July 1991 ISSN 0956-9979

VIRUS BULLETIN

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

CONTENTS

EDITORIAL

Two Years On...

Virus Bulletin enters its third volume at a time of escalating crisis. In the course of talking to anti-virus software developers, police officers and computer security managers, both in Europe and the United States, it is becoming clear that the incidence of real world computer virus infections is increasing at an alarming rate. Companies active in this field (including *VB* itself) are reporting a noticeable increase in the number of genuine virus attacks.

The number of virus samples (which includes a veritable menagerie of assorted 'flora and fauna') always was something of an academic curiosity, of most interest to the 'schoolboy collector' mentality - it's hard to believe that two years ago computer virus samples really were coveted rather like marbles or prize conkers. However, an exponential curve is now clearly drawn on the graph - new computer virus samples are appearing daily and the emergence of virus exchange Bulletin Board Systems and the widespread dissemination of source code will only serve to fan the flames.

All of this, of course, is of no interest unless these samples manifest themselves in the real world. Unfortunately, this is exactly what is happening, with more and more different specimens being encountered. Summing up this alarming state of affairs, one cynical software developers (who shall remain anonymous) was heard to say: "With all these *real* viruses hitting *real* companies, who knows - we may eventually have to provide our customers with a *real* product and *real* technical support!"

Under the strain of exponential virus development and demands from corporate end-users who are now aware of the extent of the problem, some anti-virus software developers have adopted the strategy of the ostrich - by burying their heads in the sand and refusing to update their virus-specific products, they believe that the problem will simply go away. Some of the newcomers to this field (those companies which have 'jumped on the bandwagon'), upon realising the enormity of the task, must dearly regret the day that their marketing advisers persuaded them to undertake it.

The major problems are associated with virus-specific solutions. Ultimately, scanning software is intellectually bankrupt as a means to generic detection. This has been a persistent theme within the pages of this journal - back in 1989, Keith Jackson wrote: 'Virus-specific methods are in the end doomed to failure... they come into their own when it is thought that a virus infection exists and needs to be eradicated.' Surprisingly, his statement was then the subject of heated debate. The fact that Dr. Jackson has now been vindicated is not so much testament to his powers of augury as to his good common sense.

Some software developers will privately concede the futility of this approach - one major developer, recognising the inevitable demise of virus-specific detection, recently declared his intention 'to be the last off the cliff'. Estimates from various software developers as to how long their virus scanning programs can remain effective vary from between eighteen months to five years. If these software developers had listened to Dr. Fred Cohen (who predicted this and other fundamental truths years ago) they would have saved themselves much heartache.

In terms of numbers and propagation, the virus problem is getting out of control - a fact with which the 'virus industry' is having to come to terms. However, in terms of good housekeeping and prevention the problem remains eminently manageable. But as Dr. Peter Tippett said earlier this year: 'It is eminently manageable - it's also eminently badly managed.'

The Lunatics Are Taking Over The Asylum

Two unusual anti-virus packages arrived on the doormats of *VB* correspondents last month. Neither is particularly efficacious, but they have one thing in common - they are both developed by virus writers.

Mark Washburn, self-confessed author of the V2P1, V2P2 and V2P6 viruses, has released his *SECURE* package. More alarmingly, *Abacus* - the Michigan-based publishing company - has released *VIRUS SECURE FOR WINDOWS* by none other than Ralf Burger.

As seasoned *VB* readers will know, Burger is directly responsible for two virus families - the Vienna group and the Burger/Virdem group. He provided the full source code of a number of computer viruses in *Computer Viruses A High-Tech Disease*, a virus writer's recipe book published initially by German publishing house *Data Becker* in 1988 and in a translation subsequently printed in the United States by *Abacus*. It is rumoured that German authorities have foiled Burger's attempt to publish a sequel in that country; whether or not similar pressure is put on *Abacus* remains to be seen.

It is inevitable that would-be virus writers will buy such books or otherwise obtain such published viral source code. Both Washburn and Burger are simultaneously fuelling the computer virus phenomenon and profiting from it. No sensible person can condone, or in any way excuse, these individuals' duplicitous behaviour. Recent calls to 'accept them back into the fold' should be summarily dismissed. As it is not the editorial policy of this journal to promote or lend credibility to those who indulge in such schizophrenic activities, neither package will be reviewed. Third parties afflicted by computer viruses connected with the publication of virus source code should consider seeking legal advice about the feasibility of sueing the publisher concerned. *Abacus* have been informed in writing of the dangers associated with published source code and the proven links between Burger's first book and subsequent computer virus outbreaks - they cannot plead ignorance.

TECHNICAL NOTES

Upset Offset

The table of known IBM PC viruses, published in this edition, contains a search pattern for almost all of the viruses listed. Each pattern is followed by a three-digit hexadecimal offset.

This number is the position of the search pattern within the virus body. Unfortunately, there is no simple relationship between the offset and the position of the pattern within an infected file.

In the case of viruses located at the beginning of .COM files the pattern offset is equal to the file offset. As an example, take the pattern for the Amstrad (Pixel family) virus:

```
C706 0E01 0000 2E8C 0610 012E FF2E 0E01 ; Offset
114
```

In every file infected with this particular variant, this pattern can be expected to be found at offset 114H.

The relationship is more complicated in the case of viruses which append their code to the files they infect. The start of the virus code cannot be determined from the entry point of the virus, or the target address of the initial JMP instruction if there is one. Instead, it requires knowledge of the length of the virus code, as the pattern is in most cases located a fixed distance from the end of the file.

A sample file infected with the 10-past-3 virus (748 bytes) illustrates this requirement. The search pattern is:

```
B840 008E D8A1 1300 B106 D3E0 2D00 088E ; Offset
069
```

The length of the infected sample file is 848 bytes, so this 748 byte virus starts at offset 100 (64 hex) in the infected file. Adding the search pattern offset, we arrive at the number CDH - the offset in this particular file, where the pattern should be found.

This method does not work in all cases, in particular not for any viruses which add a variable number of 'garbage' bytes after the virus code.

An equally serious problem with the use of published offsets arises if the same search pattern is used to detect multiple variants of the same virus. Usually, under such circumstances, the offsets will vary.

This was not a problem two years ago - the first edition of the *Virus Bulletin* listed only 14 virus patterns and tabulating the offsets made perfect sense. A virus scanning program could gain a considerable increase in speed by only looking for each pattern at one specific location within each infected program.

This does not apply today - the ever-increasing number of new variants of older viruses has made this method redundant.

So, the answer to the question: 'What are the offsets good for?' appears to be 'Not much.' In fact publication of this information appears to be generating confusion rather than reducing it. The hexadecimal patterns remain valid for detection when used in conjunction with a 'secure' byte-by-byte file search, but highly specific detection using the published offsets is no longer recommended.

The offset is yet another victim of exponential virus development: the sheer number of virus samples and a multitude of infection strategies have rendered further publication of this information useless for the majority of *VB*'s subscribers. A decision as to whether to continue publishing offsets will be taken in August.

Selectivity

Selectivity is the process whereby developers of virus scanning programs omit search routines for computer virus samples on the basis that the virus poses no threat, is extinct or is a research or 'lab' virus never likely to be encountered in the wild. The recent decision by *McAfee Associates* to exclude detection data for 70 crude overwriting viruses is an example of this process. Another manufacturer, probably motivated more by panic than pragmatism, is reported to have declared ninety percent of all viruses to be 'extinct'

Anti-virus software developers now realise that scanner development and maintenance is costly in terms of resources. Those manufacturers currently engaged in maintaining virus-specific defences (both virus scanners and disinfectors) are becoming aware that this approach is extremely time-consuming and costly in terms of programming time. Some developers have thus decided that the simplest solution to exponential virus growth is simply to ignore it.

As the numbers of virus samples increase, scan run times (in Kilobytes per second) will deteriorate. Well implemented scanning programs have highly optimised search engines which will serve to minimise the impact of exponential virus development. However, the simplest detection engines are already beginning to degrade as more and more detection data is required to keep them up-to-date. By declaring certain functioning viruses 'non-threats', 'extinct' or 'crippled' and by subsequently removing them from scanning data, these developers will recover hard-pressed scanning resources.

The other motivation behind 'selectivity', is the realisation by some software developers that continually updating and distributing a product in a secure way on a monthly or bimonthly basis to all registered users is both time-consuming and expensive. Some of the newcomers to anti-virus software, particularly those companies which are fundamentally market-driven, now realise that anti-virus software is not the 'pot of gold' it once promised to be.

If a virus functions it is a potential threat. Risk assessment defines overwriting viruses as less of a threat than most other specimens and informed observation suggests that a virus as cumbersome as Whale will not spread far before discovery.

However, the existence of virus exchange bulletin boards, the capacity to transfer viruses by modem, the continuing appearance of modified and optimised viruses, the availability of source code and the reappearance of viruses thought to be 'extinct' suggest that product developers dare not be too selective about those threats from which their customers should be protected.

Companion Viruses

Recent communications to *VB* have shown that some confusion persists about the so-called companion viruses and the way in which they infect. Companion viruses exploit the MS-DOS property that if two programs with the same name exist, the operating system will execute a COM file in preference to an EXE file. A companion virus will create a COM file for every EXE file it 'infects' in the same directory, for example WS.COM for WS.EXE. The COM file will be marked hidden and contain the actual virus code (the EXE file is left unchanged).

If the instruction is entered at the command line for the legitimate program to execute (which will usually be entered without the addition of a file extension (e.g. C:>WS to execute WS.EXE), the infected and bogus COM file will execute first and then transfer control to the legitimate EXE file which will function normally. All currently known companion viruses take the elementary precaution of setting the COM file attributes to hidden. In practice, companion viruses do not spread well from computer to computer, since the DOS COPY command does not copy hidden files.

That Bit's Wrong

Virus researchers are creating new virus variants. This may be a surprising statement, but it is nevertheless true. As entire collections of viruses are shipped back and forth between continents, some of the viruses become garbled. This may happen when files are read or written, or if a memory byte becomes corrupted in a way which does not affect parity.

An example was discovered recently in a virus which had been sent from Europe to the US and was sent back as part of a collection of 1700 infected files supplied by the *National Computer Security Association* in the States. The file which was received did not match the original one, but the difference was very small, only a couple of bits.

This modified virus functions normally, and the change is sufficient to classify it as a separate variant, even though it only exists in the collection of virus researchers.

Ultimate Minimalism

The challenge to write ever more highly optimised viruses continues unabated, a trend confirmed by the recent appearance of a 30 byte virus. Unfortunately, it was posted in source code form on *Fidonet*, so it has now spread worldwide. It is safe to predict an even smaller virus in the future, although the minimum size appears to be 24 or 25 bytes. As these viruses are of the primitive overwriting sort, they are a less serious threat than most other specimens.

PKLITE and *LZEXE*

It is becoming standard practice among virus authors to distribute viruses in compressed form, which automatically unpack on execution. The files are generally created by *PKLITE* or *LZEXE*, although other less well known programs such as *DIET* and *EXEPACK* have also been used.

The benefits to the virus distributor are obvious - most virus scanners are not able to unpack these dynamic decompression programs for scanning. Thus, only files infected with *second* or *later generations* of the virus can be detected - the original 'distribution' program will contain the virus on disk in *compressed form* and this will consequently remain undetected. Every time this compressed file is executed, the virus will be decompressed and placed in memory thus reinfecting the PC. Some scanners are capable of diagnosing whether or not a file has been compressed in this way, thus giving the user the opportunity to identify and delete it.

False-Positives

A report appeared in *Computers & Security* (April 1991) about false-positives caused by a file called VIRPATS.BIN which is part of the *SWEEP* virus detection package by *Sophos Ltd.*

Sophos says that VIRPATS.BIN was included as a 'target' file with which the user could conveniently verify correct installation of the *SWEEP* package. According to *Computers & Security*, other virus scanning programs went 'berserk' when scanning the VIRPATS.BIN file due to the fact that it contained commonly used search patterns. Due to this and subsequent complaints, the file VIRPATS.BIN has now been removed from the *SWEEP* package.

False-positives (whereby a virus is reported but is not actually present) are an ongoing hazard for the developers of virus scanning programs. *VB* is itself guilty of publishing a pattern for one virus (Kamikaze) which has caused numerous false alarms. The pattern published in *VB*, February 1991, p. 8 should **not** be used. (See also Mark Drew's letter on page 10).

A recent example of a particularly embarrassing false-positive (for both companies involved) once again involved the *SWEEP* package from *Sophos* and the *VIRFIND* program from *S&S Ltd.* The *VIRFIND* program itself contains a pattern used by *SWEEP* version 2.26a for detecting the Syslock virus!

TUTORIAL

Fixed Disk Boot Sectors and Post-Attack Recovery

When considering boot sector viruses there are two main types - those which infect the Master Boot Sector and those which infect the DOS Boot Sector. Since these types of virus are invariably the most awkward for non-technical users to eradicate, it may be useful to describe exactly what the differences are between the Master and DOS Boot Sectors and just how their corruption may affect various machines.

In examining the boot process, it is important to differentiate between the *physical* presence of various disk drives and the *logical* appearance that they may present to the machine's controlling software.

As an example we will examine a computer containing two 64 Mbyte fixed disk drive units, referred to as *physical drives*. Once the machine has been booted, these two units may actually appear to be two or more *logical drives* referenced as C:, D:, E:, F: and so on, of various sizes depending upon the current configuration. This dual identity is achieved by a combination of data stored on the physical drives themselves and the Disk Operating System software (DOS) which is loaded from them.

Communication with each physical drive is handled by the Basic Input/Output System software (BIOS) which is usually stored in the Read Only Memory (ROM) of the machine and is therefore unalterable.

It is important to recognise that the BIOS requires information to address 'physical' drives, while DOS needs 'logical' drive information. The translation process between logical and physical takes place at the lower levels of DOS and it is this translation which needs fixed information about the actual location of each of the logical drive areas on the disk surfaces.

The Boot Process

To make this a little clearer, let's first examine the actual boot process and then consider how this affects the higher level functions of disk file access.

When a machine is first switched on, all of the Random Access Memory (RAM) areas are in an indeterminate state and contain no intelligible program code. The hardware design of the machine is such that power is applied to the central processing chips together with a 'reset' signal such that processing always begins at a predetermined point.

This point is within the fixed ROM area and thus contains intelligible program code. Thus, after a few milliseconds in which the power supply settles within accepted limits and fixed disk drive motors come up to their normal operating speed, the machine begins processing the instructions contained in the ROM.

The first section of these instructions completes various tests, of which the most apparent to the user are the testing and clearing of memory and the polling of potential hardware additions. Later sections test and initialise the various hardware peripherals (keyboard, VDU etc.) and set up highly specific memory areas which make the various BIOS routines available to higher level programs.

POST

Included in this section might be reference to non-volatile (i.e. battery driven) RAM and clock areas where variable configuration information (like the time, date, password, etc.) may be stored. All of this testing and initialisation is referred to collectively as the *Power On Self Test* (POST) routine.

Once the POST routine has completed without error, the machine is ready to begin loading information from the disk. In order to preserve maximum flexibility, the final part of the POST tests (on all IBM compatible PCs) issues a read request to the BIOS to collect the very first sector on the first floppy drive. If a disk is not available in that drive, the POST routine goes on to issue a similar BIOS request to read the first sector of first hard drive. If that also fails, the machine may display an error message, or just pass processing to an inbuilt language system.

On the earliest IBM PCs (many of which had no fixed drives) the POST routine invoked CBASIC (Cassette BASIC) if no boot disk was available. In most normal circumstances however, there is a fixed disk and data in its first sector is read into memory.

Since the conventional addressing method counts both tracks (or cylinders) and heads from 0, and sectors from 1, the physical address of the Master Boot Sector is always Track 0, Head 0, Sector 1.

Track 0, Head 0, Sector 1

The contents of this first sector usually consist of a combination of data and executable code. On most machines, the data includes a Partition Table as well as one or two plain-text error messages. Once the POST routines have loaded this sector into memory, processing is passed to the executable code that it contains, and from that point the machine is under the control of the disk-based software.

The sole purpose of this Master Boot Sector is to provide a uniform starting point for disk-based software, to which end it will attempt to locate and load a viable operating system. To understand the location process, we must first examine the structure of the Partition Table contained within the Master Boot Sector.

The Partition Table

The Partition Table consists of 4 records, each containing 16 bytes (64 bytes in total). Each record contains information about the location, type and size of one partition on the disk. A partition is a predefined area of the physical disk drive and the operating system must know exactly where these areas are located. A physical drive therefore contains at least one and perhaps as many as four partitions. Unused records in the table are marked with zeros. Each used record in the Partition Table contains such information as the start and end addresses (Track, Head and Sector) of the partition area and the partition size (expressed as a number of sectors). Other information relating to the type and status of each partition is also stored.

To locate the operating system, the executable code within the Master Boot Sector accesses the Partition Table records, looking at a status byte which contains either 80H or 0. Only one record in the table may have a status of 80H and this is used to indicate the 'active' partition with which the bootstrap process will continue. Once this record is identified, other information is collected and used to generate a sector address for the next read request to the BIOS.

Partitioning

Just as the first sector on the disk is called the Master Boot Sector, the sector at the start of each partition (as noted by the address in the relevant Partition Table entry) is referred to as the DOS Boot Sector. Thus a single fixed drive unit could have up to four partitions containing five boot sectors (one Master and four DOS), although only one DOS Boot Sector can be 'active' (i.e. selected for bootstrapping) at any time.

It should also be understood that although the Master Boot Sector **must** be the first sector of the disk, the DOS Boot Sectors can be located virtually anywhere.

To sum up so far, at boot time - after the Power On Self Test routines have been executed, the machine loads the Master Boot Sector from the disk. This in turn locates and loads the first sector of the active partition and passes control to it. The DOS Boot Sector (which is similar to the Master Boot Sector in that it contains executable code) also contains more detailed information about the actual structure of data stored within its own area on the disk. This information is referred to as the BIOS Parameter Block (BPB) and it contains details such as the size of the File

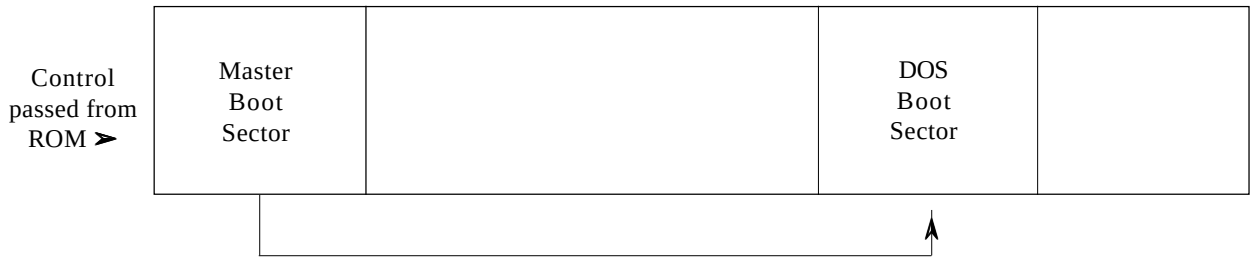


Figure 1. Hard disk before virus infection

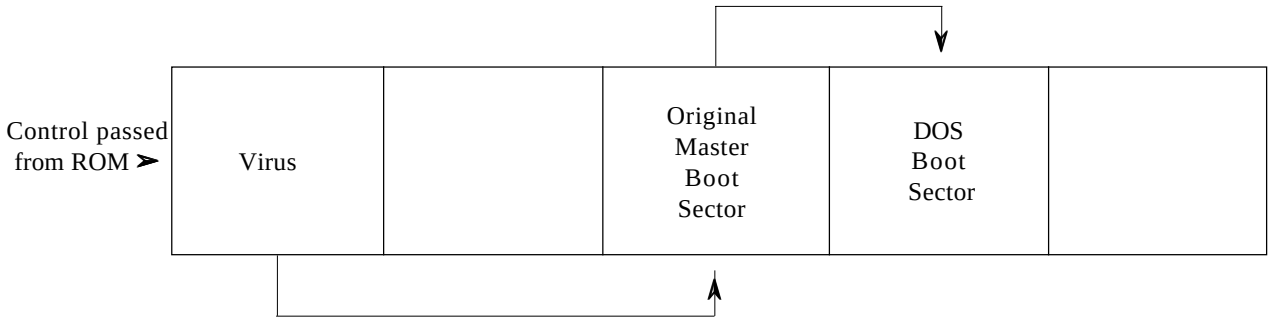


Figure 2. Hard disk after infection by the New Zealand virus

Allocation Table, how many sectors are occupied by the Root Directory and so on.

The contents of the BPB are used by the executable code within the DOS Boot Sector to calculate a disk address at which specific system files may be found. Once located, these are loaded to build DOS in memory.

Under most operating systems, the Partition Table information is used to build an internal 'map' of the layout of the fixed disk. The various areas are then categorised into 'primary' and 'extended' partitions and given the next available letters as 'names' for high level access. Referring back to our example machine configuration containing two physical drives, if each of these has been partitioned into two equal sections, then after the operating system has loaded, the machine has four logical drives (C:, D:, E: and F:) each of 32 Mbytes in size.

The actual location of these areas within the physical disk drive(s) is needed only by the lower levels of DOS. Setting up the various partitions is achieved by a program called *FDISK* (or an equivalent in other systems) which allows the user to allocate space on a physical drive by modifying the addresses contained within the Partition Table.

To simplify calculations, each partition is usually marked as starting at Sector 1 of whatever Track and Head are chosen. This means that as far as Track 0, Head 0 is concerned, the sectors from 2 to the end of the track are 'in limbo' - like the Master Boot Sector, they do not belong to any logical drive.

High-Level Formatting

Once DOS has been loaded, executable programs can communicate with the disk drive by issuing an appropriate request to one of the many DOS functions. As far as DOS is concerned, each partition now becomes a 'logical' drive and it becomes possible, for example, to FORMAT drive D: even though at BIOS level, drive D: is only a particular section of the larger physical drive.

In fact DOS maintains a separate addressing system within each partition by numbering the sectors sequentially from 0 and there is no provision at the DOS level for access to the drive on a track by track basis. This makes it impossible to access the Master Boot Sector using DOS and it is thus apparent that a virus stored in the Master Boot Sector will survive a normal DOS FORMAT command and becomes active again whenever the machine is rebooted.

Low-Level Formatting

The low-level format requirement suggested by many researchers for removing such viruses may communicate with highly specific sections of the BIOS or even directly with the hardware but the result is the same - every Head or every Track on the physical unit is cleared of data and formatted according to the specifications of the formatting program.

Certainly a low-level format will clear the problem but users may not be aware of exactly which logical drives may be affected by it. For example, on our hypothetical machine with two physical drives (i.e. two hard disks) partitioned into four logical ones, a low-level format of the first physical drive would erase all data from logical drives C: and D:, leaving the data on logical drives E: and F: intact. However, on booting the computer from a system diskette, the computer would register drive C: (formerly drive E:) and D: (formerly drive F:) because the operating system assigns drive letters sequentially starting with drive C: as the first logical drive.

The important point to remember is that the Master Boot Sector is *outside* the normal DOS structure, while the DOS Boot Sector is not. Clearing the Master Boot Sector makes *all* partitions inaccessible, while modifications to the DOS Boot Sector only affect that particular partition.

Viral Activity

Since both Master and DOS Boot Sectors contain executable code, they can be infected by a virus.

When a boot sector virus is loaded into memory, there is no Disk Operating System. Thus any writing to the disk (as part of the virus' replication mechanism) must be done at low-level, i.e. on a Track, Head and Sector basis.

Some of the earlier boot sector viruses had code which accessed the File Allocation Table (FAT) area of a partition in order to discover spare sectors to store their excess code. Fortunately, this tactic has recently become much more difficult as new versions of DOS (notably 3.31, 4.xx and 5.xx) use different FAT structures.

Boot sector viruses are almost invariably memory-resident and usually intercept the BIOS interrupt 13H in order to spread their code to other disks during normal disk operations.

