February 1992 ISSN 0956-9979

VIRUS BULLETIN

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

CONTENTS

EDITORIAL

Some Recent FUBARs

Those inveterate hackers out there, who vainly pester anti-virus developers for virus samples with which to tinker, will be glad to know that virus code is increasingly available in both shrink-wrapped and cover disk form. As a first port of call, a regular visit to the newsagent is recommended. Careful scrutiny of the computer titles should yield results fairly quickly - some titles, it seems, offer a free virus with every issue at no extra cost.

The mass distribution of virus infected disks is the software industry's equivalent to the military's FUBAR - a 'Foul Up Beyond All Recognition'. *(The editor is informed that the vernacular etymology of this acronym may be subject to colourful Anglo-Saxon variations.)* FUBARs are on the increase with the tide of commercial software and cover disks contaminated at source rising at an alarming rate. Reviewed in isolation, none of the cases could be described as catastrophic but when listed in succession, the accumulative effect should send a shockwave through the offices of software houses everywhere. The following catalogue of carelessness is by no means comprehensive, but simply lists some of the more widely publicised debacles in recent history:

- On November 20th 1991, *Zinc Software* of Pleasant Grove, Utah, USA contacted its customers to warn them that a *Zinc Interface Library* update disk was infected with the Form virus. *McAfee Associates' Scan* and *Clean* software was hurriedly made available on *Zinc's* BBS.

- In December 1991 software supplied with *2the^{MAX}* VGA cards manufactured by Taiwanese company *Focus* was found to be infected with the Michelangelo virus. UK distributor *Scan International* had sold 1,000 of the cards (which produce 32,000 colours at a resolution of 800x600) but the company's director stated that only a small batch of the product was infected.

- On December 11th 1991, network software giant *Novell* warned 3,800 customers of a possible virus infection on the latest release of the *NetWare Encyclopedia* (supplied on a series of ten diskettes). The story was reported in the *New York Times*. (see *VB*, January 1992, p. 2).

- January 1992 saw the most serious incident of virus distribution in the United Kingdom. Computer magazine *PC Fun* distributed live virus code to high street retailers and newsagents nationwide. The source of infection was traced to an infected master disk (see story, page 6).

- Hard on the heels of the *PC Fun* incident, *ITS* (*Information, Trade & Software*) of Sofia, Bulgaria, is reported as distributing approximately 1,000 diskettes contaminated with the Dir-II virus to Europe, the USA and Canada.

Count Otto von Bismark, between spasmodic bouts of excessive violence, is recorded to have said 'only a fool learns by his own mistakes' - a harsh platitude admittedly, but typical of the man who unified Germany. The Iron Chancellor would have certainly felt a sense of *Schadenfreude* had he chosen a career in software development (a non-trivial task in late nineteenth century Europe). Whimsy apart, the evidence strongly suggests that future such accidents will only be averted when software manufacturers (large, small and indifferent) learn from others' misfortunes and embrace the practice of quality assurance.

What is striking about the aforementioned FUBARs is that, without exception, they were avoidable by compliance with rudimentary software QA. A check with any reputable virus scanner can pre-empt much wailing and gnashing of teeth.

Sending out a shrink-wrapped virus does little to enhance customer relations or inspire confidence in one's product range. It can also be expensive - mail and fax alerts, which most companies feel obliged to distribute in the wake of a FUBAR, are not cheap. Moreover, there are questions of civil and criminal liability to be considered.

Of course all of this hassle and loss of credibility can be prevented by a modest investment in the simplest tools and adherence to basic procedures. Software companies may ignore this fact at their peril, but their customers should not.

Archimedes World Ships New Virus

The February 1992 edition of UK magazine *Archimedes World* was withdrawn by its publisher (*Argus Specialist Publications*, Argus House, Boundary Way, Hemel Hemstead HP2 7ST, Tel 0442 66551) because the cover disk supplied with it was infected with a virus.

