July 1992 ISSN 0956-9979

VIRUS BULLETIN

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

CONTENTS

EDITOR'S DIARY

It's A Mad, Mad World

The *Hyatt Regency* hotel is situated in Crystal City, a suburb of Washington D.C. where granite-jawed men in razor sharp trousers hail taxi cabs to the Pentagon, FBI headquarters, the White House and Capitol Hill. The rule to success and happiness in the American capital is to look important at all times. As any cab driver will tell you: '*Everyone's* important here. *Everyone* knows the President personally. No sir, there's *no* unimportant people in Crystal City. You get the picture?' A more suitable venue for a convention on computer viruses would be hard to imagine. This particular conference was hosted by the *National Computer Security Association*.

Second Chance® body armour (.44 magnum resistant) is the prescribed day-wear for any publisher of unfavourable software reviews; it's a life saver but can become uncomfortable given the typical humidity of America's east coast. Certainly the software developers were in abundance at this particular conference - 'Diagram A illustrates the unrelenting appearance of anti-virus software over a three year period - notice that the curve shows a classic exponential growth rate.' A vision of hell (circa 1994) is an exhibition hall populated exclusively by software developers all intent upon flogging their products to each other. Significantly, the subject of the reception lecture was 'Marketing to the Federal Government'. Thirty companies were demonstrating their wares at this event alone. The homogeneity between their products was depressing but predictable; scanners and checksummers, scanners and checksummers, scanners and...

Standing alone amongst the dark suits was a distinctive young man sporting a blonde pony tail and pebble glasses. According to his lapel badge this was Mark A. Ludwig, author of the *Little Black Book of Computer Viruses*. Knee-capping may be the sentence for troublesome editors, but as far as the anti-virus community was concerned Ludwig deserved nothing less than termination with extreme prejudice. Dr David Stang intimated that only a level of physical intimidation commensurate with that displayed by that other well-known researcher Dr Hannibal Lecter could make Ludwig and his ilk see the light. Mr Ludwig was not entirely without sympathy, however, finding comfort in the company of Mr Patrick Toulme, author of viruses 90 and 101.

Ludwig had been invited to join a panel session to discuss the ethereal proposition that a 'useful' virus *could* exist. The basic freedom to remain uninfected (however 'beneficial' a virus might be) visibly dawned on Mr Ludwig as the discussion progressed. One got the impression that no 'respectable' virus person wanted to stand too near to Mr Ludwig lest a photographer should suddenly appear and compromise the united stance of condemnation. Nobody was clever enough to think of a beneficial virus - suggestions on a postcard please.

Dr Solomon assured everyone that his scanner would go on detecting viruses ad infinitum (*FindVirus* has reportedly just been upgraded to detect an upper limit of 50,000 viruses) and that a virus undetectable by a scanner could *never* be written. His assertion is correct - a scanner which marks all files, clean or otherwise, as infected will detect *any* virus - a fact which is of no comfort whatsoever. For practical purposes, Solomon's confidence may be misplaced, after all he has constantly asserted that in the field of computing nothing is impossible - *except, that is, a virus undetectable by a scanner*? His optimism derives from the reasoning that he always has the last laugh, i.e. he always gets to see the code *after* it has been written. The Mutation Engine, he is quick to cite, succumbed to his enormous charms within 24 hours, and there aren't many virus writers capable of developing such encryption methods. Moreover, hashing and other clever wheezes would ensure that scanner run-times didn't deteriorate while the ongoing development of ever more powerful machines and greater disk capacities meant that run-times would remain constant or even decrease.

While sipping from a cold bottle of *Becks*, a gentleman from the *Mitre Corporation* leaned over and confided: 'You know, some of these software developers are an arrogant bunch - do they really think that their programming skills are any better than those found in most large organisations? We've got guys developing safety-critical and real-time applications which make the technology on display here look positively prehistoric.' Now there's a salutary thought.

A Mr John DeHaven, director of *Bangkok Security Associates* took first prize for ineptitude. After much soul-searching Mr DeHaven concluded that virus-specific scanning does not provide generic protection against unknown viruses. In his quest to prove the undisputed he developed a 'virus factory' which variably encrypts a dropper program and launches a modified version of the Jerusalem virus into memory. Surprise, surprise, no current scanner program (except that provided by *Bangkok Security Associates*) can detect these encrypted dropper programs. As Ken van Wyk of *CERT* observed, Mr DeHaven has discovered how to transform plaintext into cyphertext! Unfortunately, Mr DeHaven's 'virus factory' is now in circulation. Moreover, by tampering with the Jerusalem virus he has effectively created a new specimen necessitating updated disinfection routines in a number of packages. *Bangkok Security Associates* may find the American market somewhat impenetrable; the company's product is called *Victor Charlie* - a phonetic reference to the Viet Cong.

The *United Airlines* stewardess poured a gin and tonic and smiled a perfect smile. Visions of Ludwig and his little black book mingled with the picture of DeHaven and his ludicrous contraption, gradually transforming the Editor's thoughts into a spiralling nightmare of shadowy virus writers, smooth-talking salesmen and over-confident software developers. The vernacular of the virus world with its 'mutation engines' and 'virus factories' suggests an industry teetering on the edge of insanity. It's a mad, mad world.

INDUSTRY WATCH

Certification Confusion

A press release currently in circulation for *Dr. Solomon's Anti-Virus Toolkit* (version 5.56a) carries the headline 'Anti-Virus Toolkit receives highest level of certification'. A certificate accompanies the press release, demonstrating that the product has received a 'UKCVCC Level 1' rating from the *United Kingdom Computer Virus Certification Centre* run by Simon Shepherd, who works at the *University of Bradford*. This certificate has caused some confusion, resulting in a spate of enquiries to *VB* and, we daresay, other knowledgeable bodies.

A little explanation may prove illuminating.

The Official CESG Scheme

The official British certification scheme for evaluating computer security products is run by the *Communications Electronics Security Group* (*CESG*), a division of *Government Communications Headquarters* (*GCHQ*).

CESG has to date certified three anti-virus products to UK Level 1 (which should **not** be confused with Shepherd's UKCVCC Level 1). *CESG* is aware of this other scheme (it has received copies of Shepherd's certificate from various sources) and has made it clear to *VB* that the two schemes are *totally* unrelated.

The Shepherd Scheme

The scheme is operated by Simon Shepherd who was formerly involved in anti-virus product development with a company called *Defiant Systems*. *VB* telephoned Shepherd to find out more. According to Shepherd the *University of Bradford UK Computer Virus Certification Scheme* was instigated 'by Dr Solomon and others' when it was decided that a more independent evaluation facility was needed to test anti-virus products. Shepherd said that four anti-virus products had been evaluated (*Dr Solomon's Anti-Virus Toolkit* and three others, the names of which he could not remember). He also charges fees to companies to verify that master disks intended for duplication are clean.

Information from:

Head of UK CLEF Scheme Certification Body
CESG/GCHQ, Room 2/0805, Fiddler's Green Lane
CHELTENHAM, GLOS GL52 5AJ

Simon Shepherd
Department of Electrical Engineering
University of Bradford
West Yorkshire BD7 1DP

UK NEWS

New Virus - Police Request Information

An alert posted to *CIX* on 23rd June warned of a new virus that had deliberately been sent to a shareware vendor on diskette. The virus contains the text the 'M.S Jurusalem' (sic) but bears no resemblance to the Jerusalem virus. It is not known whether the virus has been distributed more widely.

S&S International, which posted the warning, calls the virus 'MSJ'; it is referred to here as the Palestinian virus. It is a non-resident, non-encrypting parasitic virus which prepends its code to EXE and COM files and has an infective length of 15392 bytes. This excessive size is because the program has been written in a high level language (possibly Pascal) and contains more than one set of library routines.

Text within the virus, which contains a political tirade against Israel and its allies, states that a virus remover will be distributed to computer magazines on October 30th 1992.

The *Computer Crime Unit* is keen to hear from software developers, vendors or individuals in the United Kingdom who have come into contact with this virus. Information can be relayed to DC Noel Bonczoszek. Tel 071 230 1177.

Palestinian

Aliases : MSJ, M.S Jerusalem

Type : Non-resident parasitic file infector, prepending its code to COM and EXE files.

Infective Length : 15392 bytes

Recognition : Plaintext message may be seen in files. ASCII string '99919991999-88888888' is located at beginning of files (from byte 2 onwards).

Detection : Hex pattern will detect this virus:

```
Palestinian E872 F2E8 B7FA E8D0 F0E8 08E5
            3C01 3575 BFF2 3F1E 57BF 8C1C
```

Intercepts : No intercepts except during execution.

Trigger : Between August and December 1992 (inclusive) the virus displays (on a random basis) a screenful of text bewailing the plight of the Palestinians.

Removal : Infected files should be deleted and replaced from master software or clean backups.

WORLDWIDE

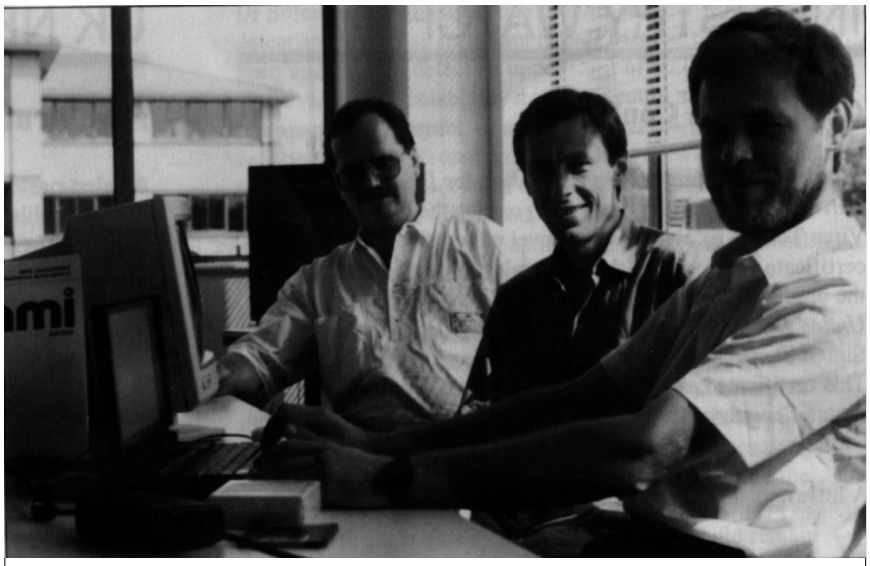
The Russians Are Coming!