The Master Boot Sector is usually the only sector on Track 0, Head 0, which contains any critical data. However, some disk drives use the spare sectors for special configuration information which is not held in the normal DOS structure of the drive. For example, IBM PS/2 machines contain global drive information on Track 0, Head 0, Sector 2 - and if this is overwritten by virus code then the computer will not boot correctly.

The widespread New Zealand virus was the first specimen to store its code in 'spare' sectors. This storage of virus code in spare sectors, which has now become a common tactic, often causes unpredictable side-effects. Even the so-called 'benign' boot sector viruses can easily wreak destruction when they attempt to infect disks (both hard and floppy) with which their authors are not familiar. For example, the Evil Empire virus saves its code in sectors 1, 6 and 7 of Track 0, Head 0, and this use of the 'spare' sectors is likely to cause serious corruption on an increasing number of machines.

KNOWN BOOT SECTOR VIRUSES

Viruses which store the original DOS Boot Sector or Master Boot Sector in a fixed location. Note: DOS Boot Sector locations (Head, Track, Sector) are for diskettes as these locations on hard disk are variable.

	DOS Boot Sector (Head, Track, Sector)	**Master Boot Sector** (hard disk)
Aircop	(1,39,9) [1]	-
Azusa	(1,39,8)	Overwrites
Beijing	(1,0,3)	(0,0,6)
Den Zuk	(0,40,33)	-
E.D.V.	(0,39,8)	-
Evil Empire A	(1,0,2)	(0,0,6)
Evil Empire B	(1,0,3)	(0,0,3)
Filler	(0,40,7)	-
Korea	(1,0,3)	-
Leszop	(1,0,3)	(0,0,6)
New Zealand	(1,0,3)	(0,0,7) or (0,0,2)
Ohio, Hacker	(0,40,33)	-
PrintScreen	(1,0,3)	-
Yale	(0,39,8)	-

Viruses which store the DOS Boot Sector or Master Boot Sector in a variable location.

Form	variable	-
Joshi	variable	(0,0,9)
Music Bug	variable	-

Viruses which store the DOS Boot Sector in sectors marked as 'bad' in the FAT. They do not infect the Master Boot Sector.

Brain [1], **Italian**, **Typo**, **Disk Killer**

Multi-Partite Viruses which infect boot sectors and programs.

Anthrax	-	variable
AntiCAD	variable	-
Flip	-	variable
Spanish Telecom	variable	(0,0,7)
Tequila	-	variable
V1	variable	variable

Boot sector viruses awaiting analysis.

Swap, **X-BOOT/Trackswap, Mardi Bros**, **Microbes**

[1] Virus infects diskettes only

Disinfection

Removing a Boot Sector virus can be extremely inconvenient for the average user. There are, however, a number of commercial programs which offer either specific or generic 'cures'.

The 'specific' type attempts first to recognise the particular virus involved and then to use its knowledge of where that virus stores the original boot sector, to collect the copy and return it to the proper boot sector. This works only as long as the routines which identify the virus strain are 100 percent reliable; even a single byte changed in the original virus code could result in the original boot sector being stored at a different address.

The 'generic' type of boot cure works only if the user runs the program *before* any infection has been encountered. By taking a copy of the original uninfected boot sector, the generic cure program can replace it on demand and thus disable any virus code which may have been placed there. Unfortunately, the corruption which may have occurred to the remainder of Track 0, Head 0, may not be so easily repaired. If the contents of this track were static information which did not change during normal machine operation, the problem could be solved by extending the original uninfected copying procedure to included the larger area at risk. Thus after infection, the whole track could be repaired from this copy. However, there are machines which use this track for dynamic information which is constantly changing as information is written to and from the disk. In this case, the best that can be hoped for is to copy off as much data as possible (from *all* the partitions) and then rebuild the drive structure.

Disk Editors

Removing a DOS Boot Sector virus can usually be accomplished by a reformat of the logical drive under DOS. Removing a virus from the Master Boot Sector (in the absence of a reliable disinfection program) can be attempted using a sector editing program (such as *The Norton Utilities*) to identify and then transfer the original copy of the Master Boot Sector back to Track 0, Head 0, Sector 1. In many cases this will

succeed. If it doesn't then it may be necessary to run *FDISK* to reset the Partition Table values and any other variable areas that it may maintain, remembering that most versions of *FDISK* will automatically clear the information contained in each partition.

It should be noted that the Azusa virus does not store the original Master Boot Sector anywhere. Instead the virus fulfils the most important function of the Master Boot Sector by examining the Partition Table (which the virus does not overwrite) in order to locate the bootable partition. This obviously complicates the recovery process. (See *VB*, April 1991, p.23.)

Interrupt Interception

If present in memory, boot sector viruses usually monitor any DOS requests for the boot sector and redirect these interrupt calls to the genuine boot sector - in this way DOS receives the correct information but from the *wrong* location. This interrupt interception deceives the operating system, thus rendering diagnostic software (including 'ignorant' virus scanning programs) incapable of locating the virus on disk.

It must therefore be emphasised yet again that the use of disk editors, scanning software or other diagnostic tools should only commence after the infected PC has been booted from a **clean write-protected system diskette**. This enables the trusted copy of DOS to gain *genuine* access to the hard disk and an appropriate utility can then be used to restore the affected boot sector.

Caution

Only as a *last resort* should a low-level format be attempted, and only then by someone who knows *exactly* what is being done and what other actions (such as the high-level format) are necessary to rebuild the disk structure. Before commencing a low-level format, all data files should be backed-up (preferably twice) and verified. Software should be restored from write-protected master copies.

Finally...

The best protection against virus activity is regular verified back-ups.

RECOVERY FROM MOST BOOT SECTOR VIRUS INFECTIONS IS EASY IF CLEAN WRITE-PROTECTED BACK-UPS OF THE MASTER BOOT SECTOR AND DOS BOOT SECTOR ARE AVAILABLE!

Back-ups of these vital sectors can be taken using disk editing software such as the *The Norton Utilities.*

LETTERS

Virus Bulletin Signatures With IBM's Virscan

16th May 1991

Dear Edward,

Sue Roll at *HFC Bank* was in the process of preparing an ADDENDA.LST of *Virus Bulletin* signatures before last month's issue (May 1991) came out advocating this approach. Well done Sue for all her hard work. As we had some reports of false alarms with the *Virus Bulletin* Kamikaze signature I arranged for a test of all the signatures against a large repository of PC executable file in the *IBM High Integrity Laboratory* at the *Thomas Watson Research Center*.

Bill Arnold at *IBM Yorktown* has tested the *Virus Bulletin* signature file against our false positives test machine and found that some search strings are susceptible to false alarms when they are used as *VIRSCAN* signatures because of *VIRSCAN*'s ability to check for 'mutants' (See *VB*, May 1991, pp. 16-17). He advises that the 'no mutants' keyword should be used for signatures that use these strings, along with IBM *VIRSCAN* 1.3 or 2.00.01 or later. The Vienna (6) and Kamikaze signatures are particularly susceptible and should be avoided entirely.

VIRSCAN has a 'no mutants' option. On the same line in the signature as the "infects com, exe. Scan memory" statements, add the phrase 'no mutants' and *VIRSCAN* will disable mutant detection for that signature.

These are the only false positives he found:

Vienna (6) virus	(1 mismatched byte)
Fichv 2.1 virus	(4 mismatched bytes)
Internal virus	(5 mismatched bytes)
Jocker virus	(4 mismatched bytes)
Kamikaze virus	(1 mismatched byte)
Leprosy	(5 mismatched bytes)

I have enclosed a disk with the ADDENDA.LST with the *Virus Bulletin* signatures, I have edited out both the Kamikaze and Vienna (6) signatures.

You may wish to publish this letter so that people taking your advice don't get false alarms. How does *VIRSCAN* fair against your virus test-set now?

Yours sincerely,

Mark Drew
IT Security Consultant
IBM UK Ltd.

We are grateful to Mark Drew (*IBM UK Ltd*), Bill Arnold (*IBM Inc*., USA) and Sue Roll (*HFC Bank*) for their meticulous work in entering and testing *VB* search patterns and for sharing their findings with our readership.

The ability of IBM's *VIRSCAN* to detect byte changes within established search patterns (which increases the likelihood of detecting unknown virus variants) is just one of many options which makes the program so attractive. This option is also liable to cause false positives and if *VB* search patterns are incorporated it should be disabled (certainly for the six viruses listed above which are known to cause problems). The Kamikaze pattern has caused numerous false positives, which makes one question the value of trying to extract patterns for high level language viruses (see *VB*, February 1991, p.4).

Regarding Mr. Drew's enquiry about the current performance of *VIRSCAN* when used in conjunction with *VB* patterns, all that can be said is that the results (see table on page 35) speak for themselves. Ed.

Vetting Procedure

Sir,

I was very disappointed in the review of *VET* in the May edition of *Virus Bulletin*. In disregarding everything but the total number of viruses detected you are doing your readers a grave disservice. As David Chess pointed out on *Virus-L* (May 8th, 1991, see Figure 1.), few of your specimens have ever be seen in the wild and when a virus does appear in the wild for the first time it is almost always one that was previously unknown. We are enjoying considerable success with *VET* and we believe the reasons for this are:

1. The average customer is more impressed by ease of use and efficiency in recovering his infected files, than in having his time wasted by a program which searches for vast numbers of rare and endangered species of viruses he is most unlikely to get, but cannot recover programs infected by common viruses.

2. We provide a fast and responsive service when a customer does find a new virus.

3. For most of last year we have been able to detect and recover files infected with at least one virus not detected by major packages.

4. We have found that *VET* can recover infected files that the better known anti-virus programs would either fail to detect, or could not remove without destroying the infected file.

I should be grateful if you would publish my reply to your review.

Roger Riordan
Technical Manager
Cybec Pty Ltd,
Australia

[To set Mr. Riordan's letter in context, the *VET* virus scanning program was found to have the lowest detection rating (just under 10 percent) of any virus scanner so far reviewed by *VB*. (see *VB*, May 1991, pp. 18-19.)]

Editor's Reply...

Mr. Riordan raises an important point: *should the developer of a virus scanning program concentrate only upon those viruses known to be an active threat, i.e. 'in the wild'*? A growing body of opinion (among anti-virus software manufacturers) maintains that the inclusion of search routines for viruses *believed* to be 'research' samples is wasteful in terms of scanning resources. This body of opinion also maintains that viruses which become 'extinct' or which have a seriously retarded ability to spread should be excluded from virus scanning software.

This movement towards 'selectivity' has partly been enforced by deterioration in scanner 'run times' - scanners with primitive search algorithms are quite simply incapable of searching for the ever-increasing number of virus samples at an acceptable speed. (This, incidentally, does *not* apply to *VET* which has an extremely *fast* search algorithm).

It is probable that the calls for 'selectivity' will also become louder as more and more manufacturers begin to realise that the development and maintenance of virus scanning software is research-intensive and consuming in terms of programming time and energy. Companies new to the anti-virus field soon realise that anti-virus software is not market-driven, but that it is the product of exhaustive and continuing research combined with programming ingenuity. 'Selectivity' reduces this essential research and development effort - the economic corollary is that 'selectivity' actually saves money.

In fact, in the real commercial world, the product developers who become the recognised market leaders are those who *maximise* the security offered to the customer while at the same time maintaining acceptable scanning speeds. These are the companies and individuals prepared to make the essential commitment in terms of ongoing research and development.

In any case, there are a number of inherent dangers to 'selectivity' which should render it unacceptable to the end-user. Primarily, no-one is in a position to classify computer viruses as 'non-threats' - if a virus functions and has been circulated (even within the 'research community' which is a nebulous underworld in itself), then it must be classified as a potential threat and treated accordingly. The proliferation of virus Bulletin Boards, the uncontrolled exchange of new and extant virus code, the ongoing appearance of hacked virus 'clones' and the re-emergence of 'improved' viruses thought to be extinct are all factors which render 'selectivity' an extremely dangerous and ill-advised path to follow.

One thing is certain: exponential virus development will not be tackled by pretending that it is not happening.

```
This is an open note to other folks in the
anti-virus field, to see if some (potentially
significant) things that we've noticed about
(primarily PC-DOS) viruses look the same from
other peoples' perspectives. Some informal
questions to individuals suggest that these are
reasonably common observations; is there anyone
out there who would disagree with them?

1) Most viruses in the collection of anti-virus
workers have, as far as anyone knows, never
been found on an end-user system (We, for
instance, have a few hundred viruses, but know
of only about 50 that have ever bothered an
end-user.)

2) When a virus shows up on an end-user system
('in the wild', as we say) that has never been
seen on an end-user system before, it's usually
a brand new virus, rather than a virus that's
previously been in collectors' collections.
That is, it's very rare for a virus from the
'collectors only' category to move into the 'in
the wild' category.

Do these things match the experiences of other
anti-virus workers? Can anyone give some
examples of viruses that were at one time
thought to be 'collector only' but which later
showed up in the wild? (Isolated incidents,
such as the rather obvious direct 'seeding' of
an end-user machine with the Whale virus, don't
really count.)

As a sort of a spot check, has anyone ever seen
any of the 'Anti-Pascal' viruses infecting an
end-user machine? (I ask about these just
because they're prototypical 'collector only'
viruses; rather stupid and seemingly unlikely
to spread.) Dave Chess

Dave: A telling anecdote: at the Trenton Computer
fair last month, about 100 people crammed into a
room to hear about some of the new viruses. When
asked who had been infected with a virus, about
80 of the people raised their hands. I asked
those infected with Jerusalem, Stoned and Ping-
Pong to drop their hands. One hand was left,
Cascade. The loud cry for protection against
research-only viruses is quite, quite bothersome
- the numbers game we have to play (as a vendor)
in order to counter 'my scanner can beat up your
scanner' types of game is sorta foolish - yet we
must play the game. Ross Greenberg
```

Figure 1. Seasoned computer virologists Chess and Greenberg kicking off the 'selectivity' debate on *Virus-L*. The approach is pragmatic, reducing the research and development effort, and thus attractive to software developers. It might also stop the ludicrous 'numbers game' so beloved by the marketing men. However, there are associated dangers with this strategy - no functioning virus can be classified as a 'non-threat'.

KNOWN IBM PC VIRUSES

This is a list of the known viruses affecting IBM PC/XT/AT/PS2 and compatibles. The first part of the list gives aliases and brief descriptions of viruses which have been seen, while the second part lists viruses which have been reported. Each entry consists of the virus group name, its aliases and the virus type (See 'Type codes' table). This is followed by a short description (if available) and a 10 to 16 byte hexadecimal pattern which can be used to detect the presence of the virus by the 'search' routine of disk utility programs such as The Norton Utilities or your favourite disk scanning program. Offset (in hexadecimal) normally means the number of bytes from the virus starting point. For parasitic viruses, the infective length (the amount by which the length of an infected file has increased) is also given.

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 on disk) **N** = Not memory-resident after infection

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

SEEN VIRUSES

8 Tunes - CER: The virus probably originates in Germany and infects COM and EXE files. The length of the virus code is 1971 bytes. When triggered, it will play one out of eight different tunes. The virus attempts to deactivate two anti-virus programs: Bombsquad and Flushot+.

```
8 Tunes          33F6 B9DA 03F3 A550 BB23 0353 CB8E D0BC; Offset variable
```

10-past-3 - CR: A 748 byte virus which is awaiting analysis.

```
10 past 3        B840 008E D8A1 1300 B106 D3E0 2D00 088E; Offset 069
```

268-Plus - CN: When this virus is run it will infect all COM files in the current directory increasing the first one by 240 bytes, the second by 269 bytes, the third by 270 bytes and so on. The virus is encrypted and is awaiting analysis.

```
268-Plus         8EC1 0650 BE00 0156 31FF B90B 01F3 A4BD; Offset 005
```

432 - C?: Virus awaiting disassembly.

```
432              50CB 8CC8 8ED8 E806 00E8 D900 E904 0106; Offset 076
```

483 - CER: This virus does not work properly, as infected programs will never run. As this could be fixed by a minor correction, a search pattern for the current version is provided.

```
483              0256 5AB9 1800 F614 46E2 FBCD 215E 81BC; Offset 181
```

535A - CN: A mutation of the Vienna virus. Second generation copies do not appear to replicate.

```
535A             ACB9 0080 F2AE B904 00AC AE75 EEE2 FA5E; Offset 040
```

555, Dutch 555 - CER: A compact 555 byte virus awaiting analysis. It does not seem to do anything apart from replicating.

```
555              5B58 072E FF2E 0500 813E 1200 4D5A 7406; Offset 19E
```

777 Revenge - CR: Virus awaiting analysis.

```
777 Revenge      B8FF FF33 C9CD 2183 F906 7243 B856 0250; Offset 0
```

800 - CR: Infective length is 800 bytes. The virus code is written into a random location in the infected file. Like Number of the Beast, it uses an undocumented DOS function to obtain the original INT 13H address, and instead of intercepting INT 21H, it intercepts INT 2AH, function 82H. The virus is encrypted. (*VB* June 90)

```
800              B981 0151 AD33 D0E2 FB59 3115 4747 E2FA; Offset 00E
```

905 - ER: A Bulgarian virus, still awaiting analysis.

```
905              488E C08E D880 3E00 005A 7415 0306 0300
```

928 - CER: Virus awaiting disassembly.

```
928              E9AD 00B8 BBBB CD21 3D69 6974 03E8 3500; Offset 020
```

1024PrScr - CR: This virus increases the length of infected programs by 1024 bytes. The main side-effect is to perform a Print Screen function at different times.

```
1024PrScr       8CC0 488E C026 A103 002D 8000 26A3 0300; Offset 033
```

1028 - CER: Virus is 1028 bytes long. Awaiting analysis.

```
1028            0606 005E 561E 0E33 FF8E DFC5 0684 002E; Offset 0E9
```

1067 - CR: This virus is closely related to the Ambulance virus, but is still awaiting analysis.

```
1067            018A 5405 8816 0001 B42A CD21 F6C2 0175
```

1077 - CER: This 1077 byte virus infects COM and EXE files, but is unable to infect EXE files larger than 64K.

```
1077            4E01 EACD 21C3 B44F CD21 C351 33C0 3B86
```

1226 - CR: This Bulgarian virus is related to Phoenix, Proud and Evil. As in the case of its relatives, no search pattern is possible.

1260, V2P1 - CN: Virus infects COM files, adding 1260 bytes to them. The first 39 bytes contain code used to decrypt the rest of the virus. A variable number of short (irrelevant) instructions are added between the decoding instructions at random in an attempt to prevent virus scanners from using identification strings. An infected file has the seconds field set to 62. No search pattern is possible. (*VB* Mar 90)

1575 - CER: Virus awaiting analysis. Infected files grow by 1576-1593 bytes.

```
1575            D087 ECBE 3C01 BF00 00B9 1000 FCF2 A4E9; Offset 18C
```

1600 - CER: A 1600 byte Bulgarian virus, reported to be written by the same author as the Nina, Terror and Anti-Pascal viruses. Many infected programs, including COMMAND.COM will fail to execute when infected.

```
1600            8B35 8936 0001 8B75 0289 3602 01C7 4514
```

2100 - CER: This is a Bulgarian virus, related to the Eddie and Eddie-2 viruses and contains extensive segments of code in common with both. The previously published pattern for Eddie-2 can be found within this virus, but they can easily be differentiated on the basis of length.

2144 - CER: A 2144 byte Russian virus which may totally disable the hard disk when it activates. A computer with a disabled disk cannot be rebooted from a system floppy disk without disconnecting the hard disk.

2480 - CR: This virus only spreads if the year is set to 1988, so it is not a serious threat. It is rather long, 2480 bytes, but has not been analysed yet. This virus first appeared in Finland.

```
2480            81C6 0301 01C6 B904 008C C88E C08E D8BF
```

3445 - CER: This 3445 byte encrypted virus has not been fully analysed, but infected programs often fail to execute.

```
3445            D2BB 1000 F7E3 03C1 83D2 00F7 F359 50B8; Offset 034
```

5120 - CEN: This is one of the largest viruses known, 5120 bytes long. When an infected program is run, it will search recursively for EXE and COM files to infect. Infected programs will terminate with an 'Access denied' message after 1st June 1992. Parts of the virus seem to have been written in compiled BASIC.

```
5120            40B1 04D3 E88C DB03 C305 1000 8ED8 8C06; Offset 026
```

4K, 4096, Frodo, IDF, Israeli Defence Forces - CER: Infective length is 4096 bytes. The virus may occasionally cause damage to files, as it manipulates the number of available clusters, which results in crosslinked files. If the virus is resident in memory, it disguises itself from detection by pattern-searching or checksumming programs. Infected systems hang on 22nd September. (*VB* May 90, Nov 90)

```
4K              E808 0BE8 D00A E89A 0AE8 F60A E8B4 0A53; Offset 239
```

Advent - CEN: An old 2764 byte mutation of Syslock, which is detected by the Syslock pattern. It was not included earlier, because of problems in getting samples to replicate. This virus activates in December and plays a Christmas tune.

Aircop - DR: Virus displays the blinking message '.Red State, Germ offensing —Aircop' after infecting every eighth floppy disk. Originated in Taiwan. (*VB* Feb 91).

```
Aircop          32E4 CD16 CD12 33C0 CD13 0E07 BB00 02B9; Offset 0C4
```

Agiplan - CR: Infective length is 1536. The virus attaches itself to the beginning of COM files. Agiplan has only occurred on one site and may be extinct.

```
Agiplan         E9CC 0390 9090 9090 9C50 31C0 2E38 26DA; Offset 0 (?)
```

AIDS - CN: Not to be confused with the AIDS Trojan, this virus overwrites COM files and is about 12K long. When an infected program is executed, the virus displays 'Your computer now has AIDS' and halts the system.

```
AIDS            0600 AE42 6E4C 7203 4600 0004 00A0 1000; Offset 2C7F
```

AIDS II - PN: A 'companion' virus, 8064 bytes long, which displays a message when it activates. To locate and remove the virus, search

for COM files corresponding to EXE files, but marked 'Hidden' and located in the same subdirectory and delete them.

```
AIDS II          5589 E581 EC02 02BF CA05 0E57 BF3E 011E; Offset 014
```

Akuku - CER: 889 byte virus, probably written by the same author as the Hybrid virus.

```
Akuku            E800 005E 8BD6 81C6 2A01 BF00 01A5 A481; Offset 24E
```

Alabama - ER: Infective length is 1560 bytes. May cause execution of wrong files and FAT corruption.

```
Alabama          803D C673 0726 C605 CF4F EBF0 26FF 0603
```

Ambulance, RedX - CN: The major effect of this virus is to display a moving ambulance with the sound of a siren. The virus is 796 bytes long.

```
Ambulance        0001 8A07 8805 8B47 0189 4501 FFE7 C3E8; Offset 016
```

Amoeba - CER: Virus adds 1392 bytes to the length of the infected files. It does not have any known side-effects.

```
Amoeba           CF9C 502E A107 0140 2EA3 0701 3D00 1072; Offset 0D1
```

Amstrad - CN: Adds 847 bytes to the front of any COM file in the current directory. The rest contains an advertisment for Amstrad computers. (*VB* June 90). Cancer is a 740 byte long mutation, which infects the same files repeatedly. These viruses are members of the Pixel family.

```
Amstrad          C706 0E01 0000 2E8C 0610 012E FF2E 0E01; Offset 114
```

Amstrad-852 - CN: Almost identical to the original 847 byte version, with only a text string changed.

Amstrad-877 - CN: This mutation is 877 bytes long, and detected by the 'Amstrad' pattern.

