standard	100%
Polymorphic	100%

A stunning result for this Russian product from *KAMI Associates*. The test used the update file dated 23 May 1995, which enabled the product to detect every sample, bar one - the Unashamed virus.

There are only two minus points - firstly, scan times: this product is one of the slowest tested, along with *IBM AntiVirus*. However, in its default mode it checks inside packed files, unlike quite a few of the other products.

In addition, *AVP* (*Anti-Viral Toolkit Pro*) offers extremely precise identification of the viruses it finds, which also goes some way to explaining the extremely slow infected file scan times. The second problem concerned the fact that the product reported two files in the false-positive test-set as being infected with a virus of 'Type_ComExeTSR suspicion'. *AVP* offers in addition a very fine virus information section, the contents of which should make *KAMI* feel justifiably proud.

CPAV v2.2

In the Wild	85.0%
Boot Sector	73.3%
Standard	82.4%
Polymorphic	Incomplete

Once the signature updates supplied with the product had been installed, its detection rates improved somewhat, giving those listed here. However, the great tradition of *Central Point Anti-Virus* not being able to survive the polymorphic test intact continues.

It seems curious that the product is able to survive its Quality Assurance procedures if it is unable to complete the tests to which it is subjected in the *VB* comparative. Elsewhere, detection rates are unremarkable, and one false positive cropped up - One_Half was falsely reported as being in one file.

Doctor Lite v95.06

In the Wild	85.6%
Boot Sector	86.7%
Standard	86.3%
Polymorphic	52.6%

This is the shareware version of the full-blown DOS scanner from *Thompson Network Software*, but was submitted for review as the scanning engine for both is identical. Although *Doctor* is not yet up with the leaders, *Thompson* has dramatically improved its technology. One to watch for the future, perhaps.

Dr Solomon's AVTK v7.12

In the Wild	100%
Boot Sector	100%
Standard	94.9%
Polymorphic	88.3%

Another good performance from *Dr Solomon's Anti-Virus Toolkit*, which achieved perfect scores in the Boot Sector and the In the Wild test-sets. An excellent base score against the Polymorphic test-set is masked by the fact that it missed two samples of Pathogen and five of SMEG, in addition to just under half of the samples of RDA.Fighter.5871. Thus, it missed out on some of the 'bonus points' awarded for complete detection.

Product Name	In the Wild (160)	Boot Sector (15)	Standard (256)	Polymorphic (5000)	Overall (%)
AVAST!	157	14	254	4902	97.2
AVP	160	14	256	5000	98.3
AVScan	148	14	244	4800	94.3
CPAV	136	11	211	Incomplete	60.2
Doctor Lite	137	13	221	3305	77.8
Dr Solomon's AVTK	160	15	243	4950	95.8
F-Prot	158	15	228	4635	91.4
IBM AntiVirus	159	15	225	4844	93.4
InocuLAN	144	14	224	2807	79.8
McAfee Scan	146	12	223	2704	75
MSAV	84	3	194	466	38.8
Norton AntiVirus	144	14	218	1650	75.4
Novell DOS7	117	8	215	501	55.1
PCVP	139	9	218	3064	71.5
ScanVakzin 4	127	12	207	562	62.3
Sophos' Sweep	160	15	231	4900	90.2
ThunderBYTE	160	14	230	4889	92.6
VET	142	14	249	4921	89.1
ViruSafe	126	14	209	2569	73.2
Virus ALERT	159	13	229	4781	90.3
Virus Buster	126	15	208	700	68.5
Virus Buster Lite	141	14	213	716	69.7
Vi-Spy	149	13	227	3633	84.1

Whilst, as in January, no-one escaped unscathed, this time it was a very close thing. The inclusion of the Unashamed virus in the Boot Sector test-set seems to have produced the most problems, and many products failed to detect it. In the Polymorphic set, it was RDA.Fighter.5871 which did the damage - as well as being relatively new, this virus appears to be designed to give prohibitively long scan times for decryptors.

McAfee Scan v2.2.0

In the Wild	91.3%
Boot Sector	80.0%
Standard	87.1%
Polymorphic	41.6%

Using the v221 of the virus data file, it would appear that *McAfee* still needs to put in work on its polymorphic detection rate, above all. This aside, there is room for improvement in all the tests.