To old cold warriors and armchair generals everywhere, the sudden dismemberment of the Soviet Union came as something of a shock. Guaranteed defence budgets and armaments contracts evaporated overnight while western intelligence agencies twiddled their thumbs and sought new directions into which they could channel their energies. The reverberations caused by the August 1991 counter-revolution in the West are as nothing compared to the cultural, political and economic shockwave which has hit the people of the loosely formed *Commonwealth of Independent States*. Finally divesting itself of a command economy, the Russian population is taking its first tentative steps towards capitalism and liberal democracy.

One of the companies which forms the vanguard of Russia's economic resurgence is *KAMI*. Established as a private company in 1989 at the height of Perestroika, the Moscow based company has seen its turnover increase from 1.5 million roubles to 9 billion roubles in just three years of trading. The company is unashamedly 'high-tech'; its computer systems division and *Center of Parallel Systems and Technologies* has invested in the manufacture, sale and export of PCs, software, computer networks, telecommunications and even supercomputers.

Eugeny Kaspersky

In charge of *KAMI's* team of 18 programmers is Eugeny Kaspersky, author of '*-V*', which is the most powerful commercial anti-virus software program developed in the CIS. This software has gained a formidable reputation in its country of origin and is the first Russian anti-virus product to have gained a commercial toehold in the West with distributors established in the United Kingdom and Germany. In June, Kaspersky and two of his colleagues visited the United Kingdom to discuss co-operation with British computer virus specialists, thus providing VB with a rare insight into viral developments in mother Russia.



Vorsprung durch Perestroika: Russian technologists Dr. Viktor Lopatin and Dr. Alexei Remizov, with virus expert Eugeny Kaspersky (right), visit *Virus Bulletin.*

Endemic Problems

According to Kaspersky, the computer virus problem within Russia is endemic; he claims to receive between two and three new virus samples daily and that nearly all of these samples come from the wild. 'We have tens of thousands of unemployed programmers, even when employed a gifted programmer in Russia earns only 100th the salary of his Western equivalent.' Scarce employment opportunities, boredom, and seething resentment have combined to create a climate in which virus writing is the principal programming activity amongst the young and disenchanted.

Most of the viruses are primitive and, ironically, the most virulent specimens in Russia (New Zealand II and Michelangelo) are not 'home grown'. Kaspersky warns, however, that in recent months a new breed of viruses has been developed which employ encryption, error-correcting (Hamming) code and full-stealth features. 'We very rarely see global infections - that is the simultaneous outbreak of a virus at numerous sites. The last virus to do this was DIR II which spread throughout Moscow very rapidly. Since then we have seen only localised outbreaks caused by a variety of different specimens.'

Anti-Virus Software

The most widely used anti-virus software in Russia is Lozinsky's *AIDSTEST*, a shareware program, closely followed by McAfee's *SCAN* - both programs have been systematically targeted by the virus writers.

Kaspersky and his friend and associate Lozinsky (the two Muscovites live within walking distance of each other) represent the Russian research effort, channelling virus samples and timely intelligence to the major researchers in Europe including Bontchev at the *University of Hamburg*, Kadlov in Poland and Skulason in Iceland. According to Kaspersky, Lozinsky

regards the maintenance of *AIDSTEST* as a hobby whereas *KAMI's* anti-virus programme is professionally directed comprising regular software upgrades, full telephone, fax and e-mail support. The complete anti-virus service from the company costs US$90 per annum.

Kaspersky has personally disassembled 600 computer virus samples and claims that he can recognise code similarities between any two samples at a glance. He is one of that rare breed of researcher who can recite the 8088 instruction set in hexadecimal!

There are two million PCs in Russia and the CIS, the majority of which were imported from manufacturers in Europe, the United States and the Far East. The most popular pirated XT clone is called the ES 1842; Kaspersky has seen a number of virus specimens which do not work on his own 'true blue' research machine but which happily replicate on these indigenous clones.

Permissive Exchange

Software theft is perhaps the most immediate problem facing the burgeoning Russian software industry. Kaspersky estimates that for each legitimate copy of *-V* (*KAMI* has sold 800 sets to date) there are at least 100 illegal copies.

A vicious circle has developed in which permissive software theft and exchange contributes to the spread and circulation of virus code. To prevent software theft, his team developed *SUPER GUARD*, a commercial program to protect proprietary software on diskette using a hidden authorised installation counter.

With the complete absence of any legal redress against software piracy in the CIS, and with the endemic virus threat worsening perceptibly, Kaspersky's team has devoted the larger proportion of its efforts towards software security applications.

Fears of a 'Brain Drain'

One of the threats to Russian economic aspirations is the threat of a 'brain drain' as qualified specialists seek remuneration in the prosperous nations of the West. The best Russian programmers are already being offered lucrative jobs; Kaspersky cites *AutoLISP* and *Mathematica* as examples of Western programs developed with substantial Russian input.

KAMI as an organisation is keen to prevent the emigration of its skilled people and has adopted a strategy of forging research and trading links with high-tech companies thoughout the world.

With Russian viruses appearing at a rate of approximately 100 per month and with many Western software companies already floundering in the face of the virus onslaught worldwide, Kaspersky and his team of disassemblers should be busy for some time to come.

VIRUS ALERT 1

Roger Riordan
Cybec Pty Ltd, Australia

Troi Two

On May 29th, *Cybec* received a sample of a parasitic virus adding 512 bytes to EXE files. It contains the words TROI TWO and is apparently related to the Troi virus mentioned in the May edition of *Virus Bulletin* and first described (to my knowledge) by Paul Evans (pevans@jarthur.clairmont.edu) on *Virus-L* on 11 March 1992.

Delayed Replication

When an infected file is run the virus first checks the date. Until May 1st 1992 it simply ran the original program. Now it issues an Are You There? call (Int 21H, function FCH). If the reply is AH=55 the virus allows processing to continue normally. Otherwise it copies itself to the second half of the interrupt table, overwriting the vectors for interrupts 80H through E5H and hooks INT 21H.

The INT 21H handler traps function 4BH (LOAD & EXECUTE). If the file extension is .EXE, the virus reads the file header into a work area on the stack. If the file's checksum is 'T2' the file is assumed to be infected. Otherwise, the virus adjusts the file header, and copies itself to the end of the file, adding 512 bytes. It clears the attributes and restores the file date, but does not trap critical errors.

Unpredictable Behaviour

There is no warhead and the text is never displayed. The virus is extremely unreliable. Our source said that it ran (and propagated) happily under DOS 4, but crashed workstations using DOS 5. We could run an infected file on two of our three test PCs (all running under DOS 3) but the PC crashed as soon as we ran an uninfected file. We sometimes managed to infect one or two files before the PC crashed on the third.

The interrupts overwritten do not appear to be used on any of our PCs and there are no glaring bugs in the logic, but the virus does push approximately 60 bytes onto the stack. This probably causes the crashes by overloading the stack. The virus does no deliberate damage, and is too obvious to present much of a threat. The delayed activation suggests that the virus writer intended to attach the virus to some popular software, in the hope that it would be widely distributed before it began to replicate.

The virus can be detected using the following search pattern:

```
80FC FC75 04B4 559D CF50 5351 5256 571E 0683
EC28
```

VIRUS ALERT 2

Fridrik Skulason
FRISK Software, Iceland

V-Sign - A Polymorphic Boot Sector Virus

It seems easier for a boot sector virus to spread around the globe than for a program virus, and V-Sign is a clear example of this. It appeared first in Turkey, where it is known as 'Cansu', but has recently been reported in the UK, and in June the first infections were reported in the United States. The probable explanation for this rapid circulation is that the virus has infected commercial software or hardware which has been distributed unwittingly.

V-Sign Structure

V-Sign is unusual in many ways. Functionally it has a slight similarity to the New Zealand (Stoned) virus. It infects the Master Boot Sector of hard disks, storing the rest of itself elsewhere on Track 0, Sectors 4 and 5 which are usually unused, as well as DOS boot sectors of diskettes (logical sector 0), using the last two sectors of the root directory to store the remainder of its code.

The virus does not store the original boot sector anywhere, but instead it stores 38 bytes of the boot sector (Master Boot Sector or DOS boot sector on diskettes) within its own code and then overwrites them.

The most significant aspect of the virus is that this 38-byte code is polymorphic, although it can be detected with a search string containing wildcards, as the polymorphism only involves moving a few MOV instructions around:

A typical instance of the code is displayed below, where the word 9876H at the end is used to mark diskettes as infected.

```
            XOR     AX,AX
            MOV     SS,AX
            MOV     DS,AX
            MOV     ES,AX
            DEC     AX
            MOV     SP,AX
label_1:    XOR     AH,AH
            INT     13H
            JC      label_1
            MOV     BX,7E00H
            MOV     AX,0202
            MOV     CX,startsector
            MOV     DX,0100H
            INT     13H
            JC      label_1
            JMP     continue
            DW      9876H
```

Installation

When an attempt is made to boot a computer from an infected diskette, the virus reads the two sectors which contain the rest of the virus code into memory. It then allocates memory by reducing the value stored at 40H:13H. It hooks INT 13H and modifies the boot sector image in memory by restoring the 38 bytes that had been overwritten and transfers control to the 'original' boot sector.

INT 13H Handler

The INT 13H handler intercepts read and write operations (AH=2 and AH=3). Boot sectors are checked for an existing infection, using the 9876H marker. If no infection is found, the virus infects the boot sector and increments a counter. When the lowest six bits of this counter are all zero, (that is, when the counter has reached 64), the virus displays a 'V' shaped sign on the screen, using block graphics and hangs the computer. Unlike New Zealand, this virus does not have problems infecting high density disks or 3.5 inch diskettes.