Anthrax - MCER: An interesting, multi-partite virus from Bulgaria, which infects the Master Boot Sector, as well as executable files. Infected files usually grow by 1000-1200 bytes.

```
Anthrax          0E1F 832E 1304 02CD 12B1 06D3 E08E C0BF; Offset 0 in MBR
```

AntiCAD, Plastique - CER: This is a family of 7 viruses from Taiwan, based on the Jerusalem virus, but considerably modified. This group includes a 2900 byte mutation, a 3012 byte mutation and four 4096 byte mutations. Two of these four are known as 'Invader' and one as 'HM2'. The four 4096 byte mutations will also infect the boot sector. The Plastique virus triggers when ACAD.EXE (the AUTOCAD (tm) program) is executed. Drives A and B are checked for the presence of a disk which, if found, has head 0 of all tracks overwritten with random data. An 'explosion' routine (speaker noise generated every 4.5 minutes) then commences. The first and second hard disks are overwritten on all heads and tracks.

```
AntiCAD (1)      B840 4BCD 213D 7856 7512 B841 4BBF 0001; Offset 0
AntiCAD (2)      C08E D8A1 1304 B106 D3E0 8ED8 33F6 8B44
```

AntiCAD 2576 - CER: A mutation of the AntiCAD series. This 2576 byte virus is closely related to the 2900 byte mutation.

```
AntiCAD 2576     595B 5807 1F9C 2EFF 1E3B 001E 07B4 49CD; Offset 550
```

AntiCAD/Plastique 3004 - CER: Very closely related to the 3012 byte mutation of Plastique. The virus contains the text string 'COBOL' and is detected by the AntiCAD (1) pattern.

Anti-Pascal (1) - CN: Two Bulgarian viruses 529 and 605 bytes long which add their code in front of infected programs. They are targeted against Turbo-Pascal, and delete .PAS and .BAK files.

```
Anti-Pascal (1)  D1E0 D1E0 80E4 0380 C402 8AC4 8BD8 32FF; Offset variable
```

Anti-Pascal (2) - CN: A second group of Bulgarian viruses written by the author of Anti-Pascal (1) viruses. There are three viruses which belong to this group and their infective lengths are 400, 440 and 480 bytes. They are structurally different from Anti-Pascal (1) since they add their code to the end of infected files. The side-effects are similar since they may delete .PAS, .BAK and .BAT files.

```
Anti-Pascal (2)  21BE 0001 5A58 FFE6 50B4 0E8A D0CD 2158; Offset variable
```

Apocalypse - CER: Slight mutation of the Jerusalem virus. Detected by the Jerusalem-USA pattern.

Apocalypse II - CER: Slight mutation of the Eddie-2000 virus. Detected by the Dark Avenger pattern.

Arf - CN: A 1000 byte mutation of the Violator virus. Displays 'Arf Arf! Got you!' activated. Detected by the 'Violator' pattern.

Armagedon - CR: A 1079 byte virus from Greece, which interferes with the serial port. It will produce control strings for Hayes-compatible modems, dialling number 081-141 (speaking clock in Crete). Virus name is spelt with a single 'd'.

```
Armagedon        018C CBEA 0000 0000 8BC8 8EDB BE00 01BF; Offset 3F0
```

Attention - CR: A Russian, 394 byte virus. The virus has some code in common with the 'Best Wishes' virus, which is possibly written by the same author. Infective length is 393 bytes and only files longer than 786 bytes are infected. Disk writing is done by outputting directly to hardware via port 3F2H.

```
Attention        B000 8BDA B501 433A 0775 FB4B 4B81 275F
```

Australian 403 - CR: Destructive, overwriting 403 byte virus which has no side-effects other than destroying the programs it infects.

```
Australian 403   8C06 5B01 8CC8 8ED8 B821 25BA 9401 CD21; Offset 011
```

Azusa - DR: A short boot sector virus, which may damage data on diskettes larger than 360K. When it activates, it will disable COM1: and LPT1:. (*VB* April 91).

```
Azusa            B908 27BA 0001 CD13 72F1 0E07 B801 02BB; Offset 0EA
```

Backtime - CR: A 528 byte virus which is awaiting analysis.

```
Backtime         2125 CD21 8CC8 8ED8 8EC0 58BB 0001 53C3; Offset 1F8
```

Bad Boy - CR: A 1001 byte virus, which may have been written by the same author as the 'Boys' virus, but is structurally different. Awaiting analysis.

```
Bad Boy          0175 0383 C302 5351 8B07 8B4F 108B D830; Offset variable
```

Bandit - EN: This 2653 byte virus is detected by the 'Old Yankee' pattern. Awaiting analysis.

Bebe - CN: A Russian, 1004 byte virus.

```
Bebe             B104 D3EB 240F 3C00 7401 4389 1E0C 00C7
```

Best Wishes - CR: A 1024 byte Russian virus containing the message 'This programm ... With Best Wishes!'. Many programs, including COMMAND.COM will not work properly if infected with this virus.

```
Best Wishes      4C00 268C 1E4E 0007 1FB8 0400 8BF5 81EE
```

Big Joke - CN: A Norwegian virus awaiting full disassembly. Infectious length is 1068 bytes. Contains text: At last..... ALIVE !!!!! I guess your computer is infected by the Big Joke Virus. Release 4/4-91 Lucky you, this is the kind version. Be more careful while duplicating in the future. The Big Joke Virus, killer version, will strike harder. The Big Joke rules forever

```
Bigjoke          8BE8 83C5 030E 588E D88E C08D 7643 BF00; Offset 1EB
```

Black Monday - CER: This virus was first isolated in Fiji, but may have been written elsewhere. It adds 1055 bytes to infected files. The name is derived from the message 'Black Monday 2/3/90 KV KL MAL'. Infected EXE files cannot be disinfected, as the virus will overwrite a few bytes at the end of the file.

```
Black Monday     8B36 0101 81C6 0501 8B04 8B5C 02A3 0001
```

Bljec - CN: A family of small viruses, which are awaiting analysis. The following mutations are known: Bljec-3 (231), Bljec-4 (247), Bljec-5 (267), Bljec-6 (270), Bljec-7 (287), Bljec-8 (358) and Bljec-9 (369)

```
Bljec            B980 00BE 7FFF BF80 00F3 A4B8 F3A4 A3F9; Offset variable
```

Blood - CN: A simple virus from Natal, South Africa. The 418 byte virus does nothing of interest, apart from replicating.

```
Blood            1E0E 1FB4 19CD 2150 B202 B40E CD21 B41A; Offset 07F
```

Beijing, Bloody! - MR: A primitive 512-byte virus. On 129th boot and every sixth boot thereafter, the virus displays the message 'Bloody! Jun. 4, 1989'. The virus is believed to be a protest against the Tianamen Square massacre. (*VB* Feb 91).

```
Beijing          80FC 0272 0D80 FC04 7308 80FA 8073 03E8; Offset 01F
```

Boys - CN: A 500 byte virus containing the text 'The good and the bad boys'. Awaiting analysis.

```
Boys             BE01 01AD 0503 0050 8BF0 BF00 01B9 0500; Offset 042
```

Brain, Ashar, Shoe - DR: Consists of a bootstrap sector and 3 clusters (6 sectors) marked as bad in the FAT. The first of these contains the original boot sector. In its original version it only infects 360K floppy disks and occupies 7K of RAM. It creates a label '(c) Brain' on an infected disk. There is a variation which creates a label '(c) ashar'.

```
Brain            FBA0 067C A209 7C8B 0E07 7C89 0E0A 7CE8; Offset 157
```

Burger - CN: This primitive 560 byte virus overwrites infected files, which makes it easily detectable. Several mutations with slightly different lengths are known. (*VB*, May 91)

```
Burger 1         B447 0401 508A D08D 3646 02CD 2158 B40E; Offset 01B
Burger 2         CD21 B43E CD21 2E8B 1E00 E081 FB90 9074; Offset variable
```

Burger 382 - CN: Simple overwriting virus from Taiwan which overwrites part of the program.

```
Burger 382       B417 8D16 5502 CD21 3CFF 7514 B42C CD21; Offset 01C
```

Burger 405 - CN: Infects one COM file (on a different disk) each time an infected program is run by overwriting the first 405 bytes. If the length of the file is less than 405 bytes, it will be increased to 405. The virus only infects the current directory and does not recognise previously infected files.

```
Burger 405       26A2 4902 26A2 4B02 26A2 8B02 50B4 19CD; Offset 00A
```

CARA - CR: A 1025 byte virus. Awaiting analysis.

```
CARA             812E 0200 C000 B44A BB00 B0CD 2181 EBC0; Offset 029
```

Carioca - CR: This virus adds 951 bytes to the end of infected programs, but it has not been analysed yet.

```
Carioca          01FC F3A4 B800 0150 C32E 8B1E 0301 81C3
```

Cascade, Fall, Russian, Hailstorm - CR: This encrypted virus attaches itself to the end of COM files, increasing their length by 1701 or 1704 bytes. The encryption key includes the length of the infected program, so infected files of different lengths will look different. After infection it becomes memory-resident and infects every COM file executed, including COMMAND.COM. The original version will produce a 'falling characters' display if the system date is between 1st October and 31st December 1988. The formatting version will format the hard disk on any day between 1st October and 31st December of any year except 1993. Both activations occur a random time after infection with a maximum of 5 minutes. (*VB* Sept 89)

```
Cascade (1) 01   0F8D B74D 01BC 8206 3134 3124 464C 75F8; Offset 012, 1701 bytes, Falling characters
Cascade (1) 04   0F8D B74D 01BC 8506 3134 3124 464C 75F8; Offset 012, 1704 bytes, Falling characters
Cascade (1) Y4   FA8B CDE8 0000 5B81 EB31 012E F687 2A01; Offset 000, 1704 bytes, Falling characters
Cascade format   0F8D B74D 01BC 8506 3134 3124 464C 77F8; Offset 012, 1704 bytes, Formats hard disk
```

Cascade YAP - CR: A mutation of Cascade with a slightly modified encryption routine.

```
Cascade YAP      0F8D B74D 01BC 8206 3124 3134 464C 75F8; Offset 013
```

Casino - CR: Virus infects COM files smaller than 62905 bytes and when triggered the virus destroys the FAT and then offers to play the Jackpot game. If you win, it reconstructs the FAT, while if you lose, the machine hangs. The virus triggers on 15th January, 15th April and 15th August of any year. (*VB* Mar 91).

```
Casino           594B 7504 B866 06CF 80FC 1174 0880 FC12; Offset ?
```

Casper - CN: This virus was written by Mark Washburn and uses the same encryption method as the 1260 virus. The infective length is 1200 bytes. The virus sets the seconds field to 62. The source code for this virus has been widely circulated and it includes a 'manipulation task' (payload) which will format cylinder 0 of the hard disk. No search pattern is possible.

Cemetery - ER: A 1417 byte mutation of the Murphy virus. Detected by the Murphy(2) pattern.

Christmas in Japan - CN: A 600 byte virus from Taiwan, which will activate on 25th December, and display the message 'A merry christmas to you'.

```
Christmas Japan  32E4 CF8A 1446 80F2 FE74 06B4 06CD 21EB; Offset 23F
```

Christmas Tree, Father Christmas, Choinka - CN: This is a Polish 1881 byte version of the Vienna virus, which only activates from 19th December to the end of the year and displays a 'Merry Christmas' message. Damage to files has been reported, but not confirmed. This virus is also detected by the Vienna (4) string.

```
Christmas Tree   CD21 81FA 130C 7308 81FA 0101 7202 EB0E
```

Christmas Violator - CN: A 5302 byte mutation of the Violator virus.

```
Xmas Violator    11AC B900 80F2 AEB9 0400 ACAE 75ED E2FA; Offset 1EB
```

Cinderella - CR: The name of this 390 byte virus is derived from the text 'cInDeReL.la' contained within the virus. After a certain number of keystrokes, the virus creates a hidden file, and jumps to a location in ROM, which caused a cold-boot on a test machine.

```
Cinderella       FA0E 1FBE 8A03 BF90 00AD 8905 AD89 4502; Offset 131
```

Cookie - CER: This 2232 byte virus may display the message 'I want a COOKIE!', and wait for input from the user. It is closely related to the Syslock/Macho/Advent viruses, and is identified by the Syslock string.

Crazy Eddie - CER: A 2721 byte virus which has not been fully analysed.

```
Crazy Eddie      0653 B803 01CF 813C 4D5A 7404 813C 5A4D; Offset 0A0
```

Damage - CER: Two related viruses 1063 and 1110 bytes long which cause 'Sector not found' errors by reformatting selected areas of disks. Detected by the 'Diamond' pattern.

Dark Avenger - CER: The virus infects when a file is opened and closed as well as when it is executed. This means that a virus-scanning program will cause it to infect every program scanned. Infective length is 1800 bytes. It only infects if a program is at least 1775 bytes long and it may overwrite data sectors with garbage. A mutation which extends the file by 2000 bytes. (*VB* Feb 90)

```
Dark Avenger     A4A5 8B26 0600 33DB 53FF 64F5 E800 005E; Offset variable
```

Darklord - CER: A mutation of the Terror virus, this 921 byte virus contains the string 'Dark Lord, I summon thee! MANOWAR'. Awaiting further analysis.

```
Darklord         8EC0 488E D88B 1E03 008 3EB6 503C 326A3; Offset 096
```

Darth Vader - CR: A family of small viruses, probably from Bulgaria. Some of the 4 known mutations contain code which will only work on '286 and above. Awaiting analysis.

```
Darth Vader      B820 12CD 2F26 8A1D B816 12CD 2F; Offset variable
```

Datacrime - CN: The virus attaches itself to the end of a COM file, increasing its length by 1168 or 1280 bytes. On execution of an infected program, the virus searches through the full directory structure of drives C, D, A and B for an uninfected COM file which will be infected. Files with 7th letter D will be ignored (including COMMAND.COM). If the date is on or after 13th October of any year, the first 9 tracks of the hard disk will be formatted after displaying the message:

```
DATACRIME VIRUS
RELEASED: 1 MARCH 1989
```

This message is stored in an encrypted form in the virus. (*VB* Aug 89)

```
Datacrime (1)    3601 0183 EE03 8BC6 3D00 0075 03E9 0201; Offset 002, 1168 bytes
Datacrime (2)    3601 0183 EE03 8BC6 3D00 0075 03E9 FE00; Offset 002, 1280 bytes
```

Datacrime II - CEN: This encrypted virus attaches itself to the end of a COM or EXE file, increasing their length by 1514 bytes. The virus searches through the full directory structure of drives C, A and B for an uninfected COM or EXE file. It ignores any file if the second letter is B. If the date is on or after 13th October of any year, but not a Monday, a low level format of the first 9 tracks will be done on the hard disk after displaying the message: 'DATACRIME II VIRUS' which is stored in an encrypted form. Datacrime IIB displays the message '* DATACRIME *'. (*VB* Aug 90)

```
Datacrime II     2E8A 072E C605 2232 C2D0 CA2E 8807 432E; Offset 022, 1514 bytes
Datacrime IIB    2BCB 2E8A 0732 C2D0 CA2E 8807 43E2 F3; Offset 01B
```

Datalock - CER: The name of this 920 byte virus is included at the end of infected programs, but its effects are not known yet.

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

dBASE - CR: Transposes bytes in dBASE (DBF) files. Creates the hidden file BUGS.DAT in the root directory of drive C and generates errors if the absolute difference between the month of creation of BUGS.DAT and the current month is greater than or equal to 3. Infective length is 1864 bytes. The destroy version destroys drives D to Z when the trigger point is reached. (*VB* Dec 89)

```
dBASE            50B8 0AFB CD21 3DFB 0A74 02EB 8A56 E800; Offset 636, 1864 bytes
dBASE destroy    B900 01BA 0000 8EDA 33DB 50CD 2658 403C; Offset 735, 1864 bytes
```

DBF Blank - CER: This virus waits for a dBASE (DBF) file to be opened and returns a blank record once every 20 disk reads. Only one DBF file is affected at a time. Infective length is 1075 bytes.

```
DBF Blank        F3A4 C38C C02E 0344 1A05 1000 502E FF74
```

December 24th - ER: A mutation of the Icelandic (3) virus. It will infect one out of every 10 EXE files run, which grow by 848-863 bytes. If an infected file is run on December 24th, it will stop any other program from running and display the message 'Gledileg jol' (Merry Christmas in Icelandic).

```
December 24th    C606 7E03 FEB4 5290 CD21 2E8C 0645 0326; Offset 044
```

Deicide - CN: A primitive 666 byte overwriting virus. When it activates, it will wipe out the first 80 sectors on drive C:. According to a message inside the virus, it is written by a person named Glenn Benton.

```
Deicide          3C00 7502 FEC0 FEC0 3C03 7516 B002 BB00; Offset 0DC
```

Demon - CN: A primitive 272 byte overwriting virus, written by the person calling himself 'Cracker Jack'.

```
Demon            02EB 02EB EFB4 2ACD 213C 0274 04B4 4CCD; Offset 01B
```

Den Zuk, Search - DR: The majority of the virus is stored in a specially formatted track 40, head 0, sectors 33 to 41. When Ctrl-Alt-Del is pressed, the virus intercepts it and displays 'DEN ZUK' sliding in from the sides of the screen. This does not happen if KEYBUK or KEYB is installed. Den Zuk will remove Brain and Ohio and replace them with copies of itself.

```
Den Zuk (1)      FA8C C88E D88E D0BC 00F0 FBE8 2600 33C0; Offset 02C
Den Zuk (2)      FA8C C88E D88E D0BC 00F0 FBB8 787C 50C3; Offset 02C
```

Destructor - CER: A 1150 byte Bulgarian virus containing the string 'DESTRUCTOR V4.00 (c) 1990 by ATA'.

```
Destructor       5255 FBCB 3D00 4B74 1980 FC3D 740F 80FC
```

Devil's Dance - CR: A simple virus which infects COM files, adding 951 bytes at the end of infected files. The virus is believed to have originated in Spain or Mexico. It monitors the keyboard and will destroy the FAT after 5000 keystrokes.

```
Devil's Dance    B800 0150 8CC8 8ED8 8EC0 C306 B821 35CD; Offset 011
```

Diabolik - CER: A 1171 byte mutation of the Murphy virus. Detected by the Murphy 2 pattern.

Diamond, 1024 - CER: A Bulgarian virus, possibly written by the person calling himself (?) 'Dark Avenger'. This virus may be an earlier version of the Eddie virus. No side-effects or activation dates have been found. Diamond-B is a minor mutation.

```
Diamond          00B4 40CD 2172 043B C174 01F9 C39C 0EE8; Offset 170
```

Diamond-1173, David - CER: A modification of the Diamond-B virus, produced by inserting NOP instructions and making other minor changes. Contains errors which will generally cause infected COM files to crash. Detected by the 'Diamond' pattern.

Dir - CR: A 691 byte Bulgarian virus, which only infects files when the DIR command is issued. No other effects have been found.

```
Dir             CD26 0E1F 580E 1FBE 0001 56C3 0E0E 1F07; Offset 04A
```

Discom - CR: A 2053 byte mutation of the Jerusalem virus. Awaiting analysis.

```
Discom          57CD 2172 1F8B F18B FAB8 0242 B9FF FFBA; Offset 139
```

Diskjeb - CER: A disk-corrupting virus with an infective length of 1435 bytes (COM) and 1419 bytes (EXE). Only infects COM files longer than 1000 bytes and EXE files longer than 1024 bytes. In October, November and December disk writes will be intercepted and corrupted. A mutation of the Tenbyte virus.

```
Diskjeb         5351 061E 9C8C C88E D8E8 5D00 803E 4903; Offset 4E8
```

Disk Killer, Ogre - DR: The virus infects floppy and hard disks and if the computer is left on for more than 48 hours, it will encrypt the contents of the bootable disk partition. The infection of a disk occurs by intercepting a disk read - INT 13H function 2. When the virus triggers, it displays the message 'Disk Killer — Version 1.00 by Ogre Software, 04/01/1989. Warning !! Don't turn off the power or remove the diskette while Disk Killer is Processing!'. A mutation (Disk Killer 2) assembled with a different assembler has been found. (*VB* Jan 90)

```
Disk Killer     2EA1 1304 2D08 002E A313 04B1 06D3 E08E; Offset 0C3
Disk Killer 2   7423 2E3A 16F4 0175 EE2E 3A36 F501 75E7
```

Do-nothing - CR: A badly-written virus from Israel that assumes a 640K system.

```
Do nothing      8CCA 8EDA BA00 988E C2F3 A41E B800 008E; Offset 020
```

Doom2 - CER: This 1252 byte virus is not always able to infect files. The machine hangs immediately after a file is infected.

```
Doom2           803E 0A01 4574 052E 033E 0301 2E30 0547; Offset 017
```

Doom II-B - CER: This mutation of Doom-2 has not been able to replicate under test conditions - infected programs hang or overwrite the FAT and root directory on drive C. Version B uses the same encryption method as the other known mutation.

```
Doom-II-B       803E 0901 4574 052E 033E 0301 2E30 0547; Offset 01A
```

Dot Killer - CN: This 944 byte Polish virus will remove all dots (.) from the screen when they are typed. The effect can be disabled by typing a caret '^'. Seconds field is set to 62. Files set to Read-Only will not be infected.

```
Dot Killer      582E A301 0158 2EA2 0001 B800 01FF E0B8
```

Durban, Saturday 14th - CER: Adds 669 bytes to the end of infected files. On any Saturday 14th the first 100 logical sectors of drives C, then B and then A are overwritten.

```
Durban          B911 00A4 E2FD B4DE CD21 80FC DF74 47C6; Offset 02F
```

Dyslexia, Solano - CR: Virus adds 1991 bytes in front of the infected file and 9 bytes at the end. Occasionally transposes two adjacent characters on the screen.

```
Dyslexia        B4C0 CD21 3D34 1275 0E2E 8B0E 0301 1E07
```

Eddie-2, 651 - CER: A non-destructive virus from Bulgaria. It marks infected files with a value of 62 in the seconds field of the timestamp, which makes them immune from infection by Vienna or Zero Bug. Infected files grow by 651 bytes, but this will not be seen if a DIR command is used - the virus intercepts the find-first and find-next functions, returning the correct (uninfected) length. (*VB* June 90)

```
Eddie-2         D3E8 408C D103 C18C D949 8EC1 BF02 00BA; Offset 02D, 651 bytes
```

Eddie-1801 - CER: A minor mutation of the Eddie (Dark Avenger) virus, one byte longer and detected by the same pattern.

E.D.V. - DR: E.D.V. marks infected disks with 'EV' at the end of the boot sector and stores the original boot sector code in the last sector of the last track on 360K disks, just like the Yale virus. Program crashes and data loss have been reported on infected systems.

```
E.D.V.          0C01 5083 EC04 B800 01CF B601 B908 2751; Offset 0C1
```

Enigma - ER: A mutation of the 'Old Yankee' virus, claiming to have been written by the same author as HIV. It is 1624 bytes long, and is detected by the pattern published for Old Yankee.

Erasmus - CER: A 1682 byte version of the Murphy virus. Detected by the HIV pattern.

ETC - CN: A 700 byte virus, containing the text 'Virus, (c) ETC'. Awaiting analysis.

```
ETC             8B16 0201 83C2 33CD 2172 CD89 D68B 043D; Offset 061
```

Evil - CR: This is a close relative of the Bulgarian Phoenix virus, but is shorter, 1701 bytes instead of 1704. It uses the same encryption method, which makes the extraction of a search pattern impossible.

Evil Empire - MR: Virus infects Master Boot Sector and relocates original boot sector to Sector 6, Head 0, Track 0. Virus displays a text message questioning the United States' involvement in the recent Gulf War. (*VB* May 91)

```
Evil Empire     734C 80FC 0275 4731 C08E D880 3E6C 0416; Offset 01E in MBS
```

Evil Empire B - MR: An encrypted mutation, probably written by the same author as Evil Empire.

```
Evil Empire B    8CC8 8ED8 8EC0 BF05 00B9 9A01 FC8A 0504; Offset 19F
```

Faust, Spyer - CER: Infects on calling the Load-and-Execute function, but does not infect COMMAND.COM. On 13th day of every month the virus displays the message 'Chaos!!! Another Masterpiece of Faust...' and the machine hangs. The virus also writes random garbage to disk. Infective length is 1184 bytes. (*VB* Feb 91).

```
Faust            B87A 0050 06B8 FD00 5026 C706 FD00 F3A4; Offset 044
```

Fellowship - ER: This 1019 byte virus attaches itself to the end of EXE files, damaging them by overwriting the last 10 bytes or so. Other effects are being analysed.

```
Fellowship       BAF5 02E8 3A00 B60A E84A 00BA 1403 E82F; Offset 389
```

Fichv 2.1 - CN: A 903 byte encrypted virus, which contains the text 'FICHV 2.1 vous a eu'. Awaiting analysis.

```
Fichv            B801 35CD 218C 0602 0189 1E04 01B8 0335; Offset 015
```

Filler - DR: A Hungarian virus with unknown effects.

```
Filler           CD12 BB40 00F7 E32D 0010 8EC0 BA00 00EB; Offset 074
```

Finger - CER: A 1172 byte version of the Murphy virus. Detected by the Murphy-2 pattern.