Microsoft Anti-Virus (MSAV)

In the Wild	52.5%
Boot Sector	20.0%
Standard	75.8%
Polymorphic	7.0%

This product is usually included to give a good baseline above which other products should fall, and by quite some considerable margin. There is not really a great deal one can say about *MSAV*, apart from ‘don’t’…

Product Name	Version Number	Time to Scan Clean Floppy (min:sec)	Time to Scan Infected Floppy (min:sec)	Time to Scan Clean Bernoulli (min:sec)	Clean Floppy Read: KB/Sec	Clean Bernoulli Read: KB/Sec
AVAST!	7	0:48	1:21	12:32	29.2	106.8
AVP	2.2	2:55	10:07	38:00	8	35.2
AVScan	2.22	1:22	2:05	31:13	17.1	42.9
CPAV	2.2	0:50	3:10	Incomplete	28	Incomplete
Doctor Lite	95.06	1:58	4:51	14:23	11.9	93.1
Dr Solomon's AVTK	7.12	0:52	15:47	9:44	26.9	137.6
F-Prot	2.17	1:10	1:55	11:09	20	120.1
IBM AntiVirus	2.2	1:56	1:30	43:25	12.1	30.8
InocuLAN	3.01	1:02	12:28	14:40	22.6	91.3
McAfee Scan	2.2	1:10	4:04	12:28	20	107.4
MSAV	None	1:06	2:15	13:50	21.2	96.8
Norton AntiVirus	3	0:28	1:55	7:39	50	175
Novell DOS7	28.02	0:49	1:55	8:07	28.6	165
PCVP	2.23	0:40	0:54	5:05	35	263.4
ScanVakzin 4	4.207	0:44	1:18	11:09	31.8	120.1
Sophos' Sweep	2.74	0:48	1:47	11:07	29.2	120.5
ThunderBYTE	6.35	0:27	1:57	3:20	51.9	401.7
VET	8.212	0:40	1:17	8:30	35	157.5
ViruSafe	6.5	0:40	2:31	8:30	35	157.5
Virus ALERT	3.34	0:27	2:32	3:21	51.9	399.7
Virus Buster	4.76	0:45	10:08	13:42	31.1	97.7
Virus Buster Lite	4.76	0:31	10:30	8:55	45.2	150.2
Vi-Spy	12	0:56	1:37	11:41	25	114.6

Once again *ESaSS' ThunderBYTE* scanning technology leads the way in terms of scan time with a very impressive 401.7 KB/Sec from the *Bernoulli* drive. Contrary to the last review, *CPAV* was the only product which failed to complete the clean *Bernoulli* test. Particular attention should be paid to the floppy disk scan times, as these represent a frequent task for a scanner.

Novell DOS7

In the Wild	73.1%
Boot Sector	53.3%
Standard	84.0%
Polymorphic	10.0%

Like a certain other product, the detection rates of the scanner 'Search And Destroy' (version 28.02, from *Fifth Generation Systems Inc.*) included with *Novell DOS7* are poor, chiefly because of its age.

F-Prot Professional v2.17

In the Wild	98.8%
Boot Sector	100%
Standard	89.1%
Polymorphic	77.8%

Command Software's F-Prot fell foul of the Polymorphic test-set, which dragged its eminently respectable score lower than it otherwise would have been. *F-Prot* is extremely well respected for the accuracy with which it identifies viruses, and this is none the less true with this test-set. This aside, it would be nice if the product had identified more of the samples in the test-sets.

IBM AntiVirus v2.2

In the Wild	99.4%
Boot Sector	100%
Standard	87.9%
Polymorphic	86.2%

This completely revamped product from *IBM* scored considerably better than the last version tested - its polymorphic score went up by almost the same order of magnitude as its version number.

There was a minor problem in the installation: before rebooting, the installation routine appeared to attempt to execute *IBMAV*, and then sent up the message 'Unable to execute IBMAV'. However, a scan did take place after the reboot, and there were no problems.

Along with *AVP*, this was one of the slowest products tested, in particular where infected files were concerned; however, this has resulted in a seemingly much improved detection rate.

InocuLAN v3.01

In the Wild	90.0%
Boot Sector	93.3%
Standard	87.5%
Polymorphic	48.4%

A dramatic improvement in polymorphic detection rate belies the small increase in version number - unlike *IBMAV*, it seems for this product they are not mathematically related.

Similarly, Boot Sector detection is better, with the product, like many others, only missing Unashamed. As ever, however, there is room for improvement in the In the Wild and Standard test-sets, and in spite of the improvement, Polymorphic detection is a long way off perfect.

Norton AntiVirus v3.0

In the Wild	90.0%
Boot Sector	93.3%
Standard	85.2%
Polymorphic	33.0%

This product arrived (like *CPAV*) with a signature update disk, which explains why the results are different from the otherwise identical product tested in January. Polymorphic detection, however, does not seem to have improved since then, which is a shame, as detection in the other categories is not as bad as it might have been.

PCVP v2.23

In the Wild	86.9%
Boot Sector	60.0%
Standard	85.2%
Polymorphic	53.7%

For some reason, this product always seems to have trouble specifically with the Boot Sector test-set. Even its other scores are far from impressive, especially those in the Polymorphic test-set. Particularly noticeable in this area is the product's inability to detect One_Half.

ScanVakzin v4.207

In the Wild	79.4%
Boot Sector	80.0%
Standard	80.9%
Polymorphic	8.9%

This Japanese product is still unable to detect Quox, Peanut, and Unashamed from the Boot Sector test-set, and polymorphic detection is little better than appalling. The remaining results, whilst better, never quite manage to get out of the 'mediocre' category.