Removal

Disinfection must be undertaken in a clean DOS environment, i.e. having booted from a write-protected clean system diskette. The code in the boot sector must be used to determine where the rest of the virus is stored, the 38 bytes of genuine boot sector code read from that sector and written back to the boot sector. Alternatively, the FDISK /MBR command (DOS 5 only) can be used to restore the Master Boot Sector. The recommended approach to disinfect diskettes is to transfer any data using the DOS COPY command and then format the infected floppy. Alternatively, the SYS command can be used to destroy the virus.

V-Sign

Name: V-Sign

Aliases: Cansu

Type: Master Boot Sector infector (Track 0, Head 0, Sector 1) which stores the remainder of its code at Sectors 4 and 5. Infects the DOS Boot Sector (logical sector 0) on diskettes.

Infective length : 3 sectors

Intercepts : INT 13H for infection

Trigger : Displays a V-shaped sign in block graphics when the virus has infected 64 diskettes.

Boot sector recognition : Wildcard hex pattern will detect this virus.

```
V-Sign  1372 FA?? ???? ???? ???? ???? ????
        ??CD 1372 EAE9 A601 7698
```

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 25th June 1992. Entries consist 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 using the 'search' routine of a disk utility, or preferably a dedicated scanner which contains an updatable pattern library.

Type Codes

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

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

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

Seen Viruses

2638 (temporary name) - CER: Awaiting analysis.

```
2638            8B46 F4A3 0400 8B46 F6A3 0600 8B46 EEA3 0800 8B46 F0A3 0A00
```

BloodLust - CN: One of many primitive, overwriting viruses discovered this month. This 302 byte virus contains the following text: 'Hi! This is the virus BloodLust striking! Sorry to tell you, but your system is infected.'

```
BloodLust       32C3 AAE2 FA2E 833E 0F01 0074 29B4 402E 8B1E 0F01 2EFF 360F
```

Breeder - CER: This is a 5152 byte companion virus, which creates COM files corresponding to EXE files. It will also add a 172 byte Trojan to COM files which contains an encrypted message: 'I greet you user. I am COM-CHILD, son of The Breeder Virus. Look out for the RENAME-PROBLEM !' This virus has been reported elsewhere as 'Shield', but there does not seem to be any reasonable explanation for that name.

```
Breeder-Trojan  B404 CD1A 7221 80FE 0275 1CB4 2CCD 2183 FA3C 7F13 8D76 07FC
Breeder         8D36 1F01 8BFE 8D16 1F01 8D0E 2F0E 2BCA FCAC D0C8 AAE2 FAE9
```

Dark Avenger-1687, Pp!ko-B - CER: Detected with the Dark Avenger pattern. Awaiting analysis.

Datalock-1043 - C(E)R: This virus will append itself to EXE files, but does not seem to infect them properly. Detected with the Datalock pattern.

Ear-6 - CEN: This virus is extremely obvious, as it activates frequently and will then ask the user questions about the anatomy of the ear. The virus is highly unlikely to spread in its current form. The virus is 1024 bytes long, and encrypted. As the decryption routine is very short, only a partial search string is possible.

```
Ear-6           BB?? ??B9 ???? 2E81 37?? ??83 C302 E2F6
```

Eddie 2-B, **Eddie 2-C** - CER: These two viruses are very similar to the original virus, but assembled with a different assembler. In the 'C' variant, two instructions have been swapped, which invalidates the earlier Eddie 2 pattern.

```
Eddie 2-C       D3E8 408C D103 C18C D949 BF02 008E C1BA 2B00 8B0D 29D1 3BC8
```

Fichv EXE 1.0 - ER: This virus is clearly related to the Fichv virus, but it is structurally different, as it infects EXE files rather than COM files. Virus size is 897 bytes.

```
Fichv EXE 1.0   8CDB 8EC3 83C3 100E 582D 1000 8ED8 019C 2001 FFB4 2001 FFB4
```

Haifa-Mozkin - CER: A polymorphic virus, around 2360 bytes in length, but variable. No search pattern is possible.

Joe's Demise - CER: The name has absolutely nothing to do with the operation of the virus. It is 981 bytes in length on COM files or 1009 bytes on EXE files long. Awaiting analysis.

```
Joe's Demise    B802 3DCD 218B D81E 8CC8 8ED8 B800 57CD 2151 5280 3C65 740C
```

Keyboard Bug, 1596 - CER: This virus does not always work, but it is related to an earlier variant reported in February 1991.

```
Keyboard bug    1E53 2EFF B51B 07BB 0806 B912 0158 2E30 0143 E2FA 5B1F E8
```

Leprosy-Scribble - EN: Yet another variant of this primitive overwriting virus. This one is 595 bytes long. Also discovered this month is a variant closely related to the original and 666 bytes long.

```
Scribble        59EB 005E 5DC3 558B ECA1 0C03 051E 008B D033 C9B0 01B4 43CD
Leprosy-B2      59EB 005E 5DC3 558B ECA1 0403 051E 008B D033 C9B0 01B4 43CD
```

Plaice-1273 - CR: Detected with the previously published pattern for the Plaice virus.

Russian Tiny - CR: Two viruses, 131 and 145 bytes long which are among the shortest resident samples. The 131 byte virus infects COM files when they are copied while the 145 byte variant infects on execution. The viruses do not seem to have any side effects.

```
Tiny-132        03FF 2683 3D00 7516 B980 00F3 A4BE 8400 26A5 26A5 26C7 44FC
Tiny-145        80FC 4B75 593C CC75 0558 57F3 A4CF 5053 521E B802 3DCD 2172
```

Screaming Fist II-696 - CER: This 696 byte virus is slightly polymorphic and the decryption routine is rather short. Only a partial search pattern with wildcards is possible. Awaiting analysis.

```
Scr Fist II-696  5D8B F556 B0?? B9A3 02?? 2E30 0446 E2F9 C3
```

Sistor-1000 - CR: Possibly an earlier version of the two variants reported last February but shorter at 1000 bytes and without apparent side effects.

```
Sistor-1000     FF00 8916 8400 8C06 8600 FB33 C08E D8B8 4953 A340 032E 80BC
```

Starship - CER: This virus is not new, but it has not been included before, because of its seeming inability to replicate. However, if it is run on an IBM XT running DOS 3.xx, with a colour adapter, it will infect the Master Boot Sector. When the machine is subsequently rebooted it will sometimes infect files, as they are executed. The size of the virus is variable and as it is polymorphic, no simple search pattern is possible.

SVC 5.0 B - CER: Very similar to the original SVC 5.0 virus, but with some minor patches, possibly intended to bypass anti-virus software.

```
SVC 5.0 B       5606 86C0 25FF FF8E C00E 1F33 FFB9 990B FCF3 A607 5E74 03E9
```

Swedish Boys-Why Windows - CN: A 459 byte virus from Sweden. It may delete the file \WINDOWS\WIN.COM, if found, and it is particularly annoying in late February. On the 23rd of that month it may delete AUTOEXEC.BAT, on the 24th it deletes CONFIG.SYS, and on the 25th it trashes the contents of the root directory and the FAT.

```
Why  Windows    83C2 06CD 21B4 4E8D 9403 01B9 0600 CD21 3D12 0074 548D 940A
```

Swedish Boys-Data Molester - CN: This 538 byte virus is closely related to the Why Windows virus, but it is slightly more complicated, includes extra features and encryption. As the name indicates this is a destructive virus, but the damage is very obvious and easily detected, as it trashes drive C: by overwriting the root directory and the FAT.

```
Data  Molester  BB01 018A 27BB 0201 8A07 86C4 0503 008B F08A 8C03 01E8 E401
```

Swedish Boys-Headache - CN: Yet another virus by the same authors as the previous two viruses. 457 Bytes long. Awaiting analysis.

```
Headache        BB01 018A 27BB 0201 8A07 86C4 8BF0 B41A 8D94 C802 CD21 33C9
```

Trivial-45B - CN: This overwriting virus is closely related to a 46 byte variant reported earlier, but one byte shorter. Also discovered this month are two Swedish variants, 31 and 35 bytes.

```
Trivial-45B     BA9E 00CD 2172 0F93 BA00 01B4 40B9 2D00 CD21 B43E CD21 B44F
Trivial-31      B802 3DBA 9E00 CD21 93B4 4083 C262 B11F CD21 C32A 2E43 2A00
Trivial-35      9E00 CD21 93BA 0001 B440 B123 CD21 B43E CD21 C32A 2E43 2A00
```

Trivial-Banana - CN: A primitive, 139 byte overwriting virus, containing the text: 'BANANA, coded by Morbid Angel-92 in Stockholm/Sweden'. Infected files must be deleted.

```
Trivial-Banana  59B8 0157 CD21 B43E CD21 59BA 9E00 B801 43CD 21B4 4FEB B7C3
```

Troi II - ER: This virus is different from the Troi virus - it infects EXE files instead of COM files, so it is considered to belong to a separate family. However, the two viruses are probably written by the same author. No effects other than replication have been found.

```
Troi II         2D2D 3EFC 0E1F 2BF6 8EC6 BF00 02B9 9801 F3A4 061F A184 00A3
```

Vengence (sic) - CN: A group of primitive overwriting Swedish viruses, with sizes ranging from 194 to 656 bytes. Note that the signatures for variants C-F can be combined easily with the use of wildcards.

```
Vengence-A      BA2D 01B4 4ECD 2172 22BA 9E00 B802 3DCD 2172 1893 53B1 C283
Vengence-B      BA68 01B4 4ECD 2172 59BA 9E00 8916 0202 B802 3DCD 2172 45A3
Vengence-C      B800 C9BD 0000 CD2F 3CFF 750B 9090 9083 FD13 7603 E989 00B8
Vengence-D      B800 C9BD 0000 CD2F 3CFF 750B 9090 9083 FD13 7603 E9AC 00B8
Vengence-E      B800 C9BD 0000 CD2F 3CFF 750B 9090 9083 FD13 7603 E977 01B8
Vengence-F      B800 C9BD 0000 CD2F 3CFF 750B 9090 9083 FD13 7603 E972 01B8
```

Yankee-Micropox - CER: A 4920 byte version of the Yankee virus, which is detected with the Yankee pattern. Reported to overwrite the hard disk in March. Awaiting analysis.