Fish 6 - CER: A partial mutation of 4K having an infective length of 3584 bytes. The virus is encrypted and the decryption routine is so short that it is impossible to extract a hex pattern longer than 14 bytes. The virus seems to activate in 1991, but the exact effects are as yet unknown.

```
Fish 6           E800 005B 81EB A90D B958 0D2E 8037; Offset 0
```

Flash - CER: This 688 byte virus is awaiting analysis.

```
Flash            005E 8BDE 81C3 0F00 B000 FAD5 0A88 07EB; Offset 007
```

Flip - MCER: The primary effect of this 2343 byte virus is to 'flip' the screen by rotating it through 90 degrees on the second day of the month between 10:00 and 10:59. The virus is encrypted and self-modifying. An infected file has the seconds field set to 62. No search pattern is possible for COM/EXE files. Search pattern will be found in the Master Boot Sector. (*VB* Sept 90).

```
Flip (boot)      33DB 33FF 8EC3 2629 0613 04CD 12B1 06D3; Offset 02E
```

Form - DR: A boot sector virus from Switzerland infecting hard disks and floppy disks. On the 24th day of every month the virus produces a small delay when keys are pressed.

```
Form             B106 D3E0 8EC0 33FF B9FF 00FC F3A5 06B8; Offset 074
```

Formiche - CR: A 6258 byte virus, which uses almost the same encryption method as Cascade.

```
Formiche         0F8D B74C 01BC D217 4631 3431 244C 75F8; Offset 013
```

Frog's Alley - CR: A 1500 byte virus, which infects program when the DIR command is issued, which makes it highly infectious. The virus activates on the 5th day of any month, overwriting the FAT and root directory.

```
Frog's Alley     0105 0001 26A3 1500 268C 1E13 0026 C706; Offset 029
```

Fu Manchu - CER: The virus attaches itself to the beginning of a COM file or to the end of an EXE file. Infective length is 2086 bytes (COM) and 2080 (EXE). It is a rewritten version of the Jerusalem virus, but the marker is 'rEMHOr' and the preceding 'sU' is 'sAX' (Sax Rohmer, creator of Fu Manchu). After installing itself as memory-resident, it will infect any COM or EXE file, except COMMAND.COM. EXE files are infected only once, unlike the original Jerusalem. One in sixteen times on infection a timer is installed, which will trigger a display 'The world will hear from me again' after a random number of half-hours (max. 7.5 hours). The machine then reboots. The same message is also displayed on pressing Ctrl-Alt-Del, but the virus does not survive the reboot. If the date is after 1st August 1989, the virus monitors the keyboard buffer and adds derogatory comments to the names of politicians (Thatcher, Reagan, Botha and Waldheim), overstrikes two four-letter words, and displays 'virus 3/10/88 - latest in the new fun line!' if 'Fu Manchu' is typed. All messages are encrypted. (*VB* July 89)

```
Fu Manchu        FCB4 E1CD 2180 FCE1 7316 80FC 0472 11B4; Offset 1EE, 2086 bytes COM, 2080 bytes EXE
```

F-word, USSR-417 - CR: A 417 byte virus, probably of Russian origin. The only text inside the virus is the message 'Fuck You'.

```
F-word           C3B4 3FCD 2129 C858 75DD FFE0 B440 EBF3
```

Gergana - CN: A simple 192 byte virus, which does nothing but replicate.

```
Gergana          FFE0 5E81 C600 01BF 0001 B9B6 00F3 A4B8; Offset 091
```

GhostBalls - CN: A strain of Vienna virus. Seconds field changed to 62, as in Vienna. Infective length is 2351 bytes and the virus attaches itself to the end of the file. When run, it will infect other COM files and try to place a modified copy of the Italian virus into the boot sector of drive A. This copy of the Italian runs on 286 machines but is non-infective. Virus contains text 'GhostBalls, Product of Iceland'.

```
GhostBalls       AE75 EDE2 FA5E 0789 BC16 008B FE81 C71F; Offset 051
```

Goblin - CER: A 1951 byte mutation of the Murphy virus. Detected by the HIV pattern.

GP1 - CER: This is a Dutch, Novell Netware-oriented mutation of the Jerusalem virus. (*VB* June 91)

```
GP1             B4F7 CD21 80FC F773 1380 FC03 072E 8E16; Offset 055H (COM) 025 (EXE)
```

Gremlin - CER: A 1146 byte 'Diamond' mutation detected by the same pattern.

Grither - CN: A 774 byte mutation of Vienna, which is detected by the Vienna (2) pattern. When it activates, it overwrites part of the hard disk, including the beginning of drive C:.

Guppy - CR: A very simple 152 byte virus. It does nothing but replicate, but many programs, including COMMAND.COM will fail to execute if infected.

```
Guppy           521E B802 3DCD 2193 E800 005E 0E1F B43F; Offset 045
```

Hallochen - CER: A virus which reputedly originated in West Germany. It contains two text strings (o in Hallochen is character code 148 decimal):

```
Hallochen !!!!!!, Here I'm..
Acrivate Level 1..
```

The virus will not infect 'old' files. If the value of the month or year fields in the time stamp is different from the current date, the file will not be infected. The virus will only infect files longer than 5000 bytes, increasing their length by 2011 bytes.

```
Hallochen       EB8C C903 D98E D3BC DB08 53BB 2E00 53CB; Offset 01E, 2011 bytes
```

Hero - CER: A primitive 506 byte virus, which will not replicate beyond the first generation, as a programming error causes it to corrupt all programs it infects.

```
Hero            C0CF 80FC 4B74 2080 FC25 7516 3C80 7212; Offset 0AD
```

HIV - CER: This virus is based on Murphy and contains a text message claiming it was written by 'Cracker Jack' in Italy.

```
HIV             2BC3 1BD1 7204 2906 0600 8BF7 33FF 0E1F; Offset 5C6
```

Horse, Hacker, Black horse - CER: A family of viruses probably from Bulgaria. Currently 8 different mutations are known, which can be divided into two groups, with a different pattern required for each group. Awaiting analysis. The first group contains Horse-1 (1154), Horse-2 (1158), Horse-2B (1160) and Horse-7 (1152)

```
Horse (1)       00A3 0001 8B46 02A3 0201 B800 018C CAEB; Offset variable
```

Horse group - CER: The second group of Horse viruses contains Horse-3 (1610), Horse-4 (1776), Horse-5 (1576) and Horse-6 (1594).

```
Horse (2)       570E 07B9 0800 F3A4 B02E AAB9 0300 F3A4; Offset variable
```

Hybrid - CN: A 1306 byte encrypted mutation of the Vienna virus which marks infected files by setting the seconds field of the time stamp to 62. On any Friday the 13th after 1991 the virus will format the hard disk. It may also overwrite files and cause reboots.

```
Hybrid          81EE 7502 8BFE B9DE 01AC 34DE AA49 75F9; Offset 007
```

Hymn - CER: A Russian, 1865 byte virus related to the 'Eddie' (Dark Avenger) virus, and the 'Murphy' viruses.

```
Hymn            FF64 F500 07E8 0000 5E83 EE4C FC2E 81BC
```

Icelandic, Saratoga - ER: The virus attaches itself at the end of an EXE file and after becoming memory-resident, it will infect only one in ten (one in two for the Icelandic (2) mutation) programs executed. When a program is infected, the disk is examined and if it has more than 20 MBytes, one cluster is marked as bad in the first copy of the FAT. There is a mutation which does not flag clusters. Version (1) will not infect the system unless INT 13H segment is 0700H or F000H, thus avoiding detection by anti-virus programs which hook into this interrupt. Version (3) does not flag clusters and bypasses all interrupt-checking programs.

```
Icelandic (1)   2EC6 0687 020A 9050 5351 5256 1E8B DA43; Offset 0C6, 656 bytes
Icelandic (2)   2EC6 0679 0202 9050 5351 5256 1E8B DA43; Offset 0B8, 642 bytes
Icelandic (3)   2EC6 066F 020A 9050 5351 5256 1E8B DA43; Offset 106, 632 bytes
```

INT13 - CR: Overwriting, stealth virus which subverts DOS and BIOS. The virus is 512 bytes long. Only selected COM files are infected during FCB find next function call. (*VB* Mar 91).

```
INT13           E200 50BF 4C00 5733 ED8E DDC4 1DBF 7402; Offset 0
```

Internal - EN: Infective length is 1381 bytes. Virus contains the string 'INTERNAL ERROR 02CH. PLEASE CONTACT YOUR HARDWARE MANUFACTURER IMMEDIATELY ! DO NOT FORGET TO REPORT THE ERROR CODE !'

```
Internal        1E06 8CC8 8ED8 B840 008E C0FC E858 0480; Offset 0B1
```

Iraqui Warrior - CN: A 777 byte mutation of Vienna, where numerous NOP instructions have been added to avoid detection by current scanners.

```
Iraqui Warrior BF00 0190 B903 00F3 A490 8BF2 B430 90CD; Offset 00E
```

Italian, Pingpong, Turin, Bouncing Ball, Vera Cruz - DR: The virus consists of a boot sector and one cluster marked as bad in the first copy of the FAT. The first sector in the marked cluster contains the rest of the virus while the second contains the original boot sector. It infects all disks which have at least two sectors per cluster and occupies 2K of RAM. It displays a single character 'bouncing ball' if there is a disk access during a one-second interval in any multiple of 30 minutes on the system clock. The original version will hang when run on an 80286 or 80386 machine, but a new version has been reported which runs normally. If a warm boot (Ctrl-Alt-Del) is performed after the machine hangs, an uninfected disk will still become infected. (*VB* Nov 89)

```
Italian-Gen      B106 D3E0 2DC0 078E C0BE 007C 8BFE B900; Offset 030
Italian          32E4 CD1A F6C6 7F75 0AF6 C2F0 7505 52E8; Offset 0F0
```

Itavir - EN: When the virus activates, it will write random data to all I/O ports causing unpredictable behaviour such as screen flicker, hissing from the loudspeaker etc. Infective length is 3880 bytes.

```
Itavir           83C4 025A 595B 5850 5351 52CD 2672 0D83; Offset 198
```

Jeff - CN: Just like the Klaeren virus, Jeff can not successfully infect files longer than 4096 bytes. The virus is 812 bytes long, (not 814 as reported earlier). When it activates it may overwrite sectors on the hard disk.

```
Jeff             B89B FF8E C0B9 3F00 33D2 32E4 8BD9 268A; Offset 034
```

Jerusalem, PLO, Friday the 13th, Israeli - CER: The virus attaches itself to the beginning of a COM file or at the end of an EXE file. When an infected file is executed, the virus becomes memory-resident and will infect any COM or EXE program run, except COMMAND.COM. COM files are infected only once, while EXE files are re-infected every time that they are run. Infective length is 1813 bytes (COM) and 1808 bytes (EXE). The virus finds the end of EXE files from the information in the file header, and if this is less than the actual file length, the virus will overwrite part of the file. After the system has been infected for 30 minutes, row 5 column 5 to row 16 column 16 on the screen are scrolled up two lines, creating a 'black window'. The system then slows down, due to a time-wasting loop installed on each timer interrupt. If the system is infected when the date is set to the 13th of any month which is also a Friday, every program run will be deleted. (*VB* July 89) Jerusalem mutations matching the following two search patterns:

```
Jerusalem        03F7 2E8B 8D11 00CD 218C C805 1000 8ED0; Offset 0AC, 1813 bytes COM, 1808 bytes EXE
Jerusalem-USA    FCB4 E0CD 2180 FCE0 7316 80FC 0372 11B4; Offset 095
```

Minor jerusalem variants matching the Jerusalem search pattern: Anarkia: Virus signature is changed from 'sURIV' to 'ANARKIA'. **Anarkia-B**: Minor mutation of Anarkia. **Carfield**, CER: 1508 bytes variant of Jerusalem. Detected by Jerusalem (1) pattern. **Frere Jacques,** CER: There are two mutations known as A and B which play the Frere Jacques tune on Fridays. Detected by Jerusalem (1) pattern. **Groen Links**, GrLkDos: An 1888 byte mutation from The Netherlands. Every 30 minutes it plays the tune 'Stem op Groen Link' or 'Vote Green Left'. **Jerusalem-1600/1605**: A shortened mutation awaiting analysis. **Mendoza**: A mutation of Anarkia. **A-204**, **Payday**, **Puerto**, **Spanish** and **Jerusalem-G**: minor mutations of Jerusalem.

Jo-Jo - CR: This is a non-encrypted version of Cascade with the encryption code patched out and a few other changes made.

```
Jo-Jo            B800 F08E C0BF 08E0 813D 434F 751B 817D; Offset 0D2
```

Jocker: An overwriting virus from Poland, written in some high-level language, probably Pascal.

```
Jocker           89E5 81EC 0001 BF00 000E 57BF 401B 1E57; Offset 00B
```

Joker-01 - CR: A huge, 29233 byte virus of Polish origin.

```
Joker-01         8CC2 4A8E C28C DA4A 8EDA 5A90 26A1 0300
```

Joshi - MR: This virus from India displays the message Type "Happy Birthday Joshi" on 5th January of every year. Unless the user enters the text verbatim, the computer will hang. The virus traps disk reads and any program trying to discover it while the virus is active in memory, will not locate it. Survives warm boot. (*VB* Dec 90).

```
Joshi            03F0 03F8 B979 012B C8FC F3A6 7510 8CC0; Offset 041
```

July 13th - ER: This encrypted virus activates on 13th July, but its exact effects have not yet been determined. It is 1201 bytes long.

```
July 13th        2EA0 1200 3490 BE12 00B9 B104 2E30 0446; Offset variable
```

Justice - CR: A 1242 byte virus which has not been fully analysed. Many computers 'hang' after running an infected program.

```
Justice          509F 83C4 089E 9C83 EC06 58CF 3CFF 7504; Offset 1F8
```

Kamikaze - EN: This overwriting virus from Bulgaria is written in Turbo Pascal, and is fairly large, 4031 bytes. Like other similar viruses it is not a serious threat. The search pattern for this virus has been withdrawn due to its causing false-positives.

Kemerovo - CN: A Russian, 257 byte virus. Some infected programs fail to execute properly, but no other effects are known.

```
Kemerovo         0400 89C7 B904 00A4 E2FD 89D7 29D3 81EB
```

Kennedy - CN: A simple virus, probably from Sweden. When an infected file is run, it will infect a single COM file in the current directory, expanding it by 333 bytes at the end. The virus activates on 6th June, 18th November and 22nd November and displays:

```
                 Kennedy er dod - lange leve "The Dead Kennedys"
Kennedy          E817 0072 04B4 4FEB F38B C505 0301 FFE0; Offset 035
```

Keypress, Turku, Twins - CER: This virus was discovered at the same time in Finland, USSR and Bulgaria, which makes its origin somewhat uncertain. It will infect COM and EXE files, but the length of the virus code is different, 1232 and 1472 bytes, respectively. After being resident for some time the virus will interfere with the keyboard, causing keys to 'repeat'.

```
Keypress        7405 C707 0100 F9F5 1FC3 F606 1801 0174
```

Keypress-1228 - CER: Slightly variant of 1232 byte version. Discovered in Kansas, USA. Detected by the 'Keypress' pattern.

Kiev - CR: Infected files grow by 483 bytes, but this increase is not visible when a DIR command is issued.

```
Kiev            8BD3 81C2 FBFF 8BDF B440 CD21 5B72 0053; Offset 170
```

Klaeren - CER: This 974 byte encrypted virus contains a serious error which prevents it from infecting files larger than 4096 bytes. Virus contains the text 'Klaeren Ha<ess-tzet>, Ha<ess-tzet>' (Klaeren: the name of a professor in the school where it was written.)

```
Klaeren          5351 E800 005B 81EB AF03 B9A5 0380 37; Offset 3B0
```

Korea, NJH - DR: A simple boot sector virus with no side-effects. It may cause damage to data, as the original boot sector is always written to sector 11. There are two versions, probably due to two different assemblers being used.

```
Korea           C08E D88E D0BC F0FF FBBB 1304 8B07 4848; Offset 009
```

Kylie - CER: A 2272 byte mutation of the Jerusalem virus, which plays a tune when it activates.

```
Kylie           E2FE C3E4 6124 FCE6 61C3 5357 4343 8B3E; Offset 385
```

Lehigh - CR: The virus only infects COMMAND.COM. It is 555 bytes long and becomes memory-resident when the infected copy is run. If a disk is accessed which contains an uninfected COMMAND.COM, the copy is infected. A count of infection generation is kept inside the virus, and when it reaches 4 (or 10 in a mutated version), the current disk is trashed each time a disk is infected, provided that (a) the current disk is in either the A drive or B drive, (b) the disk just infected is in either the A drive or B drive and (c) the disk just infected is not the current one. The trashing is done by overwriting the first 32 sectors following the boot sector. Infection changes the date and time of COMMAND.COM.

```
Lehigh          8B54 FC8B 44FE 8ED8 B844 25CD 2106 1F33; Offset 1EF
```

Leprosy - CN: A 666 byte encrypted overwriting virus, similar to Leprosy-B but using a different encryption method.

```
Leprosy         558B EC56 8B76 04EB 0480 2C0A 4680 3C00; Offset 25C
```

Leprosy-B - CER: A 666 byte overwriting virus, which is easily detected, as infected programs do not run normally, but instead display a message announcing the virus.

```
Leprosy-B       8A27 3226 0601 8827 4381 FBCB 037E F1C3; Offset 021
```

Leszop - DR: Virus awaiting disassembly.

```
Leszop          1FC7 060C 7C62 008C 060E 7CFB FF2E 0C7C; Offset 039
```

Liberty, Magic, Mystic - DCER: A virus from Indonesia with an infective length of 2857 bytes after padding to a 16 byte boundary. Infects diskette boot sectors when disk full, using track 40 which may cause data loss. Booting from infected disk causes text message 'MAGIC' to appear at intervals on screen, printer and modem. On 10th boot, disinfects disk of itself but flags boot sector; if this flag is detected later, Track 0 is modified to be unreadable by DOS. 2867 and 2873 byte mutations are also known.

```
Liberty         0174 031F 595B 5053 5152 1E06 1E0E 1FE8; Offset 080
```

Little Pieces - ER: A 1374 byte virus, which has not been fully analysed. It will occasionally clear the screen and display the message: 'One of these days I'm going to cut you into little pieces'.

```
Little Pieces   9DCA 0200 33DB 8EDB C747 4C56 018C 4F4E; Offset 2c0
```

Lozinsky - CR: A Russian, 1023 byte virus, which uses a simple encryption algorithm.

```
Lozinsky        FCBF 2000 03FE B9D0 032E 3005 47E2 FAB8; Offset 013
```

LoveChild - CN: Infective length is 488 bytes. Contains strings 'v2 (c) Flu Systems (R)' and 'LoveChild in reward for software sealing.' [sic]. The virus trojanises certain program files which, when trigerred, overwrite sectors 1-16, heads 0-3 on every track of the first hard disk with garbage. (*VB* Feb 91).

```
LoveChild       33C0 8EC0 E800 005E 8BEE BFE0 01FC 2681
LoveChild Trojn B901 00BA 8003 8BD9 B810 03CD 13FE CE79; Offset 0
```

Lucifer - CER: A 1086 byte mutation of the Diamond virus. Detected by the Diamond pattern.

Macho - CEN: Swaps every string 'MicroSoft' with 'MachoSoft' on the hard disk. Searches 20 sectors at a time, storing the last sector searched in IBMNETIO.SYS which is marked hidden and system. After searching the last sector it starts again. This will only happen after 1st January 1985 and if the environment variable VIRUS is not set to OFF. Infective length is 3550 to 3560 bytes. Random directory search for uninfected files. Infects COMMAND.COM. This virus is closely related to Syslock. (*VB* May 91)

```
Macho           5051 56BE 5900 B926 0890 D1E9 8AE1 8AC1; Offset ?
```

Magnitogorsk, 2560 - CER: This virus has not been fully analysed yet, but it contains a greeting to a Mr. Lozinsky, who is the author of a Russian anti-virus program.

```
Magnitogorsk     2E8B 851F 003D FFFF 7413 BE3E 0003 F7B9; Offset 021
```

Mardi Bros - DR: The major effect of the virus is to change the volume label to 'Mardi Bros'. It is believed to be of French origin.

```
Mardi Bros       E08E C0BE 007C 31FF B900 14FC F3A4 06B8; Offset 131
```

MG - CR: A simple, 500 byte Bulgarian virus. A minor variant called MG-1A is detected by the same pattern.

```
MG               AA1F 1E07 585E 1EBB 0001 53CB 3D04 4B74; Offset 086
```

MG-3 - CR: A 500 byte Bulgarian virus, reported to be written by the same author as the MG virus.

```
MG-3             C43E 0600 B0EA 49F2 AE26 C43D 83EF DFEA
```

MG-4 - CR: A 500 byte virus from Bulgaria, which is closely related to the MG-3 virus, and is detected by the same pattern..

MGTU - CN: A simple, 273 byte Russian virus.

```
MGTU             03F8 BE00 018B 0589 048B 4502 8944 02B8; Offset 0F8
```

Micro-128 - CR: This virus from Bulgaria is the smallest memory-resident virus known. It occupies part of the interrupt table and does nothing but replicate.

```
Micro-128        7501 A5A4 31C0 8EC0 BF03 03B1 7DF3 A4AF; Offset 00A
```

Microbes - DR: An Indian virus the effects of which are not fully known, except that booting from an infected disk has been reported to cause some computers to 'hang'.

```
Microbes         042D 0400 A313 04B1 06D3 E08E C006 C706; Offset 014
```

Migram-1 - ER: A 1219 byte mutation of the Murphy virus. Detected by the Murphy(2) pattern.

Migram-2 - ER: A 1221 byte mutation of the Murphy virus. Detected by the HIV pattern.

Minimal-30 - CN: Currently the smallest known virus, only 30 bytes long. When an infected program is run, it will overwrite the first file in the current directory.

```
Minimal-30       3DBA 9E00 CD21 93B4 408B D68B CECD 21C3; Offset 00A
```

Minimal-45 - CN: This Bulgarian overwriting virus is the smallest one known - only 45 bytes long. When run, it will overwrite all COM files in the current directory with itself.

```
Minimal-45       0001 B92D 00B4 40CD 21B4 3ECD 21B4 4FEB; Offset 015
```

MIR - CER: A 1745 byte mutation of the Eddie (Dark Avenger). The first generation sample contains the text 'M.I.R. *-*-*-* Sign of the time!', but it is corrupted in later generations. Detected by the 'Dark Avenger' pattern.