Sophos' Sweep

In the Wild	100%
Boot Sector	100%
Standard	90.2%
Polymorphic	98.0%

An improved result for *Sophos' Sweep*, with a high percentage scored in the Polymorphic tests because of the weighting given for complete identification of a particular virus group. The only viruses missed overall were the samples of Cruncher and RDA.Fighter.5871.

ThunderBYTE v6.35

In the Wild	100%
Boot Sector	93.3%
Standard	89.8%
Polymorphic	87.3%

Another product whose only fault in the Boot Sector test-set was missing Unashamed. The flawless performance against the In the Wild test-set is mirrored by above average detection rates of the viruses in the Standard and Polymorphic test-sets, although results seem to have worsened since the last *VB* Comparative Review in January.

The simply staggering speed of the product when run on files which are not infected [*It's easy to run out of superlatives for TBAV's scanning speed. Ed.*] makes it a very desirable product. However, the advanced heuristics which the product utilises are not yet infallible, as one false positive was reported - the message given was: 'probably infected by an unknown virus'.

VET v8.212

In the Wild	88.8%
Boot Sector	93.3%
Standard	97.3%
Polymorphic	76.8%

In spite of the product's generic boot sector detection techniques of comparing the sector to known valid ones, it failed for some reason to detect the sample of Unashamed. Another good overall set of results, although slightly down from the last comparative. *VET* is another product whose polymorphic score is low, due to the fact that it missed a few samples of the polymorphic viruses which it has the technology to detect.

ViruSafe v6.5

In the Wild	78.8%
Boot Sector	93.3%
Standard	81.6%
Polymorphic	39.0%

A good detection rate against the Boot Sector test-set helped to offset uninspiring Standard and In the Wild scores, and a definitely unimpressive Polymorphic detection rate. However, the scores against the Polymorphic and Boot Sector test-sets have improved since January - unlike those for the In the Wild and the Standard test-sets - so things may be 'on the up'.

Virus ALERT v3.34

In the Wild	99.4%
Boot Sector	86.7%
Standard	89.5%
Polymorphic	85.7%

This product missed the BootEXE.451 and Unashamed samples in the Boot Sector test-set, which was something of a surprise. Polymorphic detection is much better than in January, but there is still ample room for improvement.

The product produced the same false positive as *ThunderBYTE*, which is perhaps unsurprising when you realise the scanning engine is virtually identical - not, however, completely so. The scores are, all round, fractionally lower than those for *ThunderBYTE*.

Virus Buster v4.76.00

In the Wild	78.8%
Boot Sector	100%
Standard	81.3%
Polymorphic	14.0%

I cannot help but wonder what happened with this test - as can be seen, the results given below for *Virus Buster Lite* are better than those for the full product. Whilst *Virus Buster* does not have a default scan type, the results for both fast and secure modes were identical.

Polymorphic detection is unimproved since January, and results of the runs against the In the Wild and the Standard test-set are unimpressive.

On the positive side, detection in the Boot Sector test-set was a strong point: it is unfortunate that this was offset by the scores in the other categories.

Virus Buster Lite v4.76

In the Wild	88.1%
Boot Sector	93.3%
Standard	83.2%
Polymorphic	14.2%

Much the same results as for *Virus Buster*, although (as mentioned above) inexplicably higher when run against the Standard, the In the Wild, and the Polymorphic test-sets.

Vi-Spy v12.0

In the Wild	93.1%
Boot Sector	86.7%
Standard	88.7%
Polymorphic	68.0%

Despite having the same number as the previously reviewed version, the file dates are much later, and it is to be assumed that new information has been added. It is interesting to note that the product is still unable to detect Quox on a floppy diskette, and that its detection of Pathogen is 'by the back door' - detecting the suspicious file date. *RG Software* should work to detect the virus by more reliable means - this is not a technique which can be trusted. In the Wild detection, however, was reasonably good.

Conclusions

This review is one of the most difficult to date - of particular note is the Polymorphic test-set, a collection of 5000 highly polymorphic viruses chosen for its extreme complexity.

One product emerged from the experience blissfully unscathed - *KAMI Associates' AVP*. This Russian product, whilst being one of the slowest tested, offers a level of identification which no other product even approached. Products gaining a honourable mention include *Alwil Software's AVAST!*, which offers considerably more than just a scanner; *Sophos' Sweep* and *Dr Solomon's AVTK*, both of which have turned in consistently good results over many years; and *H+BEDV's AVScan*.

It is interesting to note the effect of the weighting of the polymorphic detection rate. For example, looking at *H+BEDV's AVScan* and *Dr Solomon's AVTK*, the former detected 4800 of the polymorphics, and the latter detected 4950. However, after weighting, the former gains 96%, and the latter 88.3%.

The wide range of scan times is also worthy of note - the time taken to scan a clean floppy ranged from 27 seconds to just under three minutes. This 600% difference is non-trivial, and very relevant in the everyday usage of an anti-virus scanner. It is important to find a product which strikes a balance between detection and speed.

One encouraging thing is the increased awareness of the Quox virus - the number of scanners failing to detect it has dropped from thirteen to five over six months.