VIRUS ANALYSIS 1

Jim Bates

The Jabberwock Virus

This virus has an unusual history - late in 1991 it was reported as having been uploaded to a number of bulletin boards in the UK. On at least one of these boards, the SysOp kept an accurate log of just who had uploaded what. The information he had was passed to the *Computer Crime Unit* at New Scotland Yard, which began enquiries. The details of the investigation are not available for publication but the police were completely satisfied that they had identified the true source of the virus. Shortly afterwards, the following letter, together with two files of source code, was received at the premises of anti-virus vendor *S&S Ltd*. The files read:

```
Dear Sir,

I regret that I have to inform you that I am the
author of the JABBERWOCK virus.

I originally wrote the code last year as an
experiment to see if I could do it. I had heard
of many viruses but had never experienced or
examined one myself. I felt that it was all media
hype and that they did not really exist. This was
version 1 which only attaches itself to COM
files. As far as I know this one was never
distributed.

My curiosity got the better of me and I improved
the code to attach itself to simple EXE files and
uploaded it to some bulletin boards as an
experiment to see how quickly it spread. This is
version 2.

I now regret doing this greatly. It was a very
stupid thing to do.

As I guess you already know, the virus is not
malicious. The thought never crossed my mind to
make it deliberately damaging. I have no
intention of any further development or spreading
of the virus.

An infected program can be identified by the
letters 'JW' in offset 3&4. It can be detected in
memory as follows:

    MOV   AL,'J'
    MOV   AH,4BH
    INT   21H
    CMP   AL,'W'
    JZ    MEMRES

I have included the source code of JABBER1 and
JABBER2 for your information. I have also
included a program called DEJABBER which will
remove the infection (without damaging the files)
from the disk it was run from, but not from
memory. If used with a /L parameter the program
will report on infected files without removing
it. Please use, modify and distribute it in
anyway you see fit.

Please also pass on my sincere apologies to
anybody who is worried by this episode.

Sincerly(sic)

Anon
```

Version 1 of the Jabberwock virus has recently been reported at large and this prompted a more detailed examination of the technical aspects of the matter.

Firstly, the letter suggests that version 1 was never distributed. So we must conclude that the writer is either a liar or was telling the truth as he saw it. If he wasn't lying, then someone else was responsible for the emergence of this virus since the letter was written! The police are continuing investigations to try to identify the source of this new outbreak. Since the source code is now widely available throughout the anti-virus research community (which is not renowned for its security consciousness) there is a distinct possibility that some fringe 'researcher' may have been responsible.

The second point of interest concerns the source code itself. I began this whole analysis by disassembling a copy of the specimen found at large. Once I had this in a source code form, I compared it with the two source files mentioned in the letter above. Version 1 was an almost exact match but oddly, both of the author's source code listings could not be assembled in the form in which they were received. This was not noted in the original publicity about this virus and, to my knowledge has not been mentioned since.

The listings, while they looked like ordinary source code written for assembly under *Borland's Turbo Assembler*, actually contained some constructs and instructions which were simply illegal. This was almost as if the file had been written as an example or guide and used by others who were less knowledgeable. In the actual live example the constructs had been changed to achieve the same results by different methods and the virus functioned as it was supposed to do.

The virus itself is unremarkable in either version. It attempts to install itself as a legal TSR using INT 27H and once resident, it intercepts INT 21H, function 4BH (LOAD & EXECUTE).

Version 1 infects only COM files and makes no check of their length. This may result in a system malfunction if the virus attempts to infect files near the 64k limit.

Version 2 will infect EXE files also, but here there is a check to prevent infection of EXE files over 65024 bytes in length, or those which allocate memory beyond their normal file image length.

The same trigger routine is found in both versions and this consists of sounding a beep and displaying a message at intervals determined by a counter. Version 2 displays the message more often. The message is:

```
BEWARE THE JABBERWOCK!
```

This, together with a further text message which is not displayed, is encrypted within the virus code during infection and will not be seen during simple file examination. The additional message gives further food for thought when considering the true original motives of the writer:

```
HI SOLLY. THIS IS A 100% BRITISH PRODUCT.
```

Conclusions

This incident adds further fuel to the arguments about so-called 'benign' virus code. The end result is quite simply two more primitive viruses which have to be added to the growing list of those known to be at large. Specific detection software has had to be updated, users informed and yet more time taken up in the constant search for these nasty little pieces of code.

JABBER WOCK

Name: JABBERWOCK

Aliases: none known

Type: Resident Parasitic Virus

Infection: Version 1 infects COM files only

Version 2 also infects EXE files below 65024 bytes in length.

Infective length: Version 1 = 615 bytes

Version 2 = 813 bytes (on both COM and EXE files)

File Recognition: Hex pattern will identify both versions on disk or in memory.

```
Jabberwock 0500 108E C0BE 0000 BF00 00B9 FFFF
           F3A4 1E07 89D6 BF00 01B9
```

System Recognition: 'Are you there?' call involves placing a value of 4B4AH into AX and issuing an INT 21H call. If the virus is resident, AX returns with a value of 57H in the AL portion of the register.

Intercepts: INT 21H function 4BH (LOAD & EXECUTE)

Trigger: On a counting basis, displays a message. Version 2 displays more often.

Removal: Specific and Generic disinfection is possible under clean system conditions. Recommend deleting affected files and replacing them with clean copies.

VIRUS ANALYSIS 2

NoInt Virus

New Zealand (Stoned), Form and Joshi are the most common viruses but some reports have been received recently of incidences of the NoInt virus. In December 1991, NoInt gained a certain notoriety when *Novell* accidentally shipped its *NetWare Encyclopedia* contaminated with the virus (see *VB*, January 1992, p.2). The virus is also known as Stoned III, presumably because it is a derivative of the New Zealand (Stoned) virus.

NoInt was first reported in Canada about a year ago. It is essentially a variant of the New Zealand virus and functions in a similar manner. There is no trigger routine, destructive or otherwise and therefore no messages are displayed. There is no effective error-trapping within the virus code. On some diskettes the boot sector may become unreadable and errors may be reported by software which attempts to access it.

Installation

As with all boot sector viruses which infect the Master Boot Sector, the code is loaded immediately after the Power On Self Test (POST) routines have completed. Execution first initialises the Data and Stack segment registers and then collects the address of the Disk I/O service routine (which at this time will be that held in ROM). The pointer to the machine's base memory size is then decreased by 2K and the memory thus released is addressed in order that the virus code can be moved up into it.

Pointers to the virus' own Disk I/O interception routine are then installed into the Interrupt Table. This section of the code finally loads the original boot sector from the relevant place on the disk into the boot area and passes control to it so that the normal boot routine may continue.

Operation

Once installed, the virus interrupt service routine monitors read requests and does not intercept any other functions. Any read request is checked to see whether Head 0 of the first fixed disk is being accessed.

With fixed disk reads, a further check is made to see whether the request is for sector 1 (the Master Boot Sector) and if so, the addressing is changed to read sector 7 of side zero (exactly as in the New Zealand virus). This is a classic semi-stealth tactic to avoid detection.

Paradoxically, the virus does not trap writes to the boot sector on the hard disk. Therefore changes can be made to the infected boot sector, although verification of any change will return the original boot sector

Requests to read floppy disks result in a check to ensure that drive A or B is being accessed (otherwise processing jumps directly to the system interrupt handler) before the infection routine is called.

Infection

Infection commences with the virus reading absolute sector 1 from the default disk (fixed or floppy) and then checking to see whether it is already infected. This check compares the word at offset 0D6H in the virus code, with the word at the same offset in the sector just read. If the two words match, the disk is assumed to be infected and processing returns to the calling routine.

Otherwise, 59 bytes from offset 3 of the newly read sector are copied into a similar position within the virus code. This segment includes the OEM system name and the whole of the disk parameter area for floppy disks. Subsequently, 66 bytes are copied from the new sector to the virus code starting at offset 1BEH (this includes the whole of the Partition Table on fixed disks).

The original boot sector is then written back to a pre-arranged sector of the target disk. If the target is a fixed disk the address will be Track 0, Head 0, Sector 7. If the target is a floppy disk (any density) the address will be Track 0, Head 1, Sector 3. On floppy disks this sector is amongst those allocated to hold the root directory so any file name entries stored there will be lost.

Once the original boot sector has been moved, the modified virus code is written back to absolute sector 1 and the infection process is complete.

One small addition to the design ensures that the Trap flag is cleared before the interrupt request is completed. This may be a half-hearted attempt to nullify certain types of anti-virus software. It is worth noting that if this virus is resident, most software debuggers will hang if an attempt is made to read absolute sector 1 of a disk.

Removal

The virus can be removed from a fixed disk as follows: boot the machine from a known clean system disk and simply copy Track 0, Head 0, Sector 7 to Track 0, Head 0, Sector 1. This can be done using *The Norton Utilities* or any suitable disk editing utility. On machines running DOS 5.xx, the FDISK / MBR option may be used following a cold system boot.

Diskettes should be disinfected by copying files from the infected diskette using the DOS *COPY* command. This should be done in a clean DOS environment. Do not use DOS *DISKCOPY* as this is an image copier and will transfer the entire contents of the disk including the infected boot sector. Once files have been transferred, the diskette should be formatted.

As with all viruses, it is important that all instances of the virus be eradicated and this means that all diskettes associated with the infected machine will need checking.

Conclusions

This virus is unremarkable. It is obviously a development of the New Zealand virus and will cause some problems as a result of the poor coding. The blinkered vision of the virus writer is revealed by his determination to intercept read requests but apparent oversight in failing to intercept write requests to the Master Boot Sector.

Since there is no visual display, this virus is less noticeable than New Zealand, particularly to those users who do not use any anti-virus software; these users are thus far more likely to transmit infected diskettes. However, NoInt is easy to detect and easy to remove.

NoInt

Name: NoInt

Aliases: Bloomington, LastDirSect, Stoned III

Type: Boot sector virus infecting the Master Boot Sector on fixed disks. (Track 0, Head 0, Sector 1). The original boot sector is moved to Track 0, Head 0, Sector 7. The virus infects logical sector 0 of diskettes (all densities) and moves the original boot sector to Track 0, Head 1, Sector 3.