Mirror - ER: The virus is 924 bytes long, but infected programs may grow by a maximum of 940 bytes. When the virus activates it will reverse the contents of the screen, displaying a mirror image of what was there before.

```
Mirror           8A07 2688 0743 E2F8 B821 2506 1FBA DC00; Offset 04D
```

Mistake, Typoboot - DR: Exchanges letters for phonetically similar ones (for example 'C' and 'K') while they are being output to the printer. Reportedly written in Israel. A mutation of the Italian virus. The infected boot sector is almost identical to the Italian virus.

```
Mistake          32E4 CD1A 80FE 0376 0A90 9090 9090 52E8; Offset 0F0
```

MIX1 - ER: The virus infects only EXE files, attaching itself to the end. When an infected program is run, the virus will copy itself to the top of the free memory. Some programs may overwrite this area, causing the machine to crash. The virus traps printer and asynch interrupts and corrupts traffic by substituting characters. 50 minutes after infection, the virus alters the Num Lock and Caps Lock keyboard settings. 60 minutes after infection, a display similar to the Italian virus (bouncing ball display) will be produced. The virus will infect every tenth program run. Infected files always end in 'MIX1' and the infective length of MIX1 is 1618 to 1633 bytes and MIX1-2 1636 to 1651 bytes. (*VB* Dec 89)

```
MIX1             B800 008E C026 803E 3C03 7775 095F 5E59; Offset 02E
MIX1-2           B800 008E C0BE 7103 268B 3E84 0083 C70A; Offset 02A
```

MIX2 - CER: This is a 2280 byte Israeli virus based on MIX1 but improved with the addition of encryption and COM file infection.

```
MIX2             EE8C C803 C650 B826 0050 CB55 508C C0E8; Offset 01B
```

MLTI - CR: This 830 byte Russian virus contains the following text, which clearly refers to the 'Eddie' (Dark Avenger) virus. 'Eddie die somewhere in time! This programm was written in the city of Prostokwashino (C) 1990 RED DIAVOLYATA Hello! MLTI!'

```
MLTI             5B73 05B8 0001 50C3 83FC E072 F62E C747
```

Monxla, Time - CN: A 939 byte mutation of the Vienna virus, which activates on the 13th day of any month and then damages programs, instead of just infecting them.

```
Monxla           8B07 5B8E C0BF 0000 5E56 83C6 1AAC B900
```

Monxla-B - CN: This 535 byte virus is probably an older version of the Monxla virus. It retains code from the Vienna virus which deletes programs instead of infecting them 1 every 8 times.

```
Monxla-B        8994 1600 B42C CD21 80E6 0775 10B4 40B9; Offset 128
```

Murphy - CER: Two versions exist. One produces a click from the loudspeaker when any DOS functions are called, while the other may produce a bouncing ball effect when the user enters ROM BASIC. The virus will only activate between 10:00 and 11:00 a.m.

```
Murphy 1        1EE8 0000 B859 4BCD 2172 03E9 2801 5E56; Offset variable
Murphy 2        1EE8 0000 B84D 4BCD 2172 03E9 2601 5E56; Offset variable
```

Murphy-3 - CER: A 1284 byte mutation of Murphy detected by the 'HIV' pattern.

Murphy-4 - CER: A 1480 byte mutation of Murphy detected by the 'Murphy 2' pattern.

Music Bug - DR: Awaiting disassembly.

```
Music Bug       08FC F3A5 06B8 0002 50CB 5053 5152 2EA3
```

Mutant - CN: Three mutations of this virus are known, of which two, 123 and 127 bytes long, are only able to infect small files correctly. This is 'corrected' in the third mutation, also 127 bytes long. The viruses have no interesting side-effects.

```
Mutant          C98B D1B8 0042 CD21 5972 065A 52B4 40CD; Offset variable
```

New Zealand, Stoned, Marijuana - MR: The virus consists of a boot sector only. It infects all disks and occupies 2K of RAM. On floppy disks, logical sector 0 is infected, while on the hard disks the first physical sector 0 (Master boot sector) is infected. The original boot sector is stored in track 0 head 1 sector 3 on a floppy disk and track 0 head 0 sector 2 on a hard disk. The boot sector contains two character strings: 'Your PC is now Stoned!' and 'LEGALISE MARIJUANA' but only the former one is displayed, once in eight times, and only if booted from floppy disk. The version (2) stores the original boot sector at track 0 head 0 sector 7 on a hard disk. The second string is not transferred when a hard disk is infected. A mutation displays the message 'Your PC is now Sanded'. A mutation has been reported in Australia which also displays 'LEGALISE MARIJUANA'. (*VB* May 90)

```
New Zealand (1) 0400 B801 020E 07BB 0002 B901 0033 D29C; Offset 043
New Zealand (2) 0400 B801 020E 07BB 0002 33C9 8BD1 419C; Offset 041
```

Nina - CR: Yet another small virus from Bulgaria. This one is 256 bytes long.

```
Nina            03F7 B900 01F3 A458 1EBD 0001 55CB 5858; Offset 069
```

No Bock, 440 - CN: When this 440 byte virus activates it displays 'No Bock today error. System Halted' and stops the system.

```
No Bock         A48B FDC3 B104 D3E0 0AC6 FEC1 D3E0 0AC2
```

Nomenklatura - CER: Infective length is 1024 bytes, and only files longer than 1024 bytes are infected. The virus infects on executing a program or opening a file, which means that a virus scanning program will infect all files on the system if the virus is resident in memory. The virus scrambles the FAT on a random basis. (*VB* Dec 90).

```
Nomenklatura    B8AA 4BCD 2173 785E 5606 33C0 8ED8 C41E; Offset 2DD
```

NTKC, C-23693 - CN: A 23693 byte mutation of Vienna, detected by the 'Vienna (4)' pattern.

Number One - CN: An old, simple, overwriting, Pascal virus, originally published in the 'Computer Viruses - A High Tech Disease' book by Burger. Infective length depends on the compiler used, but 11980 and 12032 byte examples have been found in the wild.

```
Number1         B800 0050 BFCC 031E B142 E8E8 FEB8 015C; Offset 064
```

Number of the Beast, 666, V512 - CR: An advanced virus from Bulgaria, only 512 bytes long. The length of the file does not appear to increase since the virus overwrites the first 512 bytes of the programs it infects with itself, storing the original 512 bytes in the unused space of a disk cluster, after the logical end of file. (*VB* May 90, June 90)

```
Number of Beast 5A52 0E07 0E1F 1EB0 5050 B43F CBCD 2172; Offset 0A3
Number of Bea 1 B800 3DCD 2193 5A52 0E1F 1E07 B102 B43F; Offset variable
Number of Bea E 1607 8BD6 B102 B43F CD21 8AD1 86CD BFFE
Number of Bea F 5A52 0E1F 1E07 06B0 5050 B43F CBCD 2172
```

Ohio, Hacker - DR: Boot sector virus, which is an older version of Den Zuk and written by the same author.

```
Ohio            FAFA 8CC8 8ED8 8ED0 BC00 F0FB E845 0073; Offset 02B
```

Old Yankee - EN: This is the first of the viruses which play 'Yankee Doodle Dandy'. It only infects EXE files, increasing their length by 1961 bytes. When an infected program is run, it will infect a new file and then play the melody. (*VB* June 90)

```
Old Yankee      03F3 8CC0 8904 0E07 53B8 002F CD21 8BCB; Offset 009
```

Ontario - CER: A 512 byte encrypted virus. It uses self-modifying encryption, and a full 16-byte search pattern cannot be extracted. The asterisks in the string indicate a byte which may change from one infected file to another.

```
Ontario         8A84 E801 B9E8 01F6 **2E 3004 46E2 F8C3; Offset 1F0
```

Oropax, Music virus - CR: The length of infected files increases between 2756 & 2806 bytes and their length becomes divisible by 51. 5 minutes after infection, the virus plays three different tunes at 7-minute intervals. Does not infect COMMAND.COM.

```
Oropax          06B8 E033 CD21 3CFF 7423 8CCE 8EC6 8B36
```

Paris, TCC - CEN: The virus will infect all EXE files in the current directory, when an infected file is run. Length is 4904 bytes.

```
Paris           8CD8 03C3 8ED8 8EC0 8D3E 0301 B000 AAEB; Offset 7EE
```

Parity - CN: A Bulgarian 441 byte virus which may emulate a memory failure when an infected program is run, displaying the message 'PARITY CHECK 2' and halting the computer.

```
Parity          40B9 B901 BA00 0103 D7CD 21B8 0157 8B8D
```

PcVrsDs - CER: A destructive encrypted virus which deletes every file opened and infects every file executed. It does not infect COMMAND.COM. A routine in the virus causes occasional typing errors by incrementing the ASCII value of the character typed by 1. On Monday 23rd of every month, except in 1990, it will format side 0 of the first 32 tracks on the first fixed disk. (*VB* Apr 91).

```
PcVrsDs         33DB BE1C 00B9 4F07 2E8A 9708 002E 0010; Offset 0
```

Pentagon - DR: The virus consists of a boot sector and two files. The sample obtained does not work, but it contains the code which would survive a warm boot (Ctrl-Alt-Del). It could only infect 360K floppy disks, and will look for and remove Brain from any disk it infects. It occupies 5K of RAM.

```
Pentagon        8CC8 8ED0 BC00 F08E D8FB BD44 7C81 7606; Offset 037
```

Perfume - CR: The infected program will sometimes ask the user for input and not run unless the answer is 4711 (name of a perfume). In some cases the question is 'Bitte gebe den G-Virus Code ein', but in others the message has been erased. The virus will look for COMMAND.COM and infect it. Infective length is 765 bytes.

```
Perfume         FCBF 0000 F3A4 81EC 0004 06BF BA00 57CB; Offset 0AA
```

Perfume-731 - CR: A slight mutation of the Perfume virus, only 731 bytes long. This may well be an earlier mutation.

```
Perfume-731     FCBF 0000 F3A4 81EC 0004 06BF BC00 57CB; Offset 1AC
```

Pest - CER: A 1910 byte mutation of the Murphy virus. Detected by the HIV pattern.

Phantom - CR: A 2201 bytes long virus, which has not yet been fully analysed. The virus contains an encrypted text message stating it was written in Hungary.

```
Phantom         CF8B FA1E 07B0 00B9 5000 FCF2 AE83 EF04; Offset 1A5
```

Phenome - CER: A minor mutation of the Jerusalem virus 1808 (1813) bytes long, just like the original. Detected by the Jerusalem-USA pattern.

Phoenix, P1 - CR: This Bulgarian virus is 1701 bytes long, but a mutation, 1704 bytes long, has also been reported. Despite the identical lengths, they are not related to the Cascade viruses. These viruses use an advanced encryption method, so that no search pattern is possible.

Piter, Polish 529 - CR: A Russian, 529 byte virus.

```
Piter           8E1E 2C00 33F6 AC0A 0475 FB83 C603 8BD6; Offset 092
```

Pixel - CN: The Pixel viruses are practically identical to the Amstrad virus, although they are shorter: 345 and 299 bytes. No side-effects are noticeable until the 5th generation is reached, at which stage there is a 50 % chance that the following message will appear when an infected program is executed:

```
Program sick error: Call doctor or buy PIXEL
for cure description
```

Several new mutations of the Pixel/Amstrad virus have been discovered, most of which are very similar to previous mutations, and are detectable by the 'Pixel' pattern. (*VB* June 90)

```
Pixel (1)       0E1F 2501 0074 4CBA D801 B409 CD21 CD20; Offset 0C8, 354 bytes
Pixel (2)       BA9E 00B8 023D CD21 8BD8 061F BA2B 01B9; Offset 033, 299 bytes
Pixel (3)       0001 0001 2E8C 1E02 018B C32E FF2E 0001
```

Pixel-257,**275**,**295**,**283** - CN: detected by the 'Pixel-277' pattern.

Pixel-892 - CN: detected by the 'Pixel-345' pattern.

Pixel-779,**837**,**850**,**854** - CN: detected by the 'Amstrad' pattern.

Pixel-936 - CN: A 936 byte mutation of the Pixel/Amstrad virus.

```
Pixel-936       C706 0001 0001 2E8C 1E02 012E FF2E 0001; Offset 198
```

Plague - CR: A simple 591 byte overwriting virus, based on the Leprosy virus.

```
Plague          8A27 3226 0601 8827 4381 FB83 037E F1EB; Offset 021
```

Plastique 521 - C?: Virus awaiting disassembly.

```
Plastique 521    0681 002E 8C06 8500 2E8C 0689 008C C005; Offset 0
```

Polimer - CN: A 512 byte Hungarian virus, which only displays the following message when an infected program is executed: “A le’ jobb kazetta a Polimer kazetta ! Vegye ezt !”

```
Polimer          8CD8 0500 108E D8B4 40CD 218C D82D 0010; Offset 0F5
```

Polish 217 - CR: A simple 217 byte virus from Poland, which does nothing but replicate. Polish 217-A is a minor mutation, probably changed to bypass some scanner.

```
Polish 217       D201 BF00 01B9 0300 F3A4 5EB4 4EBA C901
```

Pretoria, June 16th - CN: Overwrites the first 879 bytes of infected files with a copy of itself and stores the original 879 bytes at the end of the file. When an infected program is executed, the virus searches the entire current drive for COM files to infect. On 16th June the execution of an infected file will cause all entries in the root directory to be changed to ‘ZAPPED’. The virus is encrypted.

```
Pretoria         AC34 A5AA 4B75 F9C3 A11F 0150 A11D 01A3
```

PrintScreen - DR: Occasionally performs a Print Screen (PrtSc) operation.

```
Printscreen      FA33 C08E D0BC 00F0 1E16 1FA1 1304 2D02; Offset 023
```

Protecto - C?: Virus awaiting disassembly.

```
Protecto         8BD6 83C2 4AB8 003D CD21 7303 EB39 908B; Offset 007
```

Proud - CR: This 1302 byte virus is a member of a Bulgarian family of 4 viruses, which also includes 1226, Evil and Phoenix. As they all use the same encryption method, no search pattern is possible. (*VB* Dec 90).

Prudents - EN: Infective length is 1205 bytes and the virus will destroy the last 32 bytes of any infected file. Activates during the first four days of May of every year, turning every write operation into a verify operation, which results in the loss of data.

```
Prudents         0E07 BE4F 04B9 2300 5651 E87E 0359 5EE8; Offset 055
```

PSQR: A mutation of Jerusalem with the signature changed to ‘PSQR’. The infective length is 1715 (COM) and 1720 bytes (EXE).

```
PSQR             FCB8 0FFF CD21 3D01 0174 3B06 B8F1 35CD; Offset 071
```

Rat - ER: This Bulgarian virus infects EXE files in a very unusual way by locating itself in the unused area between the header and the start of the program, preventing the increase in the file size. Most EXE files are immune to the infection by this virus.

```
Rat              FCB8 2B35 CD21 8CDD 0E1F 012E 6A0A BE10; Offset 0
```

Raubkopi, Raub - CR: This virus adds 2219 bytes in front of COM files, but much of that is occupied by a text message in German, directed against pirated software. The virus contains code to format the boot sector of the hard disk, but that code contains an error.

```
Raubkopi         0500 013D 0002 7204 25FF 0142 B104 D3E8; Offset 537
```

Russian Mirror - CR: This vicious virus from Russia trashes disks. Infective length is 482 bytes.

```
Russian Mirror   E89D FF80 FC4B 7403 E9C4 002E FE0E 6400
```

Saddam - CR: This virus extends the file length by 917 to 924 bytes. Displays the following string (which is stored encrypted)

```
HEY SADAM
LEAVE QUEIT BEFORE I COME
```

after 8 requests for INT 21H. Resides in the area of memory not labelled as used, so large programs will overwrite it.

```
Saddam           BB00 0153 5052 1E1E B800 008E D8A1 1304; Offset 010
```

Scott’s Valley - CER: This virus is closely related to the Australian Slow virus, using an almost identical encryption method. It is somewhat longer, 2126 bytes.

```
Scott’s Valley   E800 005E 8BDE 9090 81C6 3200 B912 082E
```

Sentinel - CR: This virus is written in Turbo Pascal and is 4625 bytes long.

```
Sentinel         FCAD 2EA3 0001 AC2E A202 0189 EC5D B800; Offset variable
```

September 18th - CEN: This virus activates on September 18th, after 7:00 AM, overwriting the hard disk. Two mutations are known, 789 and 801 bytes long, but the virus adds 1-16 extra bytes to programs before infecting them. These viruses may be related to the StarDot virus.

```
September 18th   7502 32C0 3CFF 7502 B001 5051 CD26 83C4; Offset variable
```

Sex Revolution - MR: Two versions are known and they both contain the text ‘EXPORT OF THE SEX REVOLUTION’. The virus is a mutation of the New Zealand virus and is detected by the New Zealand (2) pattern.

Shake - CR: A primitive 476 byte virus which reinfects previously infected files. Infected programs sometimes reboot when executed. Occasionally, infected programs display the text ‘Shake well before use !’ when executed.

```
Shake            B803 42CD 213D 3412 7503 EB48 90B4 4ABB
```

Simulation - CN: A self-modifying encrypted Norwegian virus awaiting full disassembly. Infectious length is 1281 bytes. Contains the same text as the Datacrime virus, the text 'dancing with the devil in the pale moonlight' (a line spoken by Jack Nicholson the film 'Batman') and 'FRODO LIVES'. No search pattern is possible.

Skism - CER: A 1808/1813 byte minor mutation of Jerusalem. Detected by the Jerusalem-USA pattern.

Slow - CER: This encrypted virus is a 1716 byte long mutation of the Jerusalem virus. It originates from Australia and its side-effect is reported to be a slow-down of the infected PC. No other side-effects are known, as the virus is awaiting analysis.

```
Slow            E800 005E 8BDE 9090 81C6 1B00 B990 062E; Offset 0
```

Smack, Patricia - CER: A mutation of the HIV virus containing a message for Patricia Hoffman. Two mutations are known, 1835 and 1841 bytes, both probably written by 'Cracker Jack'. Both mutations are detected by the HIV pattern.

South African, Friday the 13th, Miami, Munich, Virus-B - CN: Infective length is 419 bytes, but some reports suggest mutations with an infective length between 415 and 544 bytes. Does not infect files with Read-Only flag set. Virus-B is a non-destructive mutation containing South African 2 pattern. COMMAND.COM is not infected. Every file run on a Friday 13th will be deleted.

```
S African 1     1E8B ECC7 4610 0001 E800 0058 2DD7 00B1; Offset 158
S African 2     1E8B ECC7 4610 0001 E800 0058 2D63 00B1; Offset 158
```

South African 408 - CN: A 408 byte version of the South African virus, partially rewritten to foil scanners, but with no new effects.

```
S African 408   1E8B ECC7 4610 0001 E800 0058 2D5A 0090 ; Offset 04F
```

South African 416 - CN: Another minor mutation. The following search pattern detects all known mutations of this virus.

```
S African 416   FF36 0301 FF36 0501 B43F B903 00BA 0301; Offset variable
```

Spanish Telecom - MCER: This encrypted virus contains a message by 'Grupo Holokausto' demanding 'lower telephone tariffs, more services'. It proclaims to be an 'Anti-CTNE' virus where CTNE is 'Compania Telefonica Nacional Espana'. A message in English states that the virus was programmed in Barcelona, Spain. (*VB* Jan 91).

```
Spanish Head 1  8B1D B200 83FB 0074 18BF 5500 B2; Offset 034H
Spanish Head 2  83ED 09BE 2001 03F5 FCB6; Offset 024H
Spanish Trojan  BB00 7C33 C0FA 8ED0 8BE3 FB8E D8A1 1304; Offset 001E in Boot Sectors
```

Sparse - CR: This virus is 3840 bytes long, but most of it contains zero byes. It has no interesting side-effects.

```
Sparse          FF0F CD21 50B4 3DB0 02CD 2189 C3B4 42B9; Offset 222
```

Staf - CN: A 2083 byte 'demonstration' virus, which seems to have no harmful effects. The virus contains the following text: Virus Demo Ver.: 1.1 - Handle with care! By STAF (Tel.: (819) 595-0787).

```
Staf            89D3 33F6 8038 0074 0343 EBF8 C600 245A; Offset 231
```

Striker 1 - CN: A 461 byte virus, which has not been analysed yet. It contains an error which causes incorrect infection of COM files shorter than 13 bytes.

```
Striker 1       5A8B 4606 39C2 7403 42EB E840 8946 06A0; Offset 0CA
```

Subliminal - CR: This 1496 byte virus is probably an earlier version of the Dyslexia virus. When active, the virus will attempt to flash the message 'LOVE, REMEMBER' on the screen for a fraction of a second, which is too short to be easily noticed.

```
Subliminal      AE26 3805 E0F9 8BD7 83C2 0306 1F2E C706; Offset 435
```

Sunday - CER: Variation of Jerusalem. Infective length is 1631 bytes (EXE) and 1636 (COM). Activates on Sunday and displays message 'Today is SunDay! Why do you work so hard? All work and no play make you a dull boy.'. There are unconfirmed reports of FAT damage on infected systems.

```
Sunday          FCB4 FFCD 2180 FCFF 7315 80FC 0472 10B4; Offset 095
```

Suomi - CN: A 1008 byte virus from Finland, which uses self-modifying encryption, like the 1260 virus. The virus seems to disinfect previously infected files under certain conditions, but COMMAND.COM seems to remain permanently infected. No harmful side-effects have been reported, but the virus is awaiting disassembly. No search pattern is possible.

Suriv 1.01, April 1st COM - CR: A precursor to Jerusalem infecting only COM files with the virus positioned at the beginning of the file. Infective length is 897 bytes. If the date is 1st April, the virus will display 'APRIL 1ST HA HA HA YOU HAVE A VIRUS' and the machine will lock. If the date is after 1st April 1988, the virus produces the message 'YOU HAVE A VIRUS !!!' but the machine will not lock. The virus is memory resident and will not infect COMMAND.COM. (*VB* Aug 89)

```
Suriv 1.01      0E1F B42A CD21 81F9 C407 721B 81FA 0104; Offset 304, 897 bytes
```

Suriv 2.01, April 1st EXE - ER: A precursor to Jerusalem infecting only EXE files with the virus positioned at the beginning of the file. Infective length is 1488 bytes. If the date is 1st April, the virus will display 'APRIL 1ST HA HA HA YOU HAVE A VIRUS'. If the year is 1980 (DOS default) or the day is Wednesday after 1st April 1988, the machine locks one hour after infection. (*VB* Aug 89)

```
Suriv 2.01      81F9 C407 7228 81FA 0104 7222 3C03 751E; Offset 05E, 1488 bytes
```

Suriv 3.00, Israeli - CER: Early version of Jerusalem infecting COM and EXE files and displaying the side-effects 30 seconds after infection. Infective length is 1813 bytes (COM) and 1808 bytes (EXE). Program delete does not work. (*VB* Aug 89)

```
Suriv 3.00      03F7 2E8B 8D15 00CD 218C C805 1000 8ED0; Offset 0B0, 1813 COM, 1808 EXE
```

SVC - CER: A Russian, 1689 byte virus, containing the following message '(c) 1990 by SVC,Vers. 4.0'. The virus attempts to avoid detection by the use of 'stealth' methods, so any increase in file length is not visible while the virus is active in memory.

```
SVC             7416 80FC 1174 0E80 FC12 7409 9D2E FF2E; Offset 142
```

SVC 3.1 - CER: This 1064 byte virus is probably an older version of the SVC virus.

```
SVC 3.1         C39D BA90 19CF 5A1F EBBD 33C0 8EC0 26C4; Offset 13D
```

Sverdlov - CER: A Russian, 1962 byte virus, using a simple XOR-encryption.

```
Sverdlov        2D00 03FE 2E30 0547 E2FA E800 005E 83EE; Offset 019
```

Svir - EN: A simple 512 byte virus with no side-effects. Svir means 'music' in Bulgarian.

```
Svir            33F6 4626 8B0C E302 EBF8 8BD6 83C2 04E8; Offset 049
```

Swami, Guru, Bhaktivedanta - CER: A 1250 byte 'Murphy' mutation containing the text 'Bhaktivedanta Swami Prabhupada (1896-1977). Detected by the 'HIV' pattern.