As a final word, it is wise to examine the results in the In the Wild category with care. If the product you use has scored badly against this test-set, find out why - these are the viruses which today present the most risk. However, as the news story on p.3 [*Cruncher 'In the Wild' in Russia*] shows, today's collection virus is in the wild tomorrow.

TEST-SETS

One sample of each is included, unless otherwise indicated by the number in parentheses after the virus name.

In the Wild

160 genuine infections of:

Anticad.4096 (4), Arianna.3375 (4), Avispa.D (2), Barrotes.1310.A (2), BootEXE.451, Butterfly.Butterfly, Captain_Trips (4), Cascade.1701, Cascade.1704, Chill, Coffeeshop (2), CPW.1527 (4), Dark_Avenger.1800.A (3), Dark_Avenger.2100.DI.A (2), Datalock.920.A (3), Diamond.1024.B, Die_Hard (2), Dir-II.A, DOS_Hunter, Fichv.2.1, Flip (2), Flip.2153 (2), Frodo.Frodo.A (4), Ginger, GoldBug (4), Green_Caterpillar.1575.A (3), Helloween (4), Hidenowt, HLLC.Even_Beeper.A, Jerusalem.1808.Standard (2), Jerusalem.Sunday.A (2), Jerusalem.Zero_Time.Australian.A (3), Junkie, KAOS4 (2), Keypress.1232.A (2), Lamer's_Surprise, Lemming (2), Liberty.2857.D (2), Little.Red (2), MacGyver.2803.B, Maltese_Amoeba (3), Necropolis, Necros (2), Neuroquila, No_Frills.No_Frills.843 (2), No_Frills.Dudley (2), Nomenklatura (4), Nothing, November_17th.855.A (2), Npox.963.A (2), Number_of_the_Beast (5), Peanut, Predator.2448 (2), Quicky, Revenge, Riihi, Sat_Bug.Natas, Sat_Bug.Sat_Bug (2), Sayha (2), Screaming_Fist.927 (4), Screaming_Fist.696 (2), Stardot.789.D (2), SVC.3103.A (2), Tai-Pan.666 (2), Telecom (4), Tequila.A, Trojector.1463 (6), Trakia.653, Tremor.A (6), Vacsina.TP-05.A (2), Vacsina.TP-16.A, Vampiro, Vienna.648.Reboot.A, VLamix, Voronezh.1600.A (2), Yankee_Doodle.TP.44.A, Yankee_Doodle.XPEH.4928 (2).

Boot Sector

One genuine infection of each, on separate 1.44M 3.5-inch diskettes, of:

AntiEXE, BootEXE.451, EXE_Bug.A, Form, Junkie, LZR, Natas, NoInt, NYB, Parity_Boot.B, Peanut, Quox, Sampo, Stoned.Empire.Monkey.B, Unashamed.

Polymorphic

5000 genuine infections of:

Girafe (1050), Groove and Coffeeshop (500), One_Half (1050), Pathogen (1050), RDA.Fighter.5871 (100), Sat_Bug.Sat_Bug (100), SMEG_V0.3 (1050), Uruguay.4 (100).

Standard

256 genuine infections of:

1049, 1260, 12_Tricks, 1600, 2100 (2), 2144 (2), 405, 417, 492, 5120, 516, 600, 696, 707, 777, 800, 8888, 8_Tunes, 905, 948, AIDS, AIDS-II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax, Anti-Pascal (5), Argyle, Athens (2), Armagedon, Attention, Bebe, Big_Bang, Black_Monday (2), Blood, Burger (3), Cascade (2), Casper, Crazy_Lord (2), Cruncher (25), Dark_Avenger.Father (2), Darth_Vader (3), Datacrime (2), Datacrime_II (2), December_24th, Destructor, Dir, DiskJeb, DotKiller, Durban, Eddie, Eddie-2.A (3), Fax_Free.Topo, Fellowship, Fish_1100, Fish_6 (2), Flash, Fu_Manchu (2), Genesis.226, Halley (1), Hallöchen.A (3), Hymn (2), Icelandic (3), Internal, Invisible_Man (2), Itavir, Jerusalem.PcVrs.Ds (4), Jocker, Jo-Jo, July_13th, Kamikaze, Kemerovo, Kennedy, Lehigh, Liberty (5), Loren (2), LoveChild, Lozinsky, Macho (2), MIX1 (2), MLTI, Monxla, Murphy (2), Nina, NukeHard, Old_Yankee (2), Oropax, Parity, Perfume, Phantom1 (2), Pitch, Piter (2), Poison, Polish-217, Power_Pump.1, Pretoria, Prudents, Rat, SBC, Semtex.1000, Shake, Sibel_Sheep (2), Spanz (2), Starship (2), Subliminal, Sunday (2), Suomi, Suriv_1.01, Suriv_2.01, SVC.1689.A (2), Sverdlov (2), Svir, Sylvia, Syslock, Syslock.Macho (2), Syslock.Syslock.A, Taiwan (2), Terror, Tiny (12), Todor (2), Traceback (2), TUQ, Turbo_488, Typo, V2P6, variants of Vacsina.TP (6), Vacsina.Penza.700 (2), Vacsina.634, Vcomm (2), VFSI, Victor, variants of Vienna (11), Virdem, Virdem.1336.English, Virus-101 (2), Virus-90, VP, V-1, Warrier, Warrior, Whale, Willow, WinVir_14, variants of Yankee_Doodle.TP (5), Zero_Bug.