Infection: Fixed disks and diskettes of any format within drives A: or B:

Recognition: Hex pattern will identify this virus on disk or in memory.

```
NoInt  81FA  8000  7529  83F9  0175  2451  B907
       00B8  0102  9C2E
```

Intercepts: INT 13H - the ROM BIOS disk services and intercepts subfunction 02H, the read sector request. The virus thus redirects read requests for the boot sector. The virus does not intercept write requests.

Trigger: None

Removal: Specific and generic disinfection is possible under clean system conditions.

VIRUS ANALYSIS 3

Datalock

The Datalock virus was first reported in the United States during late 1990 and several incorrect reports of its effects have circulated from time to time. This virus has been around for some time but only recently have reports been received about it in Europe. Is this coincidental, or has someone deliberately unearthed it and given it new life? Whatever the answer, it's a nasty one but mercifully easy to detect and eradicate.

Installation

Datalock is a memory-resident parasitic virus which appends its 920 bytes of code to both COM and EXE files (including COMMAND.COM). Installation occurs the first time that an infected file is executed and the code begins by calculating its own position in memory in order to have a reference to its internal data area.

The virus first checks whether its host is an EXE or COM file. Datalock checks only for the 'MZ' signature in a file header and ignores the possible 'ZM'. The reason for the check is simply so that the virus can correctly repair the program header before passing control to it. However, before this happens, the code issues an 'Are you there?' call by placing 0BEH into AH and executing an INT 21H request. If the virus is resident, the call returns with 1234H in the AX register. Note that the values 0BEH and 0BFH (used later) are also used in this manner by some versions of *Novell NetWare* and this virus will certainly cause network malfunction with those versions. It must be stressed that normal software security will stop this virus spreading across a network.

If there is a 'Yes I am' answer to the 'Are you there?' call, control is returned to the host program. Otherwise, processing branches accordingly and the installation routine is executed. This begins by modifying various memory pointers in order to make 2048 bytes of conventional memory available for the virus to inhabit. The code is located into its new home and the DOS service interrupt INT 21H is hooked in the usual way using GETVECTOR and SETVECTOR calls.

Just before control is passed to the host program, a special call is issued by placing 0BFH into AH, pointing DS:DX at offset 8 in the environment segment and calling INT 21H.

On all compatible machines tested, offset 8 in the environment segment contained the ASCIIZ specification of the COMSPEC program file. This is usually COMMAND.COM but some primitive protection systems rely upon this being a different name. The 0BFH value instructs the virus code to attempt to infect the indicated file without actually executing the code that the file contains. Thus it becomes obvious that this virus attempts to infect COMMAND.COM (or its equivalent) as soon as it becomes resident!

When resident at the top of conventional memory, this virus reduces the overall system memory by 2048 bytes and the code will be found at offset 125H of the top segment (the previous bytes being used as a data area).

Operation

The INT 21H handling routine in the virus code maintains a flag byte to signal when it is being used by virus functions. This allows the virus itself to use INT 21H directly without the risk of going into an infinite loop, thus making INT 21H 're-entrant' in a limited form. This busy flag is set, reset and checked regularly within the routines. There are four functions intercepted by the virus' handler - two of which have already been mentioned (0BEH and 0BFH) and it should be noted that the 0BFH call will return with a value of 0BEH in the AX register (thus confusing Novell further.). The two system calls intercepted are 4B00H (LOAD and EXECUTE) and 3DH (OPEN file). The process is essentially the same for either call, the caller's registers are saved and a temporary INT 24H service routine is installed to avoid error displays on screen.

The attributes of the target file are collected and examined. A Read Only attribute is modified to allow access and the file is opened. The date and time stamps of the file are collected and stored and the first 32 bytes of the file are read into memory and checked to see whether the file is of EXE or COM type and whether it is already infected. The infection test differs between EXE and COM files.

With the header of an EXE file, the word value at offset 16H (the CS field) is added to the value at offset 14H (the IP field) and a value of 1234H is added to the result. This is then checked against the word value at offset 12H (the ChkSum field) and if they match the file is deemed already infected and is not molested further.

With a COM file, the values of the second and third bytes in the file are added together and compared to the fourth byte. If they match, the file is deemed to be infected.

This self recognition method works well as a pre-emptive check for the virus but it is less than satisfactory for virus detection purposes since it is obvious that clean files can exist which match the above criteria.

Infection

The actual infection process consists of modifying the header of the target file and then appending the virus code. The true length of the file is used, not the perceived length stored within the EXE header. One unusual aspect of this virus which may have given rise to erroneous reports that it does not infect COM files, is that EXE files of any length are infected but only COM files in excess of 23000 bytes are attacked.

In all cases, the infective length is 920 bytes and the plain text message 'Datalock version 1.00' can be seen at the end of the virus code. A 1043 byte of this virus has also been reported.

Trigger

This virus contains a nasty trigger routine which will cause various errors to occur if a program tries to access .DBF files.

The routine checks for a system date later than July 1990. During the interception of function 4B00H or 3DH requests, the target file is checked to see whether it has a three letter extension, the last two letters of which are 'BF'. This includes the .DBF format used by the *Borland* (formerly *Ashton Tate*) *dBase* program. If the file extension contains these letters, the segment portion of the return address as stored on the stack is modified in a way that will cause random errors when the routine returns to the calling program. A common error reported during *dBase* operations is 'out of file handles'.

Datalock

Name: Datalock

Aliases: V920, Datalock 1.00

Type: Resident Parasitic virus

Infection: Any EXE file and COM files over 23000 bytes in length (including COMMAND.COM)

Infective Length: 920 bytes on both EXE and COM infections.

'RU there?' Virus places a value of 0BEH into AH and executes an INT 21H request. If the virus is resident, the call returns with 1234H in the AX register.

File Recognition: Hex pattern will identify this virus on disk or in memory.

```
Datalock C31E A12C 0050 8CD8 488E D881 2E03
         0080
```

Intercepts: INT 21H OPEN file and LOAD & EXECUTE requests. Also installs temporary dummy INT 24H handler.

Trigger: If system date is after July 1990, attempting to open or execute a file with an extension of .?BF will result in unpredictable errors.

Removal: Specific and generic disinfection is possible under clean system conditions. Recommend deleting affected files and replacing with clean copies (note that COMMAND.COM will almost certainly be infected).

PRODUCT REVIEW 1

Mark Hamilton

Certus Novi

Novi from *Certus International* has recently been introduced to the UK anti-virus market. The product reviewed here is version 1.0.1 (serial number 303098) supplied by UK distributor *Guildsoft* of Plymouth.

This product was tested in last month's comparative scanner test (see *VB*, June 1992, p.16). A subsequent letter from *Certus* President Dr Peter Tippett asserted that this version was in fact a beta-test version and should not, therefore, have been reviewed. However, the copy which *VB* reviewed was shipping product obtained from a commercial source.

Note that the scanner results recorded here are for the very latest shipping product (1.1 with files dated April 1st 1992, serial number 350490N) supplied by *Certus* directly.

Version 1.0.1

Novi is delivered on both 5.25-inch and 3.5-inch write-protected diskettes and, when installed, the product consists of a memory-resident monitor (*NOVITSR*), an integrity checker (*NOVIBOOT*) and the core program (*NOVI*) which combines virus-specific and generic detection modules with a configuration utility for all aspects of the package.

All the programs are compressed into an installation program on the delivery diskettes. The various programs can be passed operational commands and options contained in parameter files and there are some 13 examples included on disk.

Using the configuration capabilities of the main *Novi* program you can tailor the operational capabilities of the TSR and the transient scanner/checker. For example, you can enable or disable generic checking - which *Certus* calls 'Perfect Checking'; whether or not boot sectors are scanned; whether or not files should be automatically repaired etc.

For each menu item, a multiline help panel appears at the bottom of the screen which provides a little more information on that menu option. I liked the configuration set-up and found it quick and easy to use.

One interesting feautre is that you can create a further installable copy of *Novi* - all the components recompressed into INSTALL.EXE which can then be copied to a diskette. This facility is obviously to be used by site licensees so that they can set all the default parameters and give each user one simple installation program to run. This is an excellent facility which will doubtless be much appreciated by overburdened corporate PC staff.

Scanning Performance (NOVI 1.1)

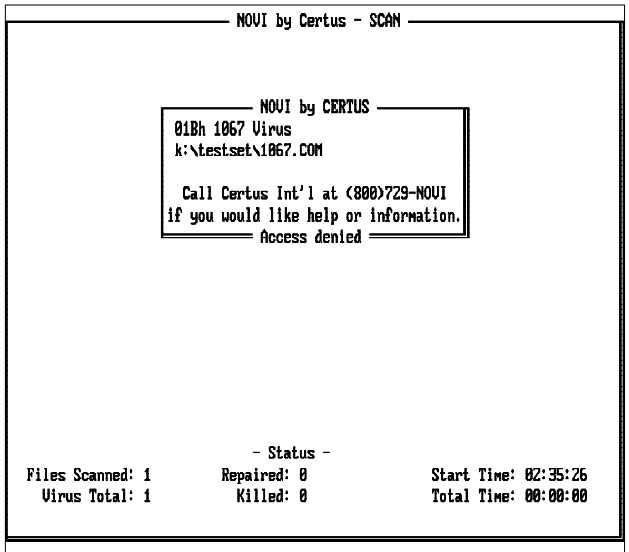
The following results are for NOVI 1.1 (see introductory paragraph). The virus test sets used have been published in previous editions of *VB* and readers are directed to them accordingly.

Novi's manual claims very fast disk scanning speeds - a 40 megabyte hard drive can be scanned in under 30 seconds. That isn't too far from the truth, my 'standard hard disk' (36 Mbytes of data of which some 11 Mbytes are executable) was scanned in just under one minute. *Novi* seems to use the 'scalpel' approach to virus detection: it knows where to look for viruses and just checks those places. This gives it an impressive scanning speed.

Using my standard test set (see *Virus Bulletin*, September 1991, page 18), *Novi's* scanner detected 347 of the 364 infections (95.32%). (Using an unofficial enlarged test set comprising 781 files infected with the same number of different viruses, *Novi's* detection rate was 88.86%.)