Swap - DR: Does not infect until ten minutes after boot. One bad cluster on track 39, sectors 6 & 7 (head unspecified). Uses 2K of RAM. Infects floppy disks only. Does not store the original boot sector anywhere. Virus creates a display similar to Cascade, but is transmitted via boot sector.

```
Swap            31C0 CD13 B802 02B9 0627 BA00 01BB 0020; Offset ?
```

Swedish Disaster - MR: The name is derived from the text inside the virus. The virus is awaiting analysis.

```
Swedish         0102 BB00 02B9 0100 2BD2 9C2E FF1E 0800; Offset 04A
```

Swiss-143 - CN: A simple 143 byte virus with no interesting effects.

```
Swiss-143       B44F 8BD5 EBBC C646 0000 45C7 4600 0D00; Offset 057
```

Sylvia - CN: The virus displays messages including 'This program is infected by a HARMLESS Text-Virus V2.1', 'You might get an ANTIVIRUS program.....' when an infected program is executed, but if the above text is tampered with, the (encrypted) messages 'FUCK YOU LAMER !!!!', 'system halted....$' will be displayed. The victim is told to send a 'funny postcard' to a genuine address of a Dutch woman called Sylvia. When an infected program is run, the virus will look for five COM files on drive C and the current drive. COMMAND.COM, IBMBIO.COM and IBMDOS.COM are not infected. The virus adds 1301 bytes to the beginning of the infected files and 31 bytes at the end.

```
Sylvia          CD21 EBFE C3A1 7002 A378 0233 C0A3 9E02; Offset 229
```

Sylvia-2, Sylvia B - CN: This version of the Sylvia virus has been patched to avoid detection, but appears functionally equivalent to the Sylvia virus. It is 1332 bytes long, just as the original, and detected by the 'Sylvia' pattern.

Syslock - CEN: This encrypted virus attaches itself to the end of a COM or an EXE file. Infective length is 3551 bytes. It infects a program one in four times when executed. Will not infect if the environment contains SYSLOCK=@.

```
Syslock         8AE1 8AC1 3306 1400 3104 4646 E2F2 5E59; Offset 0, 3551 bytes
```

Taiwan - CN: The virus activates on the 8th day of every month and overwrites the FAT and the root directory of drives C and D. Two versions are known with different infection lengths: 708 and 743 bytes.

```
Taiwan          07E4 210C 02E6 21FB B980 0033 F6BB 8000; Offset 0A0
Taiwan (2)      07E4 210C 02E6 21FB B980 00BE 0000 BB80; Offset 065
```

Taiwan-C - CN: A new 752 byte mutation of the Taiwan virus. The major effect is unchanged - destruction of the FAT and root directory on C: and D:

```
Taiwan-C        0B00 33F6 BB80 008B 0050 4646 E2F9 FE06; Offset 1FB
```

Taiwan-D - CN: Related to Taiwan-C but only 677 bytes. It is detected by the same pattern which is now located at offset 1F1H.

Tenbyte, Valert - CER: This virus was posted by accident to the V-ALERT electronic mail list recently. Adds 1554 bytes to infected files. Activates on 1st September corrupting data written to disk.

```
Tenbyte         1E0E 1F8D 36F7 04BF 0001 B920 00F3 A42E; Offset 0
```

Tequila - EMR: An encrypted, multi-partite, self-modifying virus from Switzerland. Contains encrypted text 'Welcome to T.TEQUILA's latest production', 'Contact T.TEQUILA/P.o.Box 543/6312 St'hausen/Switzerland'. No pattern for infected files is possible, but the boot sector does not change. Displays a crude Mandelbrot set pattern on screen. (*VB* June 91)

```
Tequila boot    B82A 0250 B805 028B 0E30 7C41 8B16 327C; Offset 01B
```

Terror - CER: This Bulgarian virus has not been analysed yet.

```
Terror          2E8C 1E41 0550 B859 ECCD 213B E875 3E0E; Offset 046
```

Testvirus B - CN: This 1000 byte virus is clearly written for demonstration purposes, as it asks the user if it should infect all COM files in the current directory or not. It has no harmful side-effects.

```
Testvirus B      018A 1780 FA00 7501 C3CD 2143 E2F3 2EA1; Offset 3B0
```

Tiny - CN: A mutation of the Kennedy virus only 163 bytes long. It has no side-effects other than replication. (*VB* Sept 90)

```
Tiny             408D 94AB 01B9 0200 CD21 B43E CD21 FFE5; Offset 088
```

Tiny Family - CR: This is a family of at least 10 Bulgarian viruses. The viruses are not related to the Danish 'Tiny' virus, but just like it, they do nothing but replicate. The lengths of mutations range from 133 to 198 bytes.

```
Tiny Family (1)  CD32 B43E CD32 071F 5F5A 595B 582E FF2E; Offset variable
Tiny Family (2)  2687 85E0 FEAB E3F7 931E 07C3 3D00 4B75; Offset variable
```

TPworm - PN: A 'companion' virus written by the author of the Vacsina and Yankee Doodle viruses. The virus has been distributed in the form of 'C' source code. The infective length and hexadecimal patterns, hence, depend on the 'C' compiler used.

Traceback, Spanish - CER: This virus attaches itself to the end of a COM or EXE file. Infective length is 3066 bytes. It becomes memory-resident when the first infected program is run and will infect any program run. If the date is 5th December or later, the virus will look for, and infect one COM or EXE file either in the current directory or the first one found, starting with the root directory. If the date is 28th December 1988 or later, the virus produces a display similar to Cascade one hour after infection. If nothing is typed, the screen restores itself after one minute. This display will repeat every hour. Spanish is an earlier version with a reported infective length of 2930 or 3031 bytes. (*VB* Sept 89)

```
Traceback        B419 CD21 89B4 5101 8184 5101 8408 8C8C; Offset 104, 3066 bytes
Spanish          E829 06E8 E005 B419 CD21 8884 E300 E8CE; Offset ?
```

Trackswap - DR: A small Bulgarian Master Boot Sector virus, which is awaiting analysis.

```
Trackswap        FBA1 1304 48A3 1304 B106 D3E0 8EC0 06BD; Offset 00E
```

Trilogy - ?: Virus awaiting disassembly.

```
Trilogy          9C55 568C CD83 C50A 8DB6 F6FF 56BE 2601; Offset 100
```

Tumen - CR: Two variants are known of this virus. Version 0.5 is 1663 bytes long and plays a tune when Ctrl-Alt-Del is pressed. Version 2.0 is 1092 bytes long, but has not been fully analysed.

```
Tumen            8CC8 488E D881 2E03 0000 0181 2E12 0000; Offset variable
```

TUQ, RPVS - CN: A simple virus from West Germany without side-effects. Infective length is 453 bytes.

```
TUQ              5653 8CC8 8ED8 BE01 012E 8B04 0503 0157; Offset 05E
```

Turbo 448 - CR: A 448 byte Hungarian virus which will infect COM files when they are opened, for example by a virus scanner, but not when they are executed. The virus contains the text 'Udv minden nagytudasunak! Turbo @'.

```
Turbo 448        890E 0201 8CD8 8EC0 5958 BB00 01FF E3A1
```

Turbo Kukac - CR: A 512 byte virus, which resembles the Turbo 448 virus, but is somewhat longer, 512 bytes. COMMAND.COM will crash, if infected with this virus.

```
Turbo Kukac      FFE3 8CD8 488E D8A1 0300 2D41 00A3 0300
```

Typo, Typo COM, Fumble - CR: Infects all COM files in the current directory on odd days of every month. If typing fast, substitutes keys with the ones adjacent on the keyboard. Infective length is 867 bytes. (*VB* Apr 90)

```
Typo             5351 521E 0656 0E1F E800 005E 83EE 24FF; Offset 01D, 867 bytes
```

USSR-311, V-311 - CN: A 311 byte virus, which does not seem to do anything else apart from replicating. 910620 was V-311 Skul

```
USSR-311         8BF2 83C6 0203 C12D 0300 0500 0189 04B4; Offset 099
```

USSR-492, 492 - CR: A Bulgarian virus which has not been analysed. The only available sample seems to be corrupted.

```
USSR-492         2E8B 1E01 0183 C303 B104 D3EB 8CD8 03C3; Offset 010
```

USSR-516, 516, Leapfrog - CR: This 516 byte Russian virus is the first virus which does not modify the beginning of the programs it infects, but places the jump to the virus code inside the host program.

```
USSR-516         431E 53C5 1F46 5F07 8B07 3DFF FF75 F283
```

USSR-600, 600 - CR: An encrypted, 600 byte Russian virus.

```
USSR-600         BE10 01B9 3200 8A24 80F4 DD88 2446 E2F6
```

USSR-696, 696 - CN: A 696 byte Russian virus awaiting analysis.

```
USSR-696         3C00 7412 8CC8 B10F D3E0 3D00 8074 07BA
```

USSR-707, 707 - CR: A 707 byte Russian virus awaiting analysis

```
USSR-707         83C3 0F33 C08E C033 F68C C040 3DFF 0F76
```

USSR-711, 711 - CR: A 711 byte Russian virus awaiting analysis.

```
USSR-711        C88E C08E D833 C08B F0BF 0000 BB00 01FF
```

USSR-948, 948 - CER: A Russian, 948 byte virus, which seems partially based on the Yankee virus.

```
USSR-948        5051 56B9 FF00 FC8B F28A 0446 3C00 E0F9; Offset 02D
```

USSR-1049, 1049 - CER: A 1049 byte Russian virus awaiting analysis.

```
USSR-1049       EB10 8CDA 83C2 102E 0316 2000 522E FF36
```

USSR-1594 - EN: A 1594 byte virus which uses a self-modifying algorithm. Asterisks in the pattern indicate the modifying byte.

```
USSR-1594       1E07 BB15 002E 8037 **43 81FB 3A02 7CF5; Offset 005
```

USSR-2144 - CER: A 2144 byte Russian virus, not yet analysed.

```
USSR-2144       1E06 33C0 8ED8 FB2E 8B94 1000 EC34 03EE
```

V-1 - DCR: This virus is one of the first to infect both the boot sector and programs. It is 1253 bytes long and destructive. When activated, it overwrites the disk with garbage.

```
V-1             8EC0 26A1 1304 4848 503D 0001 7203 2D3E; Offset 02B
```

V2P2 - CN: This virus written by Mark Washburn is related to the 1260 virus, but is more complicated. It will, for example, add a random number of 'garbage' bytes to the programs it infects, to make identification more difficult. No search pattern is possible.

V2P6 - CN: This virus is written by the same author as 1260 and V2P2, but is longer and more complicated. It uses several different encryption methods, which makes it impossible to provide a search pattern.

Vacsina - CER: Infective length 1206 to 1221 bytes (COM) and 1338 to 1353 bytes (EXE). After infecting a COM file, a bell is sounded. Infects any file loaded via INT 21H function 4BH (load and execute), i.e. COM, EXE, OVL and APP (GEM) files. Checks version number of itself (current is 5) and replaces with newer code. A member of the 'Bulgarian 50' (see Yankee). (*VB* June 90)

```
Vacsina (1)     8CC8 8ED8 8EC0 8ED0 83C4 02B8 0000 502E; Offset variable
Vacsina (2)     E800 005B 2E89 47FB B800 008E C026 A1C5; Offset variable
```

Vcomm - ER: This virus first increases the length of infected programs so that it becomes a multiple of 512 bytes. Then it adds 637 bytes to the end of the file. The resident part will intercept any disk write and change it into a disk read.

```
Vcomm           80FC 0375 04B4 02EB 0780 FC0B 7502 B40A; Offset 261
```

VCS 1.0 - CN: A 1077 byte virus which will delete AUTOEXEC.BAT and CONFIG.SYS when it activates. Generated by a German program called 'Virus Construction Set'.

```
VCS 1.0         89FE AC32 C4AA E2FA C35E 81EE 0301 56E8; Offset 00E
```

VFSI - CN: A simple 437 byte Bulgarian virus.

```
VFSI            100E 1FB8 001A BA81 00CD 21BE 0001 FFE6; Offset 1A3
```

Victor - CEN: A 2442 byte virus from the USSR which. The only known damaging effect is the corruption of the FAT.

```
Victor          8CC8 8BD8 B104 D3EE 03C6 50B8 D800 50CB; Offset 0C8
```

Vienna, Austrian, Unesco, DOS62, Lisbon - CN: The virus infects the end of COM files. Infective length is 648 bytes. It looks through the current directory and the directories in the PATH for an uninfected COM file. One file in eight becomes overwritten. Seconds stamp of an infected file is set to 62. A number of mutations, shorter than the original, but functionally equivalent, have been reported in Bulgaria.

```
Vienna (1)      8BF2 83C6 0A90 BF00 01B9; Offset 005, 648 bytes
Vienna (2)      FC8B F281 C60A 00BF 0001 B903 00F3 A48B; Offset 004, 648 bytes
Vienna (3)      FC89 D683 C60A 90BF 0001 B903 00F3 A489; Offset 004
Vienna (4)      FC8B F283 C60A BF00 01B9 0300 F3A4 8BF2; Offset 004, 623 bytes
Vienna (5)      CD21 0E1F B41A BA80 00CD 2158 C3AC 3C3B; Offset variable
Vienna (6)      8E1E 2C00 AC3C 3B74 093C 0074 03AA EBF4; Offset variable
```

Vienna-622 - CN: A new version of the Vienna virus from Bulgaria. It is detected by the Vienna (4) pattern.

Vienna-644 - CN: A 644 byte version of the Vienna virus, which does not infect programs every time it is run.

```
Vienna-644      BF00 01FC A5A5 A58B F252 B42C CD21 5A80
```

Vienna-645 - CN: A 645 byte mutation of Vienna, detected by the 'Vienna-1' pattern.

Vienna-822 - CN: The effects of this mutation have not been fully determined, but seem to involve the boot sector. It is detected by the pattern for GhostBalls published in the January edition.

Violator - CN: This is an unsually long mutation of the Vienna virus. It is 1055 bytes long and it activates on 15th August. The virus is awaiting analysis. (*VB* Apr 91).

```
Violator        BF00 01F3 A48B F2B4 30CD 213C 0075 03E9; Offset 00E
```

Virdem - CN: This virus was published in the R. Burger book *Computer Viruses - A High Tech Disease.* Originally intended as a demonstration virus, but now also found in the wild. Infective length is 1336 bytes. Two versions are known to exist with texts in English and German. (*VB* July 90)

```
Virdem          BE80 008D 3EBF 03B9 2000 F3A4 B800 0026; Offset 011
Virdem-1        BE80 008D 3ED7 03B9 2000 F3A4 B800 0026; Offset 011
Virdem-Gen      434B 7409 B44F CD21 72AC 4B75 F7B4 2FCD; Offset 098
```

Virdem-792 - CN: A destructive mutation of the Virdem virus, which will overwrite the first 5 sectors on all disks when it activates.

```
Virdem-792      431E 8CC0 8ED8 8BD3 B43B CD21 1FBE 5203; Offset 098
```

Virus-90 - CN: The author of this virus is Patrick A. Toulme. He uploaded the virus to a number of Bulletin Boards, stating that the source was available for $20. When an infected program is run it will display the message 'Infected', infect a COM file in drive A and display the message 'Done'. Infective length is 857 bytes.

```
Virus-90        558B 2E01 0181 C503 0133 C033 DBB9 0900; Offset 01E
```

Virus-101 - CN: This virus was written by the same author as Virus-90. The virus is encrypted and self-modifying. An infected file has the seconds field set to 62. Will not infect if the first instruction in the file is not a 'JMP NEAR'. Infective length is 2560 bytes, but COMMAND.COM length does not change. Awaiting disassembly.

Virus-B - CN: 'Test virus' which was made available on the *Interpath Corporation* BBS in the USA. It is a mutation of the South African virus, with the destructive code of the original disabled. The identification pattern is the same as for the South African virus.

Voronezh - CER: A Russian, 1600 byte virus, which overwrites the first 1600 bytes of the host, and moves the original code to the end, where it is written in encrypted form.

```
Voronezh        3E89 078E C0BF 0001 BE00 015B 5301 DE0E
```

VP - CN: Contains a variable number (1 to 15) of NOPs at the beginning followed by 909 bytes of virus code. When an infected program is run, the virus may attempt to locate, infect and execute another program.

```
VP              0001 FCBF 0001 B910 00F2 A4B8 0001 FFE0; Offset variable
```

Vriest - CN: This virus adds 1280 bytes in front of the COM files it infects. When it activates it will display 'Something's coming up ...', produce a high-pitched sound for a few seconds, and finally display 'Vriest of g greets Vic ear Moeli~'.

```
Vriest          B489 CD21 3D23 0174 32B8 2135 CD21 8C06; Offset 0
```

W13 - CN: A primitive group of viruses from Poland, based on the Vienna virus. They have no known side-effects and there are two versions, 534 and 507 bytes long. The version with 507 bytes has some bugs corrected.

```
W13             8BD7 2BF9 83C7 0205 0301 03C1 8905 B440; Offset variable
```

Warrior - EN: This virus adds 1012 bytes to any files it infects. It contains the following text: '...and justice to all! (US constitution) Dream over ... And the alone warrior is warrior. The powerfull WARRIOR!' Awaiting analysis.

```
Warrior         AC2C 8032 E403 F826 8035 01E2 F3B4 19CD; Offset 0AE
```

Westwood - CER: A 1824 byte mutation of the Jerusalem virus.

```
Westwood        4D0F CD21 8CC8 0510 008E D0BC 1007 50B8
```

Whale - CER: The infective length of this virus is 9216 bytes. The virus slows the system by about 50% and uses dynamic decryption of parts of its code. Much of the code is dedicated to disabling DEBUG. Does not run on 8086-based computers. (*VB* Nov 90).

Wisconsin, Death to Pascal - CR: This virus adds 815 bytes to the beginning of infected programs, and 10 bytes to their end. Infected programs may display the message 'Death to Pascal' and attempt to delete all .PAS files in the current directory.

```
Wisconsin       8B0E 0601 BE08 018A 0434 FF88 0446 E2F7; Offset 2F4
```

Wolfman - CER: A 2064 byte virus from Taiwan with unknown effects.

```
Wolfman         8EC0 BE04 0026 837C FC00 7404 46EB F6EA; Offset 07F
```

WWT - CN: Very simple, overwriting viruses, with no side-effects other than replication. Two versions are known WWT-01, which is 67 bytes long and WWT-02 with a length of 125 bytes.

```
WWT-01          B44E B901 00CD 2173 02EB 1EBA 9E00 B802; Offset 003
WWT-02          B44E B901 00CD 2173 02EB 10E8 0F00 BA80; Offset 003
```

XA1 - CN: The XA1 virus overwrites the first 1539 bytes of infected COM files with a copy of itself and stores the original code at the end of the file. On 1st April the boot sector will be overwritten, causing the computer to 'hang' on the next boot. The virus will also activate on 21st December and stay active until the end of the year. It will then display a Christmas tree and the text:

```
                Und er lebt doch noch: Der Tannenbaum!
                Frohe Weihnachten
XA1 (1)         B02C 8846 FF8B 7E00 884E FE8A 4EFF 000D; Offset 01E
XA1 (2)         0EE8 0000 FA8B EC58 32C0 8946 0281 4600; Offset 009
```

Yale, Alameda, Merritt - DR: This virus consists of a boot sector and infects floppies in drive A only. It becomes memory-resident and occupies 1K of RAM. The original boot sector is held in track 39 head 0 sector 8. The machine will hang if the virus is run on an 80286 or 80386 machine. If a warm boot is performed after the machine hangs, an uninfected disk will still become infected. It contains code to format track 39 head 0, but this is not accessed. Survives a warm boot.

```
Yale            BB40 008E DBA1 1300 F7E3 2DE0 078E C00E; Offset 009
```

Yankee - CER: This is a member of the 'Bulgarian 50' group of viruses, which consists of some 50 related versions, all written by the same person. Vacsina viruses belong to the same group. All the viruses in the group will remove infections by older versions, and the size varies from 1200 to 3500 bytes. The Yankee viruses will play the tune 'Yankee Doodle Dandy', either at 5:00 p.m. or when Ctrl-Alt-Del is pressed.

```
Yankee          0000 7402 B603 520E 5143 CFE8 0000 5B81; Offset variable
```

Yaunch, Wench - EN: A 2537 byte virus, which has not been analysed.

```
Yaunch          BE5C 012B DB8A 058A 2032 C488 0547 3BFA; Offset 9C2
```

Yukon - CN: A simple, 151 byte overwriting virus. Does nothing else apart from displaying the message 'Divide overflow'.

```
Yukon           01CD 218B D8B4 57B0 00CD 2151 52B4 40B9; Offset 01F
```

Zeleng - CER: Slightly modified variant of the Eddie (Dark Avenger) virus. It is 1800 bytes long and detected by the Eddie pattern.

Zero Bug, Palette - CR: Infective length is 1536 bytes and the virus attaches itself to the beginning of COM files. The virus modifies the seconds field of the time stamp to 62 (like Vienna). If the virus is active in memory and the DIR command is issued, the displayed length of infected files will be identical to that before the infection. When the virus activates, a 'smiley' (IBM ASCII character 1) may appear on the screen, and 'eat' all zeros found.

```
Zero Bug        81C9 1F00 CD21 B43E CD21 5A1F 59B4 43B0; Offset 100
```

Zero Hunt, Minnow - CR: A 416 byte overwriting virus which will only infect a file if it locates a large block of zero bytes.

```
Zero Hunt       521E B802 3DCD 2193 B43F 33C9 8ED9 41BA; Offset 0D3
```

ZK-900 - CER: A 900 byte virus, which plays a simple tune at regular intervals after an infected program is run.

```
ZK-900          B10C D3E2 8BD8 B104 D3EB 03D3 250F 0059; Offset 152
```

REPORTED VIRUSES

834 - CR: Infects COM files other than COMMAND.COM and modifies the Master Boot Sector so the computer hangs when booted.

1702 - CR: A new mutation of the Cascade virus. Some doubts whether it exists.

4870 - CER: An overwriting virus, which is compressed by LZEXE.

Arema - DR: Reported mutation of Den Zuk from Indonesia

Best Wishes-970 - CR: A modified version of the Best Wishes virus.

Captain Trips - CER: A mutation of Jerusalem, 1808/1813 bytes long.

Century A - CER: As Jerusalem-C, but activation date is 1st January 2000. Destroys FAT.

Century B - CER: As Jerusalem-C, but produces a wait during the execution of BACKUP.COM.

Chaos - DR: A new and changed mutation of Brain.

Freddy - CR?: Infects IBMBIO.COM

Freeze - CR: A 1024 byte virus which makes the computer 'hang' at random intervals.

```
Freeze          4545 5A45 B8EF EFCD 213D FEFE B800 0074; Offset 002
```

Greenpeace - EN: A primitive overwriting virus.

IKV 528 - CN: May be identical to a 528 byte mutation of Vienna, reported previously.

Italian 803 - CEN: Awaiting sample

Jerusalem-A - CER: does not display black-hole in the screen.

Jerusalem-B - CER: EXE re-infection bug removed.

Jerusalem-C - CER: no slow-down effect.

Jerusalem-D - CER: destroys FAT in 1990.

Jerusalem-E - CER: destroys FAT in 1992.

JoJo 2 - CR: A 1703 byte mutation of the JoJo virus.

Kemerovo B - ?: A US modification of the Kemerovo virus.

Kitty - ?: This is not a virus, just a harmless modified boot sector, which will display the same message over and over if it is loaded.

```
Kitty           FABB C007 8ED3 BC7A 020E E800 005E 1F83; Offset 080
```

Lazy - CR: A 720 byte virus which installs itself in unreserved memory causing a system crash if many large programs are run.

Missouri - D: some doubt whether it exists.

Nichols - D: some doubt whether it exists.

Park ESS: A new mutation of Jerusalem.

PC-Club - DR: Reported in Indonesia. Said to display a message every 30 minutes.

PC-Monster - DR: Closely related to Den Zuk.

Robert/Narvin - DR: An Indonesian virus which displays graphics on the screen.

Screen - CR: Infects all COM files in current directory, including any already infected, before becoming memory resident. Every few minutes it transposes two digits in any block of four on the screen.

Semlohe and Keongz - DR: An Indonesian virus based on Den Zuk, but producing sound effects.

Slayer - CEN: Reported to be a family of 5 or 6 viruses.

Supernova - DR: A harmful virus from Indonesia which will format the hard disk when the printer is used.