PRODUCT REVIEW

StationLOCK

Dr Keith Jackson

StationLOCK is a small plug-in card which requires a PC with an ISA bus. The card contains a 44-pin chip, an EPROM, and a DIP selector switch. *StationLOCK* software includes its own *MS-DOS*-compatible operating system. The product offers access control and virus protection security features, and, as the software is stored on a plug-in card, it activates before DOS boots. This is a major advantage in trying to combat boot sector viruses, which gain control in advance of the operating system, as it permits security programs to execute before *MS-DOS* takes over.

StationLOCK allows up to seven users on each PC, and can provide different security settings for each user. Users can be restricted to specific, logical partitions of a disk drive. Floppy disk drives can be disabled or made read-only on a per user basis. Other configurable features are concerned with disabling disk formatting facilities, and making available various combinations of the serial and/or parallel port(s). An audit trail of which users logged on at what date/time can be activated as necessary.

StationLOCK provides three levels of protection: a scanner, a memory-resident program, and a boot-sector scan of all diskettes. The *MS-DOS* system files IO.SYS, MSDOS.SYS and COMMAND.COM are checksummed by *StationLOCK* during each PC boot. In this review, I shall concentrate on testing the anti-virus features rather than looking at the available multi-user control features.

Installation

I have previously reviewed a similar product, *PC-cillin,* from the same developer [*see VB, December 1993 pp.20-22*]. This contained many similar features to *StationLOCK*. The hardware in this case was a dongle attached to the PC's parallel port. Thankfully, the developers have moved on, and now provide a plug-in card.

The installation process involves opening the PC, setting the plug-in card's DIP switches to an available memory address, inserting the card, and switching the PC on. When a PC is first powered on with the card installed, the software stored on the card detects that this is the first time the card has been used, and displays its ID number onscreen.

It is important, the documentation stresses, to keep this ID number safe: it is unique to each copy, and support personnel cannot provide immediate help unless it is known. If it is not, a 'fix' is sent by post. This caveat applies also to the Rescue Disk, which *StationLOCK* offers to create on install. The Rescue Disk may be used to restore the hard disk's boot sector and/or partition sector after damage/alteration.

During installation, *StationLOCK* can install the 'automatic virus checker' (a TSR) and the screen blanker into the file AUTOEXEC.BAT. The appropriate files are copied from the plug-in card to the hard disk's root directory. A hidden subdirectory called SLOCK is created in C: drive's root directory, and is used to hold log files created by *StationLOCK*. A virus pattern file (178 KB) is used to keep the scanner up to date: it must also be installed in the root of the C: drive.

It is possible to log in as the MASTER user, change the MASTER user's password, create users with specific IDs and passwords, and tailor security settings for each user. The MASTER user has access to all system components and all disk drives, and 'only MASTER users can access the MASTER setup program'.

If a *StationLOCK*-protected PC is switched on, and nobody logs in for 20 seconds, the PC will boot with a user ID of DEFAULT - the facilities associated can be tailored as necessary. The feature enables unmanned activation of a PC (perhaps by using a timing device on the power switch).

Although installation proved straightforward, I did find a few quirks in the process. The manual says that any type of disk can be used for the Rescue Disk, as long as the correct type of diskette for drive A is used. The README file provided on the *StationLOCK* diskette then contradicts this, stating that only high-density disks may be used. However, I made a Rescue Disk using a low-density disk, and nothing appeared to fail.

I tried to request extra drives at install time so *StationLOCK* could 'see' my Magneto-Optical drive, which is connected to the parallel port via a *Trantor* interface. This requires two device drivers: one to provide access to the *Trantor*; one to allow access to the Magneto-Optical drive. This fooled *StationLOCK* - it could access nothing with the device drivers. Similarly, no matter what I did, *StationLOCK* could not 'see' my *Stacker* drive.

The manual states that both *Stacker* and *SpeedStor* drives are recognised automatically, and that it is not necessary explicitly to provide a device driver line in *StationLOCK's* device driver file, X-LOG.SYS. However, *StationLOCK* failed to recognise my *Stacker* drive automatically. If I did name the *Stacker* device driver, it said that all sorts of files on the Stacked drive were infected, then produced random streams of gibberish for virus names.

Operation

After the installation routine is complete, the hard disk is scanned every time the PC is booted. The user logs on to the PC with the correct ID/password combination, and DOS boots as normal. Scanning can be interrupted by pressing the

Escape key. This has the advantage of speeding up the boot process if multiple reboots are required, but, conversely, scanning can be bypassed by users whenever they so wish.