The magazine with a circulation of approximately 15,000 is aimed at users of the *Acorn Archimedes* microcomputer, of which there are some 120,000 in use in the United Kingdom. The virus had infected a master disk which was supplied to *Archimedes World* by software distributor *Cambridge International Software*. According to editor Andrew Banner, the master disk was checked for viruses using a scanner (there are about twenty viruses which infect this platform) but no virus code was found.

Alan Glover, a virus specialist with *Acorn Computers*, informs *VB* that the virus, provisionally called the Module Virus, was discovered in December 1991 and search routines for it have only recently been included in the few scanners available for *Acorn* machines. The virus contains no destructive code, its only trigger effect is to display a 'politely worded' diatribe against copy-protected software on 6th September.

Cambridge International Software has released two disinfection programs into the public domain, while *Acorn Computers'* virus scanner *!Killer* is available from national dealers.

TECHNICAL NOTES

Timeslice

Virtually all existing parasitic viruses fall into two groups - those which infect other programs when an infected program is run, and those which stay resident in memory and infect other programs when they are run or copied, or possibly when the user issues the DIR command. One of the new viruses this month uses a new trick. It intercepts INT 28H - the DOS idle interrupt which is called whenever DOS is just waiting for the user to press a key, and may infect programs from within this routine, which means that the virus may infect programs at random intervals, even when the PC appears to be idle!

4870 - Under Pressure

This virus is basically a primitive overwriting sample written in some high-level language, possibly C. The author compressed the resulting file with *LZEXE* 0.91, reducing its size from 8272 bytes, down to 4870. This is a fairly common practice among virus authors, to hide 'first-generation' samples from detection, as many scanners are not able to scan compressed executables.

The author of the 4870 virus went one step further - the virus actually *replicates* in compressed form. When an infected program is run, the virus writes itself over the beginning of other programs. Unlike other viruses, it will not write a copy of the current memory image, but rather a copy of itself, as it exists in compressed form on the disk.

As implemented this virus is not a serious threat. However, the concept could be enhanced and anti-virus developers should be prepared for more efficient examples in the future.

Polymorphism

Terms such a 'variable encryption', 'variable decryption routine', 'self-modifying encryption' to describe encrypting viruses which employ a 'pic'n'mix' strategy to variate the decryption stub are somewhat cumbersome.

A simpler term, recently proposed by Dr. Alan Solomon, is to use the description 'Polymorphic' for this class of virus, examples of which include the V2Pn series, Whale, PC-Flu 2, Haifa and the Maltese Amoeba virus.

So far, the only objection to the adoption of the term 'polymorphic' was raised by a spelling checker on the technical editor's PC which suggested the word 'Pornographic' as a possible alternative!

Nomenclature - A New Initiative

At the *Anti-Virus Product Developers Conference* organised by the NCSA in Washington DC last November (see *VB*, January 1992, pp. 6-7) a committee was formed with the objective of reducing the confusion in virus naming. The objective was to agree on a standard set of virus family names, but not to attempt to name every single virus. This committee, which consists of Fridrik Skulason (*VB's* technical editor) Alan Solomon (*S&S Enterprises*) and Vesselin Bontchev (*University of Hamburg*) has now finished its work, and can now present a definitive set of virus family names with which *Virus Bulletin*, the *University of Hamburg Virus Test Center*, the *S&S Enterprises'* product range, the *Sophos* product range and the *F-PROT* anti-virus package will broadly concur.

The resulting list, which will be published in *Virus Bulletin* next month comprises those viruses which were known to all three members of the committee on January 1st 1992.

The guidelines for selecting names were divided into the following three groups:

'Must'

1) Company names, brand names or names of living people should be avoided, except where the virus is provably written by the person. Common first names are permissible, but should be avoided if possible. In particular, names associated with anti-virus products, researchers or companies should be avoided.
2) Existing names should not be used unless the virus belongs to an existing family.
3) New names should be avoided if an acceptable name already exists.
4) Obscene or offensive names should not be used.
5) Researchers should not assume that just because an infected sample arrives with a particular name, that the virus actually bears that name.

'Should'

6) A name should only use the following characters: A-Z, a-z, 0-9, dash and space. Hyphens, commas, semi-colons or other