Faced with the *VB* In The Wild test set (*VB*, June 1992, p.16), *Novi* 1.1 detected 106/116 file and boot sector infections, failing only to detect 10 instances of the Whale virus. It found all of the standard infected files including Maltese Amoeba (itself a polymorphic virus) and all the boot sector viruses.

Novi 1.1 also detected all of the polymorphic virus infections and, in contrast to version 1.0.1, passed the concordance test (both tests, *VB*, June 1992, p.16). It did, however, detect a false positive in the *Windows* 3.1 terminal program (WINTER.EXE).



Certus International provide toll free telephone support in the event of a virus attack

Generic Checking (Version 1.0.1)

Novi's generic checker works well and it found minor changes (1 bit) which I introduced into files. *Certus* doesn't state which algorithm is used so I can't comment on its strength or otherwise.

Novi's generic checker is enabled through the main menu by checking 'Perfect'. The first time that you scan, with this option enabled, *Novi* creates its database, NOVIPERF.DAT, in the *Novi* directory which consists of alphabetically sorted 128 byte records. This file does not appear to be encrypted in any way and I was easily able to identify the following items:

- File Name, minus any path or drive information (12 bytes)
- File's Directory Date, Time and Length (6 bytes)
- 16-bit CRC (2 bytes)
- File Header - first 35 bytes of the file (35 bytes)

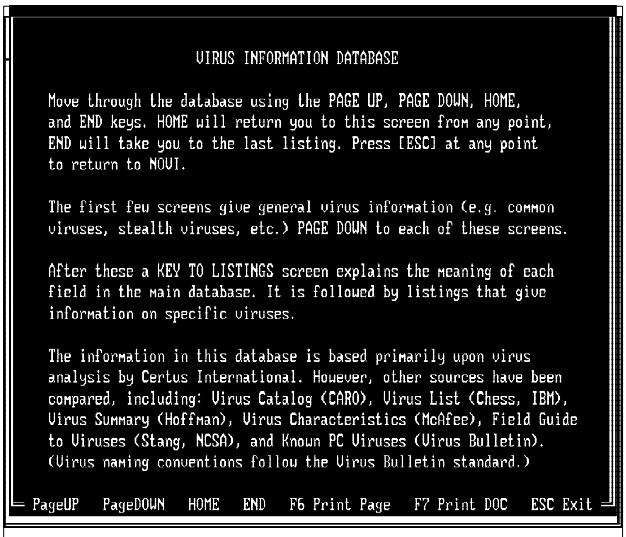
By default, *Novi* scans any file whose extension is one of COM, EXE, OVL, BIN, OVR or SYS - you can, of course, change this and include additional extensions. The test hard drive had 258 such files (totalling 11,543,732 bytes) and the times taken to scan these files are as follows:

- Scan only, 'Perfect' disabled: 7 secs (1,610 KBytes/Sec)
- Scan and add checksum details to database: 47 secs (240 KBytes/Sec)
- Scan and check checksums against database: 10 secs (1,127 KBytes/Sec)

Once the database has been created, *Novi* does not check the entire file, it checks the 35 byte header and the file's directory information - date and time stamps and its length. I was able, using *The Norton Utilities*, to modify a previously checked file without changing either the file's header or directory information. *Novi* did not detect the file had been modified. Based on this test, I have to conclude that *Novi* will not necessarily detect future viruses that insert themselves into files and modify the file internally rather than make any changes to the header. Such viruses are feasible and will doubtless appear.

The generic checking in *NOVITSR* does not impact heavily on program load and execute times, the overhead is almost imperceptable. This resident utility consumes 20k of RAM.

Based on brief testing, both *NOVIBOOT* and *NOVITSR* proved efficient in detecting virus activity. *NOVIBOOT* is designed to be run from AUTOEXEC.BAT and is responsible for checking for memory-resident viruses and checks boot sectors for known viruses. It also checks the integrity of CMOS and of the command interpreter, COMMAND.COM. Because it is loaded late in the PC's bootstrap process, *NOVIBOOT* can be subverted by boot sector viruses with a reasonable level of stealth - those that disguise their presence.



Virus database - the naming convention conforms with the *Virus Bulletin*.

NOVITSR can also be loaded via AUTOEXEC.BAT (but not a device driver) and it is designed to prevent viruses attacking critical components of your PC - such as the boot sectors - and warn you if a virus is attempting to write to a program file.

NOVITSR is highly configurable and you can include or exclude a whole raft of options. *File Watch* compares program data before and after execution in order to identify unauthorised modification. *EXE Watch* prevents modification of the header information of EXE files which is usually altered when a virus infects. A feature called *Attribute Watch* monitors attempts to alter the DOS Read Only attribute to Read/Write - a change indicative of virus behaviour.

The memory footprint size depends on which options you enable - around 17k is a typical value for a secure set up. The TSR can be loaded high, if you are using MS-DOS 5, DR-DOS 5 or 6, or a third party memory manager that supports relocating memory-resident software into the upper memory blocks (the area above the 640k and below 1 megabyte found on most '386 equipped PCs and some '286s). NOVITSR can also make use of expanded memory, if that is available.

The documentation does not state incompatibility problems with other TSRs; during testing no problems were encountered.

Adding *NOVIBOOT* and *NOVITSR* to AUTOEXEC.BAT, so that they execute upon each boot-up, added only five seconds to the boot process. But I have to question *NOVIBOOT's* checking process since it kept insisting that my boot sector had changed when it hadn't. Prior to installing *Novi*, I had installed OS/2 and the hard drive's Master Boot Sector actually contains an IBM *Boot Manager* Master Boot Sector with a non-standard Partition Table.

An OS/2 system, set up with *Boot Manager* and two operating systems, DOS 5 and OS/2 2.0, will typically contain the following Partition information:

- Non-Bootable Extended Partition (05) for OS/2 (Logical Drive D)
- Non-Bootable BigDos Partition (06) for DOS (Logical Drive C)
- Bootable Boot Manager Partition (0A) (invisible to DOS and OS/2)

If *NOVIBOOT* has detected that the bootable partition has an unknown File System (in this case, 0A) and is moaning about that, then this is an issue that *Certus* may want to address. There are a number of proprietary security systems - including *Apricot Security* - which employ non-standard Partition Table entries and such false alarms could cause a loss of confidence.

A 50 page spiral bound manual is good at explaining the various options available in the package's components, but it does contain one or two dubious statements. It claims, for example, that *Certus* has the only vaulted PC and LAN virus lab anywhere. In fact, most manufacturers could claim ownership of such a facility. The on-line documentation is amongst the least user-friendly I've seen. Help on the product and descriptions for (some) of the viruses it detects are there, but there's no index and you have to scroll through screenful after screenful of information to find the page you want.

Conclusion

Novi is a competent anti-virus product. The scanner is well maintained showing a creditable performance against a range of virus samples. The integrity checking capability, while sufficient to deal with most current virus threats, may need redesigning in the future should more viruses appear which insert themselves into a file leaving header information intact.

Technical Details

Product: Novi

Version Evaluated: 1.0.1 (version 1.1 for scanner evaluation)

Vendor: Certus, 6896 W Snowville Road, Brecksville, Ohio 44141, USA. Tel 216 546 1500, Fax 216 546 1450.

Availability: IBM PC/XT/AT/PS2 and compatible running DOS 3.1 or higher. Fully network and Windows compatible.

Serial Number: Version 1.0.1 303098, Version 1.1 350490N

Price: US$149.00, £99.99

Hardware Used: Testing was conducted on a Kamco 486 workstation. Its speed was adjusted to 10MHz which Norton's SysInfo program indicated the same processor and disk performance as a Compaq DeskPro 386/33. The hard drive contained 36 Mb in 769 files of which 11 Mb (258 files) were executable.

For details of the standard test set see *VB*, May 1992, p.23

Details about the In the Wild and Polymorphic test sets appeared in *VB*, June 1992, p.16

PRODUCT REVIEW 2

Dr Keith Jackson

SmartScan

SmartScan is one of the few anti-virus products which has never been reviewed by VB. The reason for this is simple; the vendors of this product refused to provide a review copy. *VB* thus placed an order for a copy through the usual purchasing channels, which resulted in the vendor reversing previous policy, and providing a review copy after all. Such is life.

SmartScan is an anti-virus program which claims to offer a unique feature in that it can scan for viruses 'generically'. It has three distinct component parts: a scanner (with a memory-resident option), a checksum option, and a device driver which defends against boot sector viruses. An intriguing facility called 'Dirty Mode' is provided where *SmartScan* unhooks all interrupts from any memory-resident programs so that viruses already in memory are isolated from other programs. Registered users of *SmartScan* receive monthly bulletin sheets containing information on new viruses.

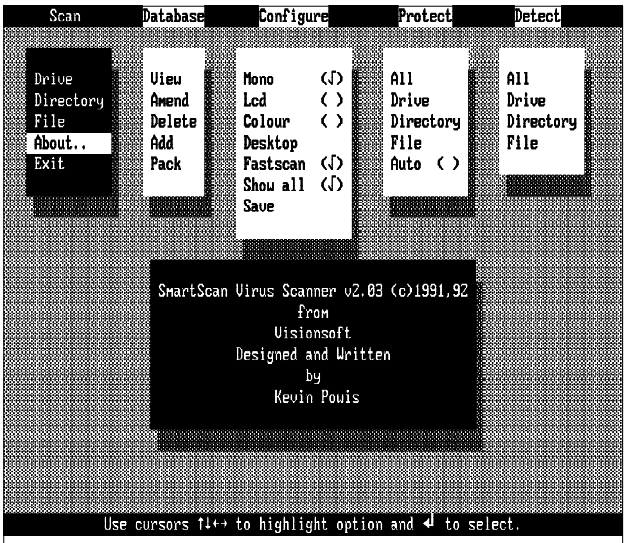
Documentation

The *SmartScan* documentation comprises a 120 page A5 book, which explains effectively how to use the software, and also contains a decent description of what a virus is and what to do if a virus infection is detected. The manual also contains a section describing several other *Visionsoft* products in detail. The *SmartScan* manual contains a section entitled 'Reviewers notes' which instructs the reviewer to ensure that the latest version is being used, provides an explanation of how to obtain the optimum scanning speed, and includes two pieces of sales blurb explaining how the cost of *SmartScan* is extremely reasonable (as if such a thing ever needed explaining). It's the first time that I've come across this idea.