On executing *Windows*, *Microsoft's* software produced a message warning that the 32-bit disk driver could not be loaded, and that unrecognised software had been found: it advised me to run an anti-virus program. However, *Windows* did continue to load. The message was produced even though *StationLOCK's* memory-resident component was inactive. If 32-bit disk access is deactivated within *Windows*, the problem disappears, but disk access slows down.

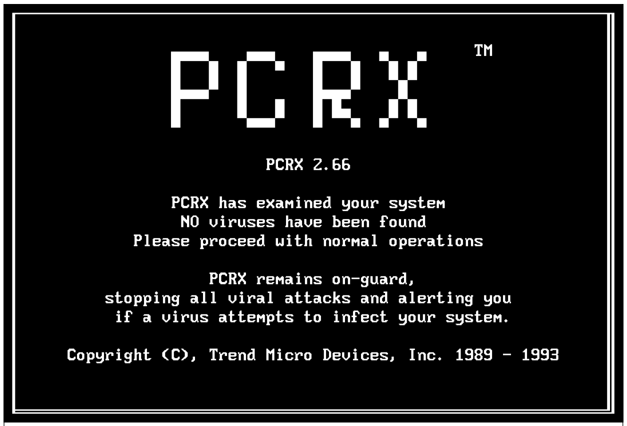
Everything worked correctly on my PC until I tried to download files from the *CIX* conferencing system. I use a modem which operates at 14.4 Kbps - this can provide data at the serial port at a rate requiring 38.4 Kbps serial communication between the modem and the serial port.

All functioned correctly when text messages were transmitted; however, when ZMODEM (a communications protocol) file transfers commenced, continual errors were detected by the communications software. These were so frequent that they prevented file transfer taking place. When I removed the memory-resident programs, the errors disappeared.

Documentation

The documentation comprises a 135-page A5 manual, a nine-page A5 booklet providing a 'Function Update' for new/altered features, a quick reference card, and various bits of bumph. The manual has a decent table of contents and is thoroughly indexed, providing many examples of how *StationLOCK* should be used.

However, it contains some claims which do not stand up to scrutiny, and a few entries which are simply wrong. The manual says that 'the StationLOCK package contains no program diskettes because the complete StationLOCK software is contained within the ROM chip on the card'. Then what is this floppy disk in my hand? Perhaps providing the virus pattern file on diskette is a recent development.



```
PCRX ™

PCRX 2.66

PCRX has examined your system
NO viruses have been found
Please proceed with normal operations

PCRX remains on-guard,
stopping all viral attacks and alerting you
if a virus attempts to infect your system.

Copyright (C), Trend Micro Devices, Inc. 1989 - 1993
```

PCRX (referred to in the documentation as PCRXVT) is a memory-resident behaviour blocker which protects the CMOS and boot sectors.

Likewise, the statement that 'StationLOCK acts like a mini network on a single PC' is misleading. *StationLOCK* simply provides some level of access control and virus protection.

Scanning

The scanner normally works at boot-time, executing before DOS boots. However, *StationLOCK* files may be copied to the hard disk and executed from there, using a ROM-based feature called 'Rescue Boot'.

Detection capabilities are determined by the latest virus pattern file present on hard disk, which is protected against alteration. No matter where I tried to introduce a single bit change into the virus pattern file, the scanner detected that something had changed: it did not report that the file was altered, but said simply that it could not find the file.

Seeking 'program files only', *StationLOCK* scanned my test computer's hard disk in 35 seconds (37 subdirectories, 892 files), rising to 1 minute 34 seconds when scanning all files. In comparison, *Dr. Solomon's AVTK* took 35 seconds, and *Sophos' SWEEP* 36 seconds, to perform the same scan.

The scanner produces a summary of what has been scanned in the middle of the screen. If no viruses have been detected, the summary disappears after a few seconds. If a virus was detected, the summary remains in place until a keypress removes it. If a floppy drive is scanned, but no diskette is present, the scanner produces a summary report - of nothing!

Detection Accuracy

When run against the test-set (see Technical Details), the scanner detected 236 (95%) of the 248 virus-infected test samples. Those which were missed were WinVir_14, Todor, Power, Tremor, Coffeeshop, Starship (2 samples), Invisible_Man (2 samples), NukeHard, 8888 and Halley.

I was surprised that the detection rate was not as high as the previous *Trend* product I reviewed; however, checking showed that *PC-cillin* failed to detect only five viruses (Pitch, plus the first four in the above list). The others had been added to my test-set since the earlier review. *StationLOCK* detected all the old test-set viruses: possibly an indication that developers are having problems keeping up with new viruses. When run across 500 Mutation Engine (MtE) samples, only 27 (5.4%) were detected correctly.

Detection of boot sector viruses was rather curious. In eight out of nine cases, *StationLOCK* produced three warnings for each virus. The virus name only appeared on the second warning screen; the first warning screen simply indicated that an infection was present. The exception was Monkey, which, when tested, produced only one warning screen.