TROJAN HORSES

AIDS Information Diskette: Widely distributed disk which is an extortion attempt. Installs multiple hidden directories and files, as well as AIDS.EXE in the main directory and REM$.EXE in a hidden subdirectory ($ is the non-printing character FF Hexadecimal). (*VB* Jan 90)

```
REM$.EXE        4D5A 0C01 1E01 0515 6005 0D03 FFFF 3D21; Offset 0
AIDS.EXE        4D5A 1200 5201 411B E006 780C FFFF 992F; Offset 0
```

Twelve Tricks: A trojan replacing the DOS bootstrap sector with a dummy version. Damage includes corruption of the FAT and twelve effects which may be mistaken for hardware failure.

```
Twelve tricks   BAB8 DBBE 6402 3194 4201 D1C2 4E79 F733; Offset 033
```

QUICK UPDATE REFERENCE

New computer viruses known to *VB* as of the 24th June 1991 which have been added to the complete *Table of Known IBM PC Viruses*. For more information refer to the relevant entry in the table.

483, **777 Revenge**, **1024PrScr**, **Advent**, **Amstrad-877**, **Apocalypse**, **Apocalypse II**, **Bad Boy**, **Big Joke**, **CARA**, **Cemetery**, **Cinderella**, **Demon**, **Diabolik**, **Erasmus**, **Finger**, **Formiche**, **Frog's Alley**, **Goblin**, **Hero**, **Kiev**, **Little Pieces**, **Lucifer**, **Migram-1**, **Migram-2**, **Minimal-30**, **Pest**, **Phenome**, **Plastique 521**, **Protecto**, **Simulation**, **Skism**, **Sparse**, **Striker 1**, **Swiss-143**, **Trilogy**, **Tumen**, **USSR-311**, **WWT-01**, **WWT-02**, **Yaunch**, **Yukon**, **Zeleng**, **ZK-900**

SCANNER UPDATE

[This month's test-suite has expanded to include viruses received during the first three months of this year. Details are provided at the bottom of the page.]

IBM Triumphs Amidst the 'Vapourware'

Since last month, a number of changes have taken place: *Visionsoft* has withdrawn *VIRFIND* from its *Immunizer* package, it is soon to be superceded by an improved scanner with the provisional name of *SMARTSCAN*. A number of suppliers promised to provide updates this month but failed to deliver in time for our publishing deadline. The defaulters are *Central Point Software* (*Central Point Anti-Virus*), *RG Software Systems* (*Vi-Spy*), Microcom (*Virex-PC*), *Symantec* (*Norton Anti-Virus*) and *S&S International* (*Dr. Solomon's Anti-virus Toolkit*). (A beta-test of version 5.11 of the *Anti-Virus Toolkit* arrived just after this month's deadline expiry.)

Symantec informs us it is supplying version 1.5 to its customers, but having promised to send this update three times and subsequently failing to do so, this version must be considered 'vapourware'. To date, the company is known to have updated the *Norton Anti-Virus* signature file just once since the product's release eight months ago - we should be very interested to hear if any *VB* reader has *actually* received *any* additional updates. Interestingly, both *Symantec* and *Central Point Software* maintain forums on *CompuServe.* In *Symantec's* case there is no mention of *Norton Anti-Virus*, while *Central Point* does at least offer a forum dedicated to its offering - but has yet to post any signature file updates.

Updates were received from *ESaSS* (*TBSCAN*), *Frisk Software* (*F-PROT*), IBM (*Virscan*), *McAfee Associates* (*Scan*), *PC Enhancements* (*PC-Eye*), *Sophos* (*Sweep*) and *Total Control* (*VIS Utilities*). In almost all cases, these updates represent a catching-up exercise - catching-up with the virus writers, that is! There are a number of enhancements to report: *Bates Associates' VIS Utilities* now includes a small utility (*VISAFE*) to assist with the backup (and restoration) of hard disk boot sectors. *Frisk's F-PROT* now reports if it detects compressed executable files. Among the compressors it knows about are versions of *ICE*, *DIET* and *EXEPACK* and *PKLITE*.

The review of IBM's *Virscan* (see *VB*, May 1991, pp. 16-17) concluded that if someone took the time and effort to enter *VB* search patterns, he would end up with a scanner which would rival those at the top. Sue Roll at *HFC Bank* did just that and submitted the updated ADDENDA.LST file to IBM which in turn added some more new signatures of its own. How did it fare? The prediction has proved correct: in testing the revised version of IBM's *Virscan* against the original test-suite of 313 samples (see *VB*, April 1991, Page 8.), it achieved a score of 97.12% in 'Turbo' mode and 97.76% in 'Secure' mode - these results represent a substantial 20% improvement.

As in previous months, *Sophos' Sweep* and *Total Control's Viscan* programs are included for control purposes. The rationale behind this is twofold. Firstly, both companies have access to the complete *VB* virus collection. Secondly, their respective authors adopt different approaches to detection: *Sophos* restricts the number of search patterns to the absolute minimum to detect the maximum number of viruses. This means that the same pattern is used to detect more than one virus. Jim Bates - the author of *Total Control's* package - seeks uniquely to identify each individual virus. For a description of the testing protocol and declaration of interests, please refer to the April 1991 edition of *Virus Bulletin*.

Alarmingly, only four scanners detected Spanish Telecom file and boot sector infections, namely *Virscan*, *PC-Eye*, *Sweep* and *Viscan*. We mention this fact in view of a spate of recent infections by this destructive virus (see *VB*, June 1991, p 2.).

Test Conditions

All the scanners were run from a 3.5 inch diskette. Timing measurements include the time required to load the program from the diskette, perform any initialisations and (where applicable) automatic memory scans. Disk caching software was disabled.

Two different PCs were used for the tests. The first consisted of a Compaq Deskpro 386/16. This is a 16 MHz 386 ISA PC with 6 Mb RAM and two 42 Mb hard disks, each of which was partitioned into two 21 Mb logical drives. The hard disk speed test was conducted on a 21 Mb partition and consisted of 887 files (of which 316 were .COM or .EXE executables) occupying 20.5 Mb. The floppy test was conducted using a 360 Kb 5.25 inch floppy disk (Microsoft C V5.1 Setup disk) which contained 10 files, of which 3 were executable, and occupied 354,747 bytes. This PC was used for the timing tests and the boot sector recognition tests.

The virus test-set was installed on an Apricot Qi 486-25-320. This is a 25 MHz 486 MCA PC fitted with 16 MB of RAM and a 320 MB SCSI hard drive which was partitioned into 10 logical drives. Part of the extended memory was configured as a RAM disk thus providing drives A to M inclusive.

Viruses Added to the Test-Suite

Files infected with the following viruses have been added to the test-suite: 1226 (C); 3445 (C); 555 (C); 789 (C); Advent (C); Akuku (C); Best Wishes 2 (C); Bulgarian 1600 v2 (C); Bulgarian 1600 V21 (C); NTKC (C); Catman (C); Christmas Violator (C); Dark Avenger 3 (C); Deicide (C); Do Nothing 2 (C); Doom 2 (E); Faust (C); Fichv (C); Gergana (C); Hybrid (C); Int 13 (C-E); Internal (E); Iraqui Warrior (C); Jerusalem - 4th Black Friday (C); Jerusalem - Antiscan (C-E); Jerusalem - Kylie (C); Jerusalem - Nazi (C); Jerusalem - NV (E); Justice (C); Leprosy (C); MG-1 (C); MG-2 (C); MG-4 (C); Micro-128 (C); Minimal 45 (C); Mirror (E); Number 1 (C); Ontario (E); PcVrsDs (C-E); Phantom (C); Pixel 5 (C); Raubkopie (C-E); Sentinel 1 (C); South African 416 (C); StarDot-801 (C); SVC Version 3.1 (C-E); Testvirus B (C); Twelve Trick Trojan (C); Vienna 774 (C); Vienna 822 (C); Virdem 792 (C); Vriest (C); Wolfman (E); Yankee - TP39 (C); Zero Hunt (C).

The following boot sector virus is included: Spanish Telecom.

This increases the Comparative Review Test-Suite to 365 parasitic and eight boot sector infections.

IBM PC VIRUS SCANNERS (UPDATE)

RESULTS TABLE - SCANNING SPEEDS [TESTS 1(i), 1(ii), 2(i), 2(ii)] (See *VB*, April 1991, pp 6-7)

Product	Version	Supplier	Hard Disk 'Turbo'	Hard Disk 'Secure'	Diskette 'Turbo'	Diskette 'Secure'
CP ANTIVIRUS	1.0	Central Point	3:13	120:39	0:05	4:34
F-FCHK	1.16	Frisk (Skulason)	6:25	11:50	0:36	1:10
FINDVIRUS	4.31	S&S	1:11	2:22	0:36	0:41
HTSCAN	1.12	Harry Thijssen	2:18	3:35	0:39	0:52
NORTON A/V	1.01	Symantec	1:56	N/A	0:39	N/A
PC-EYE	2.0d	PC Enhancements	1:12	3:57	0:24	0:43
SCAN	7.2V77	McAfee Associates	3:50	7:00	1:01	1:31
SWEEP	2.26a	Sophos	3:42	5:30	0:40	0:52
TBSCAN	2.6	ESaSS	1:22	3:50	0:14	0:30
VIRSCAN	2.0.1	IBM	3:35	4:45	1:01	1:20
VISCAN	3.17	Bates/Total Control	3:18	3:24	0:19	0:24
VI-SPY	6.0	RG Software	3:02	5:01	0:31	0:55
VPCSCAN	1.1a	Microcom	1:07	4:11	0:17	0:46

RESULTS TABLE - SCANNER ACCURACY [TESTS 3/4] (See *VB* April 1991, pp 6-7.)

Product	365 Parasitic Viruses		8 Boot Sector Viruses		Accuracy Percentages	
	'Turbo'	'Secure'	'Turbo'	'Secure'	'Turbo'	'Secure'
CP ANTIVIRUS	306	323	7	7	83.91%	88.47%
F-FCHK	357	357	6	6	97.32%	97.32%
FINDVIRUS	335	335	6	6	91.42%	91.42%
HTSCAN	328	328	6	6	89.54%	89.54%
NORTON A/V	216	N/A	6	N/A	59.52%	N/A%
PC-EYE	349	350	8	8	95.71%	95.71%
SCAN	332	332	7	7	90.89%	90.89%
SWEEP	357	360	8	8	97.78%	98.66%
TBSCAN	324	330	7	7	88.74%	90.35%
VIRSCAN	348	352	8	8	95.44%	96.51%
VISCAN	365	365	8	8	100.00%	100.00%
VI-SPY	332	332	6	6	90.61%	90.61%
VPCSCAN	177	177	5	5	48.79%	48.79%

PRODUCT REVIEW 1

Fridrik Skulason

SafeWord Virus-Safe

Enigma Logic's SafeWord is a checksumming ('fingerprinting') program, intended to provide detection of unauthorised modifications to programs and static system files, regardless of whether the modifications are caused by a virus or by tampering. As such it is functionally comparable to *Sophos' VACCINE/DIAGNOSE* program reviewed in December 1990.

The full name of the package is '*SafeWord Virus-Safe*', but to avoid confusion with the *Eliashim Microcomputers' ViruSafe* package (*VB*, April 1990, p.18-19), it is just called *SafeWord* hereafter.

Documentation

SafeWord is supplied on a single 3.5 inch diskette with the documentation in the form of .ASC files. A typeset manual is available to registered users of the package.

The documentation is not flawless, containing occasional errors, but I noticed one statement to which some people will probably take exception:

> Computer viruses need not be bad. It is interesting to note that it is possible to create computer viruses to perform quite useful functions.

Several files provide all the information necessary to install and run *SafeWord*, and a README file contains an overview of the capabilities and installation of the package, followed by the admonishment:

> You shouldn't just dive in without reading the manual first, but since everybody else does, we assume you will too.

In fact, the installation process was simple and did not require careful scrutiny of the documentation files, although they were necessary when the configuration needed to be changed during later stages of the testing.

Installation

In addition to the documentation files, the package consists of an installation program, a utility to edit the list of files checked and SWVSAFE.COM - the fingerprinting program.

To install the program I ran the installation program. By default a line will be added at the end of CONFIG.SYS, although it is possible to load SWVSAFE from AUTOEXEC.BAT. The documentation states 'If you know of no incompatibilities or dependencies which may impel you to choose one method over the other, then the choice is arbitrary.' It also explains some problems which may arise (particularly regarding TSR incompatibility), for example that the program must not be loaded after *Borland's SideKick*.

The installation went without a hitch, and a directory was created containing several files, including SWVSAFE.COM, SWVEDIT.EXE and SWVCHECK.LST

SWVCHECK.LST contains a list of all files to be checked - describing when they should be checked, how often and which algorithm should be used. Initially it only checks the *SafeWord* programs and the operating system, although COMMAND.COM is omitted.

SWVEDIT.EXE is used to edit this file - to add new programs to be fingerprinted or to change algorithms and/or the frequency of checking.

"Despite its simplicity, this method is sufficient to detect all known non-stealth viruses and is fast, adding no noticeable overhead as files are checked and loaded."

Recommended Usage

SafeWord can be used in several different ways, but the documentation recommends two particular methods of operation, declaring that 'By setting up your SafeWord VIRUS-Safe defences to operate in BOTH of these modes, you will be able to assure overlapping defences that are virtually impenetrable.'

The first installation mode is to load SWVSAFE.COM from CONFIG.SYS. *SafeWord* will then be installed in memory, occupying 19 Kbytes and will check programs as they are executed, verifying their integrity. The documentation makes it clear that this method cannot be guaranteed to provide a defence against 'stealth' viruses and 'insider' attacks.

The second (and far more secure) installation mode is to create a 'clean' write-protected system diskette, with a copy of *SafeWord*, and boot the computer periodically from this diskette, running *SafeWord* in non-resident 'batch' mode. The documentation emphasises that this diskette should never be used for anything else, and when not used it should be physically locked up in a place where it cannot be accidentally inserted into a computer that may be contaminated. (This is perhaps 'overkill' as write-protection of the diskette is sufficient to prevent accidental contamination.)

Vulnerabilities

Are there any loopholes if *Enigma's* suggestions are followed and both methods are used?

If the machine becomes infected by a sophisticated 'stealth' virus, the infection may not be detected until the next time the PC is booted and checked from a clean system diskette. Also, the computation of the fingerprints is useless if the system is already infected with a virus when *SafeWord* is run initially - although the documentation does not mention it, a virus scanner should be run before *SafeWord* is installed as part of an initial integrity check. (The same is true of all checksumming programs.)

Performance

Programs can be checked at the time SWVSAFE is loaded or when they are executed. It is also possible to check a program only every n-th time it is run (as defined by the user). If the fingerprints do not match, an alert will appear on the screen, giving the user three options: Abort, Continue or Update.

When a new program is run, a different alert message appears, allowing the user to specify whether the file should be checked and the 'fingerprint' added to SWVCHECK.LST. It is possible to disable this 'learning' mode - if the /DISALLOW command line switch is given, only programs whose fingerprints are already present in the SWVCHECK.LST file can be run. New files can also be added by a 'wildcard' operation - telling *SafeWord* to compute the fingerprints for all .EXE files on drive C:, for example, is a simple operation.

SafeWord can use one of four different methods to generate the fingerprints for each program it checks. Two of the methods, ANSI X9.9 and ISO 8731-2, are considered very secure against tampering but also take much longer to compute than the other two modes (see *VB*, September 1989, pp. 7-8, October 1990, pp.10-11). One of these is a straightforward CRC, but the last mode, called 'Turbo' just computes a CRC for the first and last 512 bytes of the file and adds the length. Despite its simplicity, this method is sufficient to detect all known non-stealth viruses and is fast, adding no noticeable overhead as files are checked and loaded.

SafeWord automatically uses ANSI X9.9 to scan the operating system, SWVEDIT.EXE and SWVSAFE.COM on every start-up. On an 8088 machine used for testing, this took 58 seconds - which is an irritating delay and might cause users to disable the program. When *SafeWord* was installed on a more representative 25 MHz 80386 machine and instructed on reboot to check ten of the most frequently used program files using ANSI X9.9, the start-up time increased by 37 seconds. If the number of files had been larger, the delay caused by the ANSI X9.9 method might prove unacceptable to many users.

SafeWord was tested against several viruses, and the results were as expected - in the case of non-stealth viruses every infection was detected. This was also true of various small changes introduced manually by altering a randomly selected byte or two in programs or fingerprinted data files.

Only one stealth virus (4K) was used for testing, and although the infection was detected when *SafeWord* was run in 'batch' mode after booting from a clean system disk, it failed to detect the virus when it was active in memory and intercepting the attempts to read the infected file. This is not surprising - all other similar programs on the market fail in the same way. This danger is mentioned in the documentation and cited as the major reason for a periodic check run from a 'clean' system diskette.

One potentially hazardous oversight by the developers is that the file SWVSAFE.COM does not perform any integrity check on itself when installed for the first time. If it is infected before it is installed, the infection will not be detected, even in the case of an simple virus such as Vienna.

Conclusions

Running under the ANSI X9.9 checksumming standard *SafeWord* proved impracticably slow on a 4.77 MHz PC, when checking files. This is not so much a criticism as an observation - for this approach to prove tolerable, a fast clock speed really is essential! Its functional performance and efficacy in detecting viruses conformed with the program's documentation. The memory-resident 'turbo' checksumming mode proved extremely fast and is effective against those viruses which do not employ 'stealth' tactics.

The documentation does not mention *Windows*, and compatibility with *Windows* 3.0 was not tested. The authors are aware of software's shortcomings, particularly subversion by stealth viruses and conflicts with other TSR programs. Considering the price of this package (a relatively cheap US$ 62.50), I can recommend *SafeWord* in most circumstances. As with all checksumming packages, problems will arise on development machines, but where executable files remain static, *SafeWord* can provide effective protection if it is used correctly.

Technical Details

Product: *SafeWord Virus-Safe*

Developer/Vendor: *Enigma Logic*, Inc., 2151 Salvio St, Ste. 301, Concord, CA 94520, USA, Tel 415 827-5707, Fax 415 827-2593

Availability: IBM PC/XT/AT/PS2 or compatible, running MS-DOS 3.0 or higher

Version Evaluated: 1.13

Serial Number: None

Price: Distributed as shareware, with a 30 day evaluation period and a registration fee of US$62.50 for a single copy. Site licences available.

Hardware Used: IBM XT with a 4.77 MHz 8088 processor, two 5.25 inch 360 K drives and a 10 Mbyte hard disk, running MS-DOS 3.30

PRODUCT REVIEW 2

Dr. Keith Jackson

Knoxcard: Anti-Virus Hardware

Anti-virus hardware. Now that's different.

Knoxcard is the first piece of anti-virus hardware that I have reviewed for *VB*; it comprises a half length plug-in card (for the ISA bus), and a 5.25 inch floppy disk containing associated software. *Knoxcard* claims to be a virus prevention system which constantly monitors a PC for virus infection.

Packaging

In another first for *VB*, this is the first ever review in which I have moaned about the packaging, but there are limits. To avoid customer complaints the manufacturers of *Knoxcard* really are going to have to provide something better than the packaging that I saw. My review copy came with holes in the sleeve of the 5.25 inch floppy disk, which had been made by the pins of integrated circuits on the *Knoxcard* itself being pressed firmly up against the unprotected floppy disk. The cardboard wrapper around the product had also all but disintegrated.

Documentation

The documentation provided with *Knoxcard* comes in the form of a small booklet. The booklet actually contains two distinct manuals, a 'User Guide', and a 'Security System User Guide'. These are so distinct that the page numbering even begins again at number '1' half way through the booklet. I found this manual very confusing, with specific information being very difficult to find due to the lack of an index. In places the documentation is very technical, describing in great detail such things as MS-DOS interrupts. Although I'm pleased to see such information available, I have to query its place in a document entitled 'User Guide'. What percentage of users will understand a word of it?

According to the manual, the features offered by *Knoxcard* include reporting when any program is attempting to become memory resident, selective enabling and write-protection of disk drives, preventing booting from a floppy disk, prevention of direct disk-writes and/or reprogramming of disk ports. Given this list of features, it is not surprising that the *Knoxcard* manual claims that 'Knoxcard is more than a virus protection system. It is a complete security package ...'. I will concentrate on the anti-virus features rather than the more general security features.

It should be noted that it is an absolute necessity that hardware is added to a PC to prevent (under all circumstances) booting a PC from a floppy disk. The addition of such hardware is a major defence when trying to prevent the spread of viruses, and is a plus mark in *Knoxcard's* favour. I am fully aware that the makers of various 'security shell' software packages claim to be able to prevent booting from a floppy disk using software alone. These manufacturers are, at best, being 'economical with the truth'.

Incomprehensible Warning Messages

If *Knoxcard* detects anything 'unnatural' (their word) while a program is executing, one of several warning messages is provided, and the user is asked to provide answers to queries as to how execution should proceed. Examples of the type of these questions include:

```
Do you permit Direct Drive Programming?
Do you permit the Write on Boot Sector?
Do you permit Program reloading itself?
```

My objection to these messages, and others which space prevents me from quoting, is that many (most?) users will find them completely incomprehensible. I'd be hard pushed myself to define the circumstances in which a program could *never* indulge in direct drive programming. The manual attempts to explain these queries in detail, but I fear that it is talking to the deaf; users will just press a key and continue. The developers of *Knoxcard* acknowledge this problem when they state near the end of the manual that 'It is important that all types of user understand what the Knoxcard messages mean'.

It is not sensible to state (as the manual does) that *Knoxcard* is designed to prevent *all* current viruses and *all* future viruses. This implies that the designers have thought of *every* possible way of accessing a DOS file - a claim which beggars belief and, as this evaluation will show, is already disproved.

Dubious Pre-Installation Advice

With commendable prescience the *Knoxcard* manual recommends taking a complete hard disk backup before *Knoxcard* is installed. However the manual then recommends that the user should carry out a low level format of the hard disk! With great understatement it says that this latter step is 'not strictly necessary'. I'd be somewhat more forthright and explain that it may also be *disastrous* if the backup is corrupted (for any reason

"I don't care what these viruses have in common, as the mere fact of having to care about such things makes the protection provided by Knoxcard deeply flawed."

whatsoever). I'd never recommend such a step unless it was absolutely vital. You've been warned!

I installed *Knoxcard* in my PC clone (after first taking the precaution of disconnecting my hard disk, see later), and the *Knoxcard* initial screen appeared before MS-DOS was allowed to boot. I was requested to authorise booting from floppy disk, and then proceeded to test *Knoxcard's* capabilities by executing virus infected programs.

Things went rapidly downhill from here onwards.

Memory Clashes

Although the *Knoxcard* was obviously capable of driving my screen (remember the *Knoxcard* initial screen had appeared correctly), whenever *Knoxcard* intervened in DOS operation the result was a thoroughly locked-up PC. The only way to persuade anything to happen was to switch the power off and then on again. *Knoxcard* caused my PC to lock-up whenever a TSR was inserted into memory during the boot process (e.g. *Borland's Sidekick*), when any program made a direct disk access (e.g. *The Norton Utilities*), as well as when virus infected programs were executed.

In short *Knoxcard* appeared to be activating as described in the manual, but then immediately locked-up quite thoroughly. I tried to rectify this problem, but nothing could persuade *Knoxcard* to do anything other than lock-up when it activated . As for measures to remedy this problem, the manual simply explains how to place the *Knoxcard* at a different location in memory, and even by trying every possible combination it still gave the same result. All this occurred on a PC clone that has served me well for 5 years and has never exhibited problems with any other software.

Detection

Even though *Knoxcard* locked-up whenever it tried to report something, I was still able to use this 'feature' as a test to measure which viruses (if any) *Knoxcard* was incapable of detecting. If *Knoxcard* failed to 'activate' (i.e. locked-up) when a virus infected program was executed, then by definition *Knoxcard* had failed to detect the execution of this virus infected program.

I tested *Knoxcard* against each of the 111 parasitic viruses (180 individual virus samples) described in the *Technical Details* below. This entailed rebooting my PC for each virus tested, almost 200 times in total. In each case I always rebooted using power-off followed by power-on; firstly, because this was essential to ensure that a virus has not remained resident in memory, and secondly, because *Knoxcard's* repeated locking-up disabled the Ctrl-Alt-Del warm reboot anyway. The things I do for *VB*!