SmartScan is provided on both 3.5 inch and 5.25 inch floppy disks (one disk of each format), and support is available either via the *CompuServe* electronic information system, or in more traditional fashion by fax or telephone call to the vendor.

Test File

SmartScan is distributed with a 'test' executable file containing the signature of a real virus and a text message stating that the file is not infected. Including this file is unwise. I routinely scan every floppy disk that is inserted into my PC, and invariably this is before I have read the manual (often while I am reading it!). I followed this procedure with *SmartScan* which confirmed that few things in this life increase ones blood pressure more than finding a positive detection of a virus on a newly received floppy disk. I checked the disk with several scanners, and found that only some of them (one of which was *SmartScan* itself) thought that the file in question was infected. In hindsight, this is obviously because not all scanners use the same virus pattern, but it was a while before I arrived at this reassuring conclusion.



SmartsScan offers a clear, easy to use, drop-down menu interface.

I've described this point in detail as I think that it is mistaken to distribute disks containing files which have been purposely created to look as though they are infected. The file may state that it is not infected when it is run, but if your scanner detected a 'virus' on a floppy disk, would you execute the file to find out whether it really was infected? I certainly wouldn't and I warn those who ask my advice never to do this.

The presence of this test file is mentioned in the *SmartScan* documentation about ten pages from the end. Although I've been criticised in the past for bemoaning the lack of a meaningful index in a manual (or even the lack of any index at all), if ever something should be in the index to a manual accompanying an anti-virus product then it's the presence of a test file containing a virus pattern or identity which will (not may) produce false alarms. The index in *SmartScan's* manual says nothing about this, or if it does I certainly can't find it.

Installation

Among several other questions, the *SmartScan* installation program asks whether the PC has been booted from a clean DOS disk. If the answer is no, the installation stops forthwith. Enforcing a virus-free system during installation is laudable, though an onscreen message explaining the consequences of answering in the negative would help the user somewhat.

The installation program requested the path of the subdirectory where *SmartScan* was to be installed, and as my test PC has three floppy disk drives, this path of necessity referenced drive D. Despite this, the installation program still

kept trying to read from drive C (a floppy disk drive), and unsurprisingly reported that it could not find any files. The PC is rebooted several times by the installation program, and during one of these reboots it finally decided that it really did require drive C and stopped working. I rebooted manually and found that in spite of all this, *SmartScan* had as I had requested been installed on drive D, but strangely I was left with six (yes 6) copies of the files AUTOEXEC.BAT and CONFIG.SYS on drive D, four copies of each in the root directory (with extensions .SMS, .OLD, .BAK, and the original file extension), and two copies of each in the designated *SmartScan* subdirectory (.CUR and the original extension). What all this means I have no idea, but deleting the spurious copies did not seem to do any harm. However, it should not be the user's task to clean up such a mess.

The *SmartScan* documentation explains at length that information about each execution (including installation) will be contained in a file on disk. However, the file (SMART.REP) left behind after installation contained literally nothing (its size was zero bytes). This problem may originally have been caused by the installation program insisting on referencing drive C, however the problem remained during normal execution of *SmartScan*. The file SMART.REP contained zero bytes after every scan, and the program always flicked the drive light on drive C. This needs fixing.

Scanning Performance

SmartScan offers a clear, easy to use, drop-down menu interface which provides access to the scanning, checksumming, and configuration features. I tested the scanning speed of *SmartScan* by scanning my entire hard disk (first making sure that I had followed the reviewer's advice in the manual to activate *FASTSCAN* mode, which searches executable files at suspected infection locations). *SmartScan* took 3 minutes 18 seconds to scan my hard disk. This is a creditable time, as *Dr. Solomon's Anti-Virus Toolkit* took 2 minutes 21 seconds and *SWEEP* from *Sophos* took 5 minutes 34 seconds to scan the same disk. The advice to turn *FASTSCAN* mode on seems necessary, as when *FASTSCAN* was turned off (i.e. when each byte of every executable file was checked), *SmartScan* took 53 minutes and 20 seconds to scan the same disk!

The scanner uses a virus database which contains just over 400 patterns. *SmartScan* can detect many more viruses than this total, as it can cope with 'fuzzy' signatures, and groups viruses generically, so that more than one virus can be detected with a single signature. Each database entry shows the virus name, properties (type of virus), offset (where it infects) and pattern. The database can be password protected.

The detection capability of *SmartScan* is reasonable as all of the standard test viruses (see *Technical Details, VB,* May 1992, p.23) were detected except for the Kamikaze virus which was not in its database. The scanner has a built-in command syntax and scans can be implemented to suit any requirements using a control file and 'macro-like' structures.

SmartScan was also run against viruses known to be in the wild and its polymorphic virus detection ability was tested (this test is exactly the same as that published in *VB*, June 1992, pp. 13-16). The product detected 91% of infected files and boot sectors from the In The Wild test set failing only to find one instance of Tequila and nine generations of Whale (106/116 infections were detected). *SmartScan* detected 66% of the polymorphic infections failing only to detect any of the fifty generations of V2P6 (100/150 infections were detected).

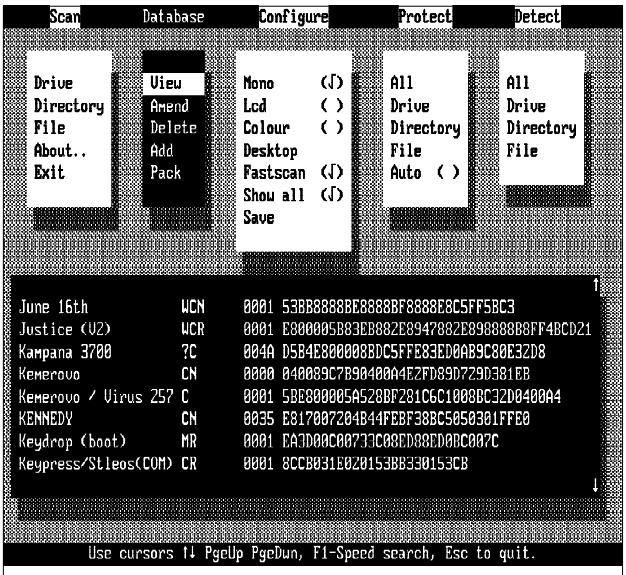
I encountered two problems with the scanner. First, when activated twice consecutively, its initial checks reported that memory 'may be infected by Amoeba/1392' during the second execution. I tested this quite extensively, even down to rebooting from a clean floppy disk, executing *SmartScan* immediately and requesting two consecutive scans. The error is consistently reported. Is it a correct report? I don't know. Curiously this virus is the one in the 'test' program described above, even though at no time did I execute this 'test' program. The second problem was that with its AUTO mode off to prevent automatic generation of checksums, the scanner still generated two hidden files in the root directory (called Magfile and Magdir respectively). I have no idea what these files do; they are not mentioned in the documentation.

Checksumming

The checksum part of *SmartScan* does not check all parts of a file. The manual quotes as an example that text can be changed within a file without triggering an error. The actual method used to checksum a file is not explained in detail. How is it possible for *SmartScan* to distinguish between code and text unambiguously? For instance, on many processors, it is possible to write a functioning program that looks entirely like a text file (by choosing the right instructions and offsets).

```
------------------------------------
--- Test Smartscan distribution disk ---
------------------------------------
SWEEP virus pattern finder
Version 2.38
(c) 1989,92 Sophos Ltd, Oxford

System time 22:53:44, System date 20 June 1992

This issue includes virus patterns and identities known to Sophos
up to 01 June 1992

Complete Sweeping 3 areas for 728 patterns and 24 identities.
Press F2 to quit.

>>> Pattern 'Amoeba' found in file a:\SMARTSCN\DUMMYV.COM starting at 000043
Elapsed time 00:27
152.2 Kbytes swept in 0 minutes and 27 seconds at 5773 bytes/second
1 virus pattern and 0 identities were discovered.
1 file out of 6 was infected.

Telephone Sophos on 0235 559933 (+44 235 559933 international) for advice.
Press any key to continue ...
```

File DUMMYV.COM triggers an alert from *SWEEP*



The pattern library can be password protected to prevent tampering

Further to this, the actual algorithm used to calculate checksums is not explained at all, the manual just says: 'The checksum algorithm that *SmartScan* uses is designed to run quickly and detect virus-like changes to a file.' This simply is not good enough. What has been compromised in the quest for speed? Exactly what is a 'virus-like change'? The information provided in the manual is devoid of such technical content.

For the record, *SmartScan* calculated all the checksums for the entire hard disk on my test PC (containing 21.7 Mbytes) in 3 minutes 30 seconds. It could then test these checksums for alteration in 1 minute 12 seconds when *FASTSCAN* was ON, and 3 minutes 20 seconds when *FASTSCAN* was OFF. These timings seem impressive, but are meaningless without any technical details of what is actually being calculated.

The device driver provided with *SmartScan* to prevent boot sector infections seemed to coexist with my hard disk, though I admit that I did not summon up the courage to damage my hard disk's boot sector in order to test its restorative properties. Such actions seem rather like setting your house on fire to see whether the fire brigade is operating efficiently!

TSR Scanning

The memory-resident option scans any executable program and will not allow manipulation (execution or copying) of an infected program. Note that testing this feature was the stated reason why the 'test' file was provided (see discussion above). Not surprisingly this program has the same scanning efficiency as the main scanner. However, performing such a scan must impose an overhead: *SmartScan* is no exception. Copying 85 executable files (2.47 Mbytes) from one subdirectory to another took 1 minute 49 seconds without the memory-resident option invoked, rising to 6 minutes 29 seconds with the option installed. I was careful to ensure that exactly the same disk locations were used during this copying test and conclude that a 256% increase will not be acceptable to the majority of users. Note that the percentage increase in the time taken to just load a program will be even greater than the figures quoted above, as no checking takes place during the write phase of the copy operation.