I tried to obtain a directory listing of each disk infected with a boot sector virus: for all except Monkey, the usual three warning screens were produced. It was the same when the memory-resident segment of *StationLOCK* was active. Apart from Monkey, *StationLOCK* spotted the boot sector viruses.

Memory-resident Program

The memory-resident part of *StationLOCK*, PCRXVT (a memorable name if ever there was one!), occupies 14.3 KB of memory. Another copy of this file is placed on the hard disk's root directory, and *StationLOCK* checks that the two are identical before PCRXVT becomes memory-resident.

Although the manual explains that the memory-resident program scans files before execution, the booklet says that this has changed, and that the memory-resident program is now a behaviour blocker. If PCRXVT is executed when already memory-resident, an error message is produced, warning that it is resident.

The overhead introduced by having PCRXVT active in memory was measured by timing how long it took to copy 40 files (1.3 MB) from one subdirectory to another. Without PCRXVT, they could be copied in 21 seconds, rising to 35 seconds with PCRXVT installed - an increase of 71%.

I tried to get PCRXVT to trigger properly; however, it must be said that in my testing, I saw no onscreen message stating that it had been produced by PCRXVT. In fact, I am fairly sure that all the conditions discussed were detected by the *StationLOCK* plug-in card itself, not PCRXVT. Is this perhaps a bid to produce completely transparent software?

Even though PCRXVT is a behaviour blocker, it let me format a diskette, read the CMOS, and reset the date and time (which writes to CMOS) - this was because specific security features are disabled by default, and must be enabled. I tried formatting a 3.5-inch diskette with format protection enabled using the FORMAT command. All seemed normal, but at the end of the process a write-protect error was returned. When *Norton's* Safe Format command was used, the error was returned immediately. The disk was not write-protected.

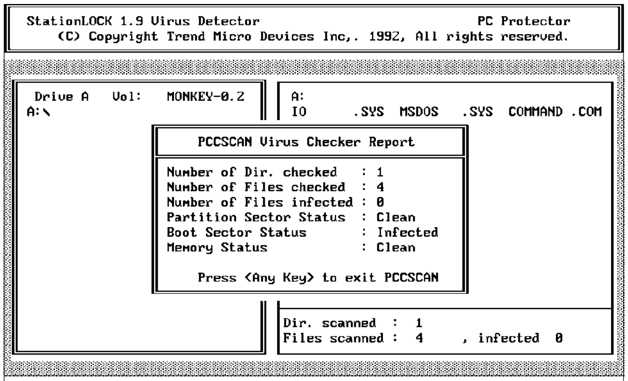
The manual warns about this, and states that formatting will appear to work but do nothing. This statement is in fact incorrect: after the product 'prevented' formatting taking place, my test floppy caused DOS to issue a 'General Failure' error whenever it was accessed.

Even though I instructed *StationLOCK* to set drive A as read-only, *Norton Commander* had no problems copying files onto a diskette in the supposedly read-only drive A. This is a bug which should be fixed.

The Rest

StationLOCK incorporates encryption features which can be activated as desired. On a hard disk, only the partition sector is encrypted, not data content. Introducing general data encryption would probably impose too much overhead.

Floppy disk data seems to be totally encrypted. I tested this by copying 20 files (612 KB) to a 3.5-inch, 720 KB floppy, then asking *StationLOCK* to encrypt it: encrypting the whole disk took 1 minute 24 seconds. Data on the encrypted floppies can be separated between groups of users using a 'token' scheme: users who have the same token can read each other's disks.



```
StationLOCK 1.9 Virus Detector                          PC Protector
    (C) Copyright Trend Micro Devices Inc,. 1992, All rights reserved.

 Drive A   Vol:   MONKEY-0.2          A:
A:\                                   IO      .SYS  MSDOS   .SYS  COMMAND .COM

             PCCSCAN Virus Checker Report

             Number of Dir. checked     : 1
             Number of Files checked    : 4
             Number of Files infected   : 0
             Partition Sector Status    : Clean
             Boot Sector Status         : Infected
             Memory Status              : Clean

                 Press <Any Key> to exit PCCSCAN

                                       Dir. scanned  :  1
                                       Files scanned :  4     , infected  0
```

StationLOCK correctly identified all boot sector viruses in the *Virus Bulletin* test-set.

Conclusions

StationLOCK can only be used on a PC with an ISA bus. I also do not like the idea that *Trend's* Technical Support cannot provide immediate help unless the ID number of the *StationLOCK* card is known. The most likely person to hit problems is a user, who is unlikely to know the ID number.

StationLOCK in its current state cannot support high-speed serial data transfer, and has no knowledge of *Windows*. Given that the *StationLOCK* software is held in an EPROM, whatever bugs are present are going to be there until the developer supplies a new EPROM, an expensive hobby which manufacturers usually try to avoid.

I would recommend using this product only where there is a clear need for multiple users on one PC. This is a shame, as the scanner is reasonably fast, and good at detecting non-polymorphic viruses, and the product takes control before a virus can foul things up. *StationLOCK* is a step forward for the company, but still has some way to go in terms of *Windows* compatibility and polymorphic virus detection.