I cannot describe the results as being very encouraging. *Knoxcard* failed to detect the following viruses: 1260, 405, 492, 696, 707, Alabama, Anthrax, Blood, Lehigh, LoveChild, Lozinsky, Nina, Rat, Shake, Suomi, Sylvia, Turbo 488, Virus-90, Zero Bug (19 in total). *Knoxcard* also failed to detect at least one variant of each of these viruses: Anti-Pascal, Burger, Datacrime, Number of the Beast, SVC, Tiny, Traceback, Vcomm, Vienna, Virus-101, W13 (11 in total). *Knoxcard* failed to react to the execution of some thirty different viruses.

I don't know what these viruses have in common. Perhaps one of the *VB* virus gurus can spot the connection in how they evade the protection provided by *Knoxcard*. Actually I don't really care what they have in common, as the mere fact of having to care about such things makes protection such as that provided by *Knoxcard* deeply flawed.

Fundamental Design Flaws

Even if *Knoxcard* worked properly on my PC, I have deep reservations about its design, and I believe that the *Knoxcard* developers have somewhat missed the point as to why hardware can be inherently more secure than software when providing the same security function(s).

To provide a level of security that cannot readily be circumvented, it is imperative that a plug-in security card contains an on-board processor. The *Knoxcard* contains only an EPROM chip, a static RAM chip, and some TTL logic chips. It does not have an on-board processor. On a PC any program can have access to any part of the computer's memory. Therefore any functions provided by a piece of software residing in the PC's main memory can (in theory, and often also in practice) be circumvented by some other software program. It makes no difference that some of the software is stored in EPROM on a plug-in card.

The developers of *Knoxcard* have thought about constraints on security software, as they discuss just this point in the Introduction to the *Knoxcard User Guide*. Their solution is worth quoting in full:

> Any virus programmers thinking of decoding Knoxcard and finding out what it does and develop a virus won't be successful because, there are more than a dozen modules of the program which are mixed in different ways to create unlimited combinations of object code. This stops anybody from writing a single program which can snatch the control away from Knoxcard.

This is drivel, utter drivel. I'm sorry to be rude, but I cannot think of any more reasoned response to such a fatuous claim. How are all the articles about the intricacies of specific viruses written if they were not begun by disassembling the software of the virus and figuring out how it works? *VB* should set Jim Bates loose on the *Knoxcard* software/firmware, the claim that the *Knoxcard* software cannot be reverse-engineered would soon be shown to be untrue.

Readers may now appreciate why I begin such tests with my hard disk disconnected. It's always a useful precaution and the hard disk can easily be reconnected as required. *Knoxcard* specifically permits booting from floppy disk and does not require a hard disk. Nothing that locks up so completely as *Knoxcard* is going to be allowed near any hard disk of mine.

Cohen Vindicated

Knoxcard claims to know about all patterns of virus activity, but given the results described above this claim is entirely unfounded. I'm reminded of Fred Cohen's oft repeated attempts to define the difference between a virus (which is after all just a computer program), and an uninfected software program. After much thought he arrives at the inescapable conclusion that there is no inherent difference between the two (there may well be specific differences), and trying to detect what is a virus merely from its actions is therefore difficult if not impossible. *Knoxcard* is a system that attempts to achieve this feat, and not surprisingly falls flat on its face.

Technical Details

Product: *Knoxcard*

Developer: *Knoxware*, 403 House of Lords, No. 15 & 16 St. Marks Road, Bangalore, India, Tel +91 (812) 215506, Telex 0845 2335 RAVI IN.

Vendor in the UK: Compute Era Systems, 18th Floor, Station Rd., Wembley, Middlesex HA9 6DE. Tel 081 903 8657, Fax 081 903 9463.

Availability: IBM PC/XT/AT/386 or 100% compatible running PC-DOS or MS-DOS version 3.1 or higher. Less than 1 Kbyte of memory is required by the software associated with the *Knoxcard*.

Version Evaluated: 2.01

Serial Number: Nx-2800

Price: £99 in the UK.

Hardware Used: An ITT XTRA (a PC clone) with a 4.77 MHz 8088 processor, one 3.5 inch (720 Kbyte) floppy disk drive, two 5.25 inch floppy disk drives, and a 32 Mbyte Western Digital hardcard, running under MS-DOS v3.30.

Virus Test-Set: Test-suite of 113 unique viruses (according to the virus naming convention employed by *VB*), spread across 182 individual virus samples. Two boot sector viruses (Brain and Italian) and 112 parasitic viruses feature. There is more than one example of many of the viruses, ranging up to 12 different variants in the case of the Tiny virus. Where more than one variant of a virus is available, the number of examples is shown in brackets.

1049, 1260, 1600, 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 800, 8 TUNES, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), Anti- Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Cascade (2), Casper, Dark Avenger, Datacrime, Datacrime II (2), December 24th, Destructor, Diamond (2), Dir, Diskjeb, Dot Killer, Durban, Eddie 2, Fellowship, Fish 6 (2), Flash, Flip (2), Fu Manchu (2), Hymn (2), Icelandic (3), Internal, Itavir, Jerusalem (2), Jocker, Jo-Jo, July 13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (2), LoveChild, Lozinsky, MIX1 (2), MLTI, Monxla, Murphy (2), Nina, Number of the Beast (5), Oropax, Parity, Perfume, Piter, Polish 217, Pretoria, Prudents, Rat, Shake, Slow, Subliminal, Sunday (2), Suomi, Suriv 1.01, Suriv 2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Taiwan (2), Terror, Tiny (12), Traceback (2), TUQ, Turbo 488, Typo, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Whale, Yankee (7), Zero Bug.

PRODUCT REVIEW 3

Mark Hamilton

Trend Micro Devices' PC-cillin

A number of anti-virus companies are currently looking at new gimmicks to incorporate within their merchandise and judging by the products sent to the *VB* office, there is now a discernible trend towards a combined hardware and software solution. This could be viewed as a clever psychological device to sell more product as uninformed users might be fooled into believing that such a product was better because it included a hardware component - something tangible - as opposed to ethereal software. One such company is *Trend Micro Devices* of California and its offering consists of a 'dongle' and some accompanying software. The dongle is very similar to those that are still used to protect the more expensive software products such as *AutoCAD*. Fortunately the use of such devices is now on the decline following the strengthening of international copyright law and user-pressure to desist from such intrusive copy-protection methods.

In this case, the dongle is a pass-through device which must be attached to the parallel port before the software is installed, which *Trend* refers to as the 'PC-cillin Immunizer Box' but, in the interests of brevity and because it is in origin a copy-protection device, I shall refer to it as a 'dongle'. According to *Trend's* documentation, the dongle is used to store 'crucial data from the hard disk's partition table'.

Installation

The distribution diskette contains three binary files and a README file containing the latest information about this version of the package. The only executable program is PCC.EXE which acts as a combined install program and scanner. The PCC program contains three options: *Install*, *Quarantine* and *Make Rescue Diskette*. Unless you have remembered to unplug your (parallel) printer at the PC end, inserted the dongle at the now vacant port and plugged the parallel cable into the dongle before starting the PCC program, the only option you can use is '2) Quarantine'. 'Quarantine' is a misnomer as its purpose is to scan disks for the presence of viruses. *Trend's* idiomatic use of English is exemplified when this option is invoked, announcing 'Verifying for virus infected files'.

The Nightmare Scenario

If you have remembered to fit the dongle, 'Option 1' installs the software and this is when my particular nightmare started. Wisely, the installation process checked memory and the destination drive (it had to be drive C:) for virus infection. It created three files in the root directory (PCCILLIN.COM, PCCILLIN.IMG, PCCILLIN.FON), checked and wrote to the

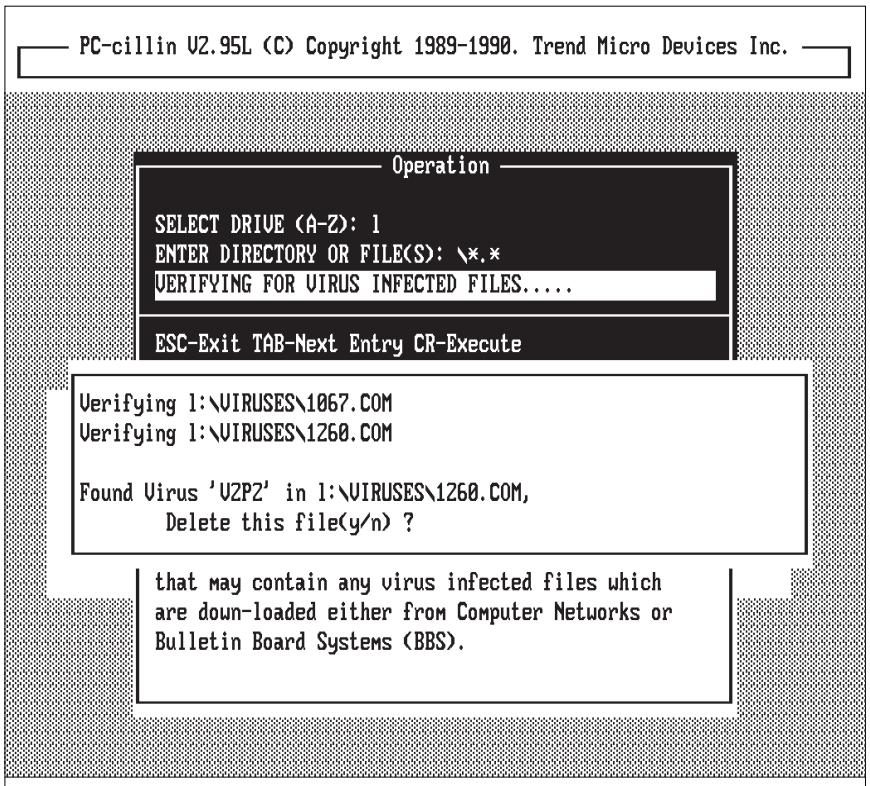
dongle, and (without requesting to do so) amended my CONFIG.SYS file. Then it told me to press a key to reboot my PC. Being a somewhat cautious fellow, I did not boot from the hard drive but from a write-protected system diskette. The PCC program had inserted the line 'INSTALL = PCCILLIN.COM' as the first line in CONFIG.SYS.

One would have thought that the developers might have envisaged the fact that users, like me, might actually be working with a grown-up PC equipped with extended memory requiring a memory manager such as *HIMEM* or even, *386MAX*. You're wrong, they have not - but maybe the developers know something I don't, so I rebooted the PC, this time from its hard disk.

If you have been following this saga so far, you would not be surprised to learn that my PC (an Apricot 486/25) did not boot-up correctly. The memory manager (which loaded *after PC-cillin*) complained about a stack violation. Commands to load memory managers must appear before all others in the CONFIG.SYS file otherwise the CPU becomes confused and somewhat perplexed. *Trend* states that its hardware/software is compatible with *Windows 3*, so its installation process really ought to check for the commonly-used memory managers such as *HIMEM*, *QEMM* and *386MAX* and, if any are found, place its installation line after that to load the memory-manager.

I edited my CONFIG.SYS to load *PC-cillin* last of all, saved the new version and rebooted. *PC-cillin* appeared to load correctly and then promptly hung the Apricot. I rebooted the PC off the floppy, removed the offending load line from CONFIG.SYS and tried running *PC-cillin* from the command line - it is, after all, a normal .COM file. Still no joy.

To cut a very long story short, in addition to my Apricot 486 running MS-DOS 5 I also tried installing *PC-cillin* on a Compaq DeskPro 386/16 running DR-DOS 5 and on a Dell 386/16 laptop running DOS 3.3. No matter which machine I tried, I could not get *PC-cillin* to install or execute properly. An inauspicious start to say the least!



Scanning report from *Trend Micro Devices* anti-virus product *PC-Cillin*. Menu driven front-ends and a fancy hardware addition do not compensate for the program's obstinate refusal to install and execute properly.

Documentation

This is an 80-page A5-sized perfect bound book (which, incidentally, refuses to stay open) containing an index and a table of contents. There is also a quick installation guide printed on a card.

The documentation uses a curious form of English. For example, the following appears on the installation quick reference card: 'You may need this diskette in case your hard disk is crashed'. In most respects the documentation is complete and answers the questions likely to arise - apart, that is, from questions about the products's refusal to install and execute.

Performance

The scanner claims to detect 176 viruses using search patterns. This rises to a more representative 260 samples using proprietary technology which Trend calls 'intelligent viral traps'. Its detection rate was low at 161 out of 365 parasitic infections and 3 out of 8 boot sector infections in the expanded test-suite (see *VB*, April 1991, p. 8 and this edition, page 34). While it took 15 seconds to scan the test 360K floppy, it took over five minutes to complete the hard disk scan test. In both instances, all files were checked. The scanner can search either a single file, a group of files, a single directory or a complete drive. I was unable to test the efficacy or otherwise of the memory-resident portion of *PC-cillin* which, according to the documentation, occupies 8 Kbytes of memory, since I could not get it to function!

In Conclusion

Having spent six hours vainly trying to get this product to work, I am not endeared to it! It is sensible to keep copies of your Master Boot Sector and DOS Boot Sector - but you don't need a dongle to do this! Copies can be kept on diskette. Some vendors provide a utility for this purpose, but you can always use *The Norton Utilities* (this process is much easier using *Norton* 4.50 rather than version 5.00) to make the necessary copies and to replace the relevant sectors if disaster strikes. The perceived added-value which hardware infers, is in this case, of little actual value. Dongles pose problems of compatibility - what happens if you have two, or more? Also, not all PCs have truly *Centronics*-compatible parallel ports. As a virus detector, *PC-cillin* leaves a lot to be desired. It does not detail which viruses it detects nor can you beef up its detection capability by adding your own patterns. In summary, there is little, if anything about this product to commend it.

PC-CILLIN

Product *PC-cillin* v.2.95L

Manufacturer *Trend Micro Devices Inc*., 2421 W. 205th Street, Suite D-100, Torrance, CA 90501, USA. Tel 213 782 8190, Fax 213 328 5892.

Price	US $139
Memory Check	Yes
Network Aware	Yes
Single File Check	Yes
Definition Format	Proprietary
Virus Removal	File deletion
Access to *VB* Test-Set	No
User Upgradeable	No
Resident Scanner/Monitor	TSR Monitor
Scanning Speeds	
Hard Disk - 'Secure'	5 mins 8 secs
Floppy disk - 'Secure'	15 secs
Scanner Accuracy	
Parasitic - 'Secure'	161 out of 365
Boot Sector - 'Secure'	3 out of 8
Accuracy Percentage	44.20 %

For an explanation of the entries in this table refer to the evaluation protocol published in *VB*, April 1991, pp. 6-7.

SHAREWARE REVIEW

PC Virus Index

This is a shareware database program which provides details about computer viruses which infect the IBM and compatible PC. Developed and maintained by Bryan Clough (a management consultant from Brighton, UK) and Dan McCool (whose activities include running a virus 'exchange' BBS in Germany), *PC Virus Index* is updated bi-weekly with the latest information about new viruses and anti-virus software.

Distributed in compressed form (the review copy was contained in file PCV305.ZIP) the program decompresses into a .DBF database file and an executable PCV.EXE. On the version inspected, the standard front-end menu flashed up a very prominent warning 'Unregistered Copy / Please Send in Your Order Form!'. To reaffirm this message the screen refuses to clear for some 20 seconds before further operation can proceed - presumably so that the user can read and inwardly digest the registration information.

No Help dialogue box appears on screen, so the user must arbitrarily cursor through the various dialogue boxes in order to get a feel for the management and layout of the database.

Acknowledgements

The actual details and history of the program reveal themselves only when the user wishes to exit from it. The Quit dialogue box reveals the registration details, acknowledgements, copyright notice and disclaimer. Acknowledgements go to Jim Bates (*Virus Information Service*, UK), Vesselin Bontchev (*Bulgarian Academy of Sciences*), Ian Leitch (*London School of Hygiene and Tropical Medicine*), Andy Sharpe (*Symantec UK*), Fridrik Skulason, Dr. Alan Solomon, *Computer Virus Catalog* (*University of Hamburg*), Patricia Hoffman, John McAfee, *Virus Bulletin* and *Virus-L* - in short a mixed bag of assorted saints and sinners.

Menus

The main screen menu comprises an index listing known and reported viruses which can be backed up by reference to a 'Report' file (providing what might best be described as 'technical details'), a 'Warning' file (which gives details of particular problems associated with each specimen) and a 'Treatment' file (which explains disinfection strategies).

Entries for each specimen include brief details including the virus' name and aliases, its infective length, date of its first being reported, number of strains and infective qualities (i.e. whether the virus infects boot sectors, programs, hard disks etc.) There is also a listing of six virus scanners (*Dr. Solomon's Toolkit*, *F-PROT*, *McAfee Associates*, *Norton Anti-Virus*, Bates' *VISCAN* and *TBSCAN*) with ticks appearing beside them if the product

'Reports' or 'Warns' of the particular virus under scrutiny. The distinction between a 'report' and a 'warning' is not clear and in the absence of a proper explanation of these subtleties, such classifications are meaningless.

The 'Scroll' option which guides the user through the index of viruses proved extremely irritating in the evaluation copy - it insisted on always returning by default to the first entry within the database. Considering that there are over 300 entries in the database, this makes viewing records towards the middle or end of the index a time-consuming and frustrating task.

Index Information

The first entry in the index supplied was the Anthrax virus which is described as being related (via the Dark Avenger connection) to the Proud, Evil and Phoenix viruses - however, when using the 'Find' facility to locate these three related specimens within the index, no matches were found! The information supplied about each virus is rather limited: in this instance the user is informed of the virus' infective length (1040-1279 bytes), that it infects the Master Boot Sector, COMMAND.COM, COM and EXE files and is memory-resident. It is also said to have been written in Bulgaria by Dark Avenger who loaded it up to bulletin boards in 1990. If one wanted to remove this virus from an afflicted PC, there is little in the way of helpful advice. COM files are described as '*usually reparable*' while a '*low-level format & FDISK are destructive. Use only if all else fails and full procedures and parameters are known*'. No further information is given.

In fact, the 'treatment' dialogue box for most (if not all?) of the viruses consists of a standard reference screen which provides cursory information about recovery by listing virus types and associated disinfection strategies. This information really is ludicrously truncated and should be regarded as nothing more than a signpost to more technical and heavyweight references. The Nomenklatura virus (which causes gradual corruption of File Allocation Tables so that contiguous clusters on disk are randomly transposed) receives exactly the same bland treatment advice as something as relatively innocuous as Jerusalem or Cascade!

Technical Reports

A number of the more detailed technical reports on virus specimens are culled straight from the *University of Hamburg's Virus Catalog* while more speculative reports (often unsubstantiated) are included from *Virus-L*. Using *Virus-L* as a source of definitive reference is an unwise strategy - Ken van Wyk, the conference moderator, has always argued that the forum is best suited for rapid communication, questions and answers and urgent alerts rather than for posting cold-blooded technical viral analysis. *Virus-L* has become the de-facto communications link between virus researchers worldwide and is an extremely important forum, but its purpose was never intended to be as an archive of technical information. Most of the virus reports culled from *Virus-L* are incomplete.

Those familiar with *VB's* virus analyses may well suffer from a disquieting (or comforting?) feeling of deja vu when reading certain *PC Virus Index* reports.

After cursory examination it soon becomes clear that a number of statements within *PC Virus Index* conflict with the findings of *Virus Bulletin.* As an example, the *PC Virus Index* says Dark Avenger 2100 causes 'file corruption, loss of bootability and system crashes', but none of these symptoms have been observed in live testing or diagnosed from disassembly. *PC Virus Index* also continues to perpetuate the myth that the Whale virus mutates in one of thirty-three different ways (both David Chess at *IBM* and Dr. Peter Lammer at *Sophos Ltd,* who have disassembled this virus confirm that it has thirty possible heads. Three extra heads were created accidentally by a Dutch researcher but are non-replicable).

Pink Elephants?

The most disturbing aspect of the *PC Virus Index* is the developers' willingness to include what appear to be completely groundless reports about viruses about which the research community has no knowledge whatsoever. In at least one case (the so-called 'Pink Elephant' virus), not one of the anti-virus software developers listed has reported the virus' existence. Other such virus vapourware includes the 'Headcrash' virus, the 'Compiler' virus, the 'Curse', 'Brunswick' and 'Austria' viruses.

Eclectic Reporting

The problem with eclectic reporting of this kind, whereby information is collated from numerous sources (all of which, incidentally, are in the public domain) without verification, is that such 'research' is wide open to the rumour-mill and to disinformation. The research community is desperately in need of a verified factual information database. For such a project to be genuinely useful, rather than the potential and actual agent of misinformation, it needs the active participation of genuine computer virus researchers and its custodian(s) would have to be technically competent with a working knowledge of virus disassembly. Simply hurling reports (many of which are of suspect accuracy) into a boiling pot, adding database flavourings and stirring for twenty minutes, is a recipe for bland confusion.

PC Virus Index v3.05

A shareware database program listing IBM PC and compatible computer viruses. The program is released bi-monthly and is available on Bulletin Board Systems. Compatible with IBM PC/XT/AT/PS2 running MS/PC-DOS.

Developers: Bryan Clough, 19 Walshingham Road, Hove, Sussex BN3 4FE, UK. tel 0273 773959, Fax 0273 778570 and Dan McCool, Giessener Strasse 44, 6308 Butzbach 3, Germany. Tel (+49) 6033 71507, Fax (+49) 6033 60758.

Annual site-licence registration fee: £195 / US$395 / DM595

Annual individual user registration fee: £20 / US$40 / DM60

END-NOTES & NEWS

Little further information has emerged about the apprehension of two Swiss virus writers for their alleged involvement in the development and distribution of the Tequila virus. (*VB*, June 1991, p.24). According to sources in Switzerland and Austria, the arrest occurred in the village of Steinhausen which is near Zurich. The UK's *Computer Crimes Unit* has received official complaints from British computer users and an application to Canton officials for extradition looks possible. Similarities between the Tequila virus (*VB*, June 1991, pp. 16-17) and the Flip virus (*VB*, September 1991 pp. 18-20) have been noted, suggesting the possibility of common authorship.

The UK's *Home Office* is planning to spend thousands of pounds on virus checking software following the discovery of the New Zealand virus on the widely used *CRAMM* risk assessment software package from *BIS Systems*. The virus had infected diskettes at the duplication stage. The *Home Office* central computing operation in Liverpool is considering supplying central offices in Whitehall, regional offices and prisons with virus checking software.

Computacenter, the UK PC systems house is to equip its 200 PCs and 120 service engineers with virus protection software. Senior manager Tony Marven said: 'We have not only made our internal *Computacenter* quality and security standards even more rigorous, but are also fully committed to providing our customers with the services and products needed in the implementation of anti-virus procedures.' *Computacenter* has adopted the *Vaccine* and *Sweep* software packages from UK company *Sophos Ltd*. All key staff with the organisation are to attend anti-virus training workshops prepared by *Sophos*. PC dealer *Businessland* is also aware of the virus problem, but the company has decided not to follow *Computacenter's* approach for the time being, due to the cost involved.

Total Control UK has formally announced the availability of the *VIS Utilities*, an integrated set of programs to combat IBM PC viruses developed by *Bates Associates* in a press release on June 3rd 1991. The *VIS Utilities* cost £50 for a single machine licence. Updates, provided on a monthly basis, cost £10 each. Alternatively, the software, including 12 monthly updates is available for £110. Technical support is provided by *Bates Associates* direct.
Tel 0488 685299.

Virus Bulletin Conference, Hotel de France, St. Helier, Jersey, 12-13th September 1991. Places are now limited and early booking is advised. Tel *VB Conference*, UK, 0235 531889.

The US *National Computer Security Association* (*NCSA*) is to issue a number of 'consumer reports' on anti-virus utilities with publication schedules planned through to March 1992. The reports will cover scanners, CRC and cryptographic checksumming programs, TSR monitors, hardware solutions, and recovery tools. Tel *NCSA* (USA) 202 364 8252.

S & S International is holding a two-day seminar on data recovery on 17-18th July 1991. The venue is Great Missenden, Bucks, UK. Tel 0442 877877.

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