Conclusion

I believe that scanning for viruses using 'fuzzy' signatures and grouping viruses 'generically' is a particularly useful feature of *SmartScan*. These methods are similar to the fuzzy matching used in IBM's *VIRSCAN* (see *VB*, May 1991 pp. 16-17) Other scanner packages have features where wildcard characters in the virus signature are allowed and I believe that the differences between such techniques and 'fuzzy matching' to be semantic. This type of feature will help to detect (some) hitherto unknown virus variants. [*Fuzzy matching should not be confused with heuristic scanning which identifies virus code by searching for suspicious instructions. Tech Ed.*]. Once again I am finding fault with the marketing spiel attached to the product rather than the product's actions per se. How often have I said that in reviews of anti-virus products?

With the exception of the installation procedure (which really does require some more development work), and the problems generated by the inclusion of a virus 'test' file, I found nothing fundamentally wrong with *SmartScan*. Indeed I wonder why the vendor so vehemently refused in the past to provide a copy for review. Have I missed something?

Technical Details

Product: *SmartScan*

Vendor: *Visionsoft Ltd*., Five Lane Ends, Bradford, England BD10 8BW, Tel. (+44) 274 610503, Fax (+44) 274 616010.

Availability: Hardware (and/or software) requirements for SmartScan are not explicitly stated. SmartScan will operate on a network, and a PIF file and an icon are provided for Windows 3. The memory-resident part requires a PC or 100% compatible running MS-DOS v3.00 or higher, and is network and Windows 3 aware.

Version Evaluated: 2.02

Serial Number: (Batch number) 06-92-624

Price: £49 for a single copy, £295 for an unlimited site-licence, £35 for 12 monthly updates.

Hardware Used: An ITT XTRA (a PC clone) with a 40 Mbyte hard disk, one 3.5 inch floppy disk drive, two 5.25 inch floppy disk drives, 640K of RAM, operating under MS-DOS v3.30.

For details of the standard virus test set see *VB*, May 1992, p. 23

Details about the In the Wild and Polymorphic test sets appeared in *VB*, June 1992, p.16

SEMINAR REPORT

Comer on Fraud

'Every fraudsman I've ever encountered has worn either fancy shoes or fancy socks - they're invariably foot fetishists.' So began a two-day *IBC Technical Services* seminar on fraud detection conducted by veteran investigator Mike Comer.

Comer is the chairman of *Network Security Management* (known simply as '*Network*'), a division of *Hambros Bank* dedicated to uncovering all forms of deviousness and corporate hanky-panky. Comer describes himself as the most cynical person he knows and he certainly has the demeanour of a man who has seen it all. According to Comer's Law: 'One in four people is honest, one in four is dishonest; the other two are open to suggestion.'

'The first thing I do is go through the incoming invoices file. Any submission which is photocopied or not printed, or which bears a PO box number and no telephone number, is taken out for further scrutiny. If the invoice isn't folded it indicates that it was never posted and most probably arrived in a briefcase. If the description of goods or services supplied is so vague as to be meaningless I know I'm onto something.'

Comer's techniques, best described as applied commonsense, have evolved during a twenty year career conducting investigations with organisations as diverse as *HM Customs and Excise* and *Shell UK*. 'When looking at personal expenses, I'm not necessarily interested in the amount spent, but I do care about the time that any restaurant bill is made out; 'lunches' which consistently take place in the evening are an obvious line of enquiry. Puddings and liqueurs often indicate the presence of a woman. None of this is evidence but it's this sort of indicator which leads to more solid ground.'

Network has uncovered some of the UK's most celebrated 'scams'. The meat processing company *Walls-Matheson* called in Comer's team when it suspected fraudulent activity among its delivery drivers. 'We put in a man undercover. During his interview he was asked whether he'd be prepared "to bend the rules a bit"; the whole division including its supervisors and foreman was bent!' Within days, numerous frauds had been detected. It transpired that, among other deceits, delivery men were removing hopelesly expired sell-by dates from pork pies and sausages using nail-polish remover and selling the goods to retailers as fresh stock.

'Most frauds occur because the perpetrator discovers a loophole by accident. We had one case at a supermarket where the cashiers, having closed down for the evening, would spend hours bashing entries into the cash-tills. They'd discovered that the tills could be 'clocked' like a mileometer in a car; each evening they'd clock the tills and enter a lesser figure than was actually received and then pocket the difference.'

One of Comer's abiding themes is that fraud is endemic, that there is little to distinguish the falsified expense account from grand larceny. 'When I'm stuck in a traffic jam because a group of workmen are filling holes in the road I get frustrated; I know they're probably there because someone is receiving a back-hander to extend the contract-hire of the excavator or generator equipment.' Another Comer theme is that fraud takes place at all levels and in all departments. 'The purchasing department is my first port of call. Collusion between suppliers and purchasers is rife. When purchasing anything, check for possible collusion not at the evaluation stage but at the specification stage; the number of times that a specification is tailor-made for a favoured product is staggering.'

Nigeria is currently *the* breeding-ground for fraudsters. 'You'll know it when you see it; a letter from Nigeria saying that X million pounds needs banking and that your company will receive 10% of that total figure for assisting in the transaction. You're then instructed to send four copies of your company's letterhead paper, full company bank account details and a signed declaration that you have read and understood the letter. The fraudsters, in one fell-swoop, thus gain everything they need to embezzle you!' When one such fraudster was asked how such a patently transparent scheme could ever succeed he replied that, rather like direct-mail, he only needed a 1% response rate!

Attendance at one of Comer's classes is obligatory, not just for auditors and security personnel, but for anyone interested in the human condition; refreshingly, his cases studies are firmly grounded in the real world, whether it be pork pie deliveries in Essex, wire fraud in The City or shenanigans on the service station forecourt. 'We dressed up in our overalls and I sat at the cash-till; the first driver had filled the tank with 20 gallons of DERV. I dutifully filled out the receipt for 20 gallons but he queried it and said that the regular cashier always added 50% to the receipt. Within hours we had a backlog of lorries whose drivers were on the take.'

On interrogation, Comer says that it is necessary to induce and heighten apprehension. 'Given the choice between making a suspect conceal his activities or making him lie, I always choose the latter. The truth comes from memory and is consistent. Lies come from the imagination and can rarely be sustained. Very often it's best to force documents or other evidence into the suspect's hands; guilty people will do anything to distance themselves physically from incriminating evidence.' Having induced anxiety, it is necessary to reduce the tension suddenly. 'At this point' says Comer 'most suspects simply gush with confession.' Seeing these tactics in practice, one got the feeling that if the Devil ever interrogates Comer, Lucifer will be odds-on favourite to confess first. Doubtless the Devil also has an appalling taste in hoof-wear.

Network Security Management Ltd, Network House, Bradfield Close, Woking, Surrey, UK. Tel 0483 750022.

END-NOTES & NEWS

2nd International Virus Bulletin Conference, 2nd-3rd September 1992, Edinburgh, Scotland. Contact Petra Duffield. Tel 0235 531889.

***PCs Under Attack* is an educational video** which provides tips on damage assessment, removal of viruses and full restoration after clean up. The video (NTSC format) is available for purchase for $495, previews at $40. Further information from *Mediamix Productions Inc*., New Jersey, USA. Tel 908 277 0058.

***The Computer Virus And How To Control It* is a 23-minute video** (VHS format) which shows the commonsense steps that the PC user can take to identify, avoid and eliminate viruses. The video is available for $395. *James C Shaeffer & Associates*, Ann Arbor, USA. Tel (toll free) 800 968 9527.

***SMARTFACTS* is a menu driven database** containing information about computer viruses, their characteristics, infection methods, trigger conditions etc. The software costs £98.00 which includes free quarterly updates for the first year. *Visionsoft Ltd*, UK. Tel 0274 610503.

Command Software Systems of Florida provide a range of PC and LAN security systems including *LANGARD*, *Security Guardian* and have been appointed **distributor for *F-PROT Professional***. Command Software, USA. Tel (toll free) 800 423 9147.

DETECTPlus and ***SCANPlus*** are anti-virus and integrity checking programs from *Commcrypt Inc*., Maryland 20705, USA. Tel 301 470 2500.

Victor Charlie is a new anti-virus software package from *Bangkok Security Associates*, PO Box 5-121 Bangkok 10500, Thailand. Tel 66-2 251 2574. The US distributor is *Computer Security Associates*, USA. Tel 803 796 6591.

S&S has been appointed UK **distributor for the *NCSA's* comparative virus scanner report**. 20 different anti-virus products are sampled, including the *S&S AntiVirus Toolkit* which gains the highest overall score in the report. Information from *S&S*. Tel 0442 877877.

V-CARE is 'The most powerful anti-virus EXPERT SYSTEM your money can buy!' The US distributor is *Sela Consultants Corporation*. Tel 800 822 7301.

VFIND is a scanner program supporting Sun Microsystems, Sun3, SPARK, NeXT, Motorola 88000 RISC, Unix and Unix-like environments. VFIND claims to detect viruses infecting MS-DOS, Macintosh and Amiga executables. *Cybersoft*, Pennsylvania, USA. Tel 215 825 4748.

InocuLAN is a NetWare Loadable Module (NLM) from *Cheyenne Software Inc*. Features include automatic scanning, backup, restore and MHS, broadcast or pager notification of infection or disk corruption. *Cheyenne Software Inc*., NY, USA. Tel 516 484 5110.

VyGARD is an 'indestructable hardware device, armed against virus invasion, which takes control the moment the computer is turned on.' The UK distributor is *Microlife*, 11 Mythop Rd, Lytham-St-Annes FY8 4JD. Tel 0253 735979.

Sophos UK is holding hands-on **Virus Workshops** in Oxford in July (22nd-23rd), September (8th-9th) and November (17th-18th). Tel 0235 559933.

IBM UK is holding a **Virus Management Course** (July 14th) and **Virus Hands-on Course** (July 15th) in Nottingham. *IBM* is also holding a **Virus Master Class** in Edinburgh on September 7th. Tel 081 864 5373.

VIRUS BULLETIN

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA
Tel 203 431 8720, Fax 203 431 8165