Technical Details

Product: *StationLOCK PC Protector v1.9* (including v5.0 of the X-DOS ROM-based operating system).

Serial number: 0A-10-24-1F.

Developer: *Trend Micro Devices Inc.*, 1F,#28 Li-Shui St., Taiwan, R.O.C. Tel +886 2 3120191; Fax +886 2 3412137.

US Office: 2421 West 205th Street, Suite D-100, Torrance, California 90501. Tel +1 310 782 8190; Fax +1 310 328 5892.

Availability: Any PC with a hard disk and an ISA bus.

Price: US$129 - single user. Site licence negotiable. Updates every two-three weeks.

Hardware used: A 33 MHz 486 PC clone with one 3.5-inch (1.4 Mbyte) floppy disk drive, one 5.25-inch (1.2 Mbyte) floppy disk drive, a 120 MB hard disk and 4 MB RAM, using *MS-DOS v5.00*, *Windows v3.1* and *Stacker v2*.

NB: For full details of viruses used for testing purposes please see *VB*, May 1995, p.23.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

Tel 01235 555139; International Tel +44 1235 555139
Fax 01235 531889; International Fax +44 1235 531889
Email editorial@virusbtn.com
CompuServe 100070,1340@compuserve.com

US subscriptions only:

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

Tel +1 203 431 8720; Fax +1 203 431 8165

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

END NOTES AND NEWS

ADS (*Computer Systems*) has announced the launch of a new 'mega-product' which they hope will be a commercial success in the UK. The company has compiled **a range of over 200 anti-virus and security utilities**, compressed onto eleven 1.44 MB 3-5 inch floppy disks, which they envision as addressing the security problems associated with *NetWare* systems. The package will retail at £30 inclusive. Information is available from the company on Tel +44 1332 875208. Alternatively, Email johnwalker@compulink.co.uk.

The UK *DTI* (***Department of Trade and Industry***) **has chosen *disk*net**, the access control package from *Reflex Magnetics*, to be added to the security measures already in place within the organisation. Information on the product is available from *Reflex* on Tel +44 171 372 6666; Fax 0171 372 2507.

Leprechaun Software Pty has responded to user demand by launching their anti-virus product, ***Virus Buster*, in a *Windows* version**. *Leprechaun* can be contacted on Tel +61 7 823 1300; Fax +61 7 823 1233.

The *NCSA* has announced a reorganisation of its *AVPD* (*Anti-Virus Product Developers* working group). The body **tests and certifies products from all over the world**: a programme for Development Assurance Criteria, which will measure a product's ability to meet *NCSA* standards, is also planned. Information is available from the *NCSA* on +1 717 258 1816; Fax +1 717 243 8642.

***Compsec 95* will take place in London**, UK, from 25-27 October 1995. For details on the conference, contact Jill Spear at *Elsevier Science Ltd* on Tel +44 1865 843643; Fax +44 1865 843971.

The ***First Cologne IT Security Forum*** (*1. Kölner IT-Sicherheitsforum*) will be sponsored by *datakontext tagungen GmbH* in Cologne, Germany on 12/13 July 1995. The main aim of the conference is to pinpoint areas of weakness within companies' structures, and the risks posed by these weak areas. Price for both days is DM 1850, inclusive of VAT. Further details are available by contacting either the company on Tel +49 2234 65633; Fax +49 2234 65635, or Ralf Herweg or Thomas Müthlein on Tel +49 2234 691961.

Sophos' Sweep for Windows NT is now available for *Digital's Alpha AXP* range of servers and workstations. Cost of the new product is £495 p/a (up to 25 users) and £895 p/a (over 25 users), including **monthly updates and full technical support**. For more information, contact Richard Jacobs on Tel +44 1235 544017; Email rj@sophos.com; Fax +44 1235 559935.

Three computer security-related conferences have been scheduled in London by *IBC Technical Services Ltd*: *Computer Investigations* - a one-day seminar to be held at the *Britannia Intercontinental Hotel* on 6 July 1995; *Theft from Electronic Systems*, on Friday 7 July 1995 (also at the *Britannia*), and *New Security Issues 1995*, 12/13 September 1995, to take place at the *London Hilton Hotel*. Information on the seminars is available from Lisa Minoprio on Tel +44 171 637 4383; Fax +44 171 631 3214.

Last month's edition of *End Notes and News* contained an incorrect fax number. **Readers wishing to contact *RG Software*** in Scottsdale, Arizona, USA should use the following number: +1 602 423 8389, and not that which was published previously.

McAfee Associates and *Intel Corporation* have released updated versions of their anti-virus products. *Intel's LANDesk v3.0* now includes a virus firewall, and *McAfee's NetShield v2.2* and *VirusScan v2.2* feature support for *Novell NetWare v4.1*. *Intel* can be contacted in the USA on Tel +1 503 264 7354; *McAfee* (also USA) on Tel +1 408 988 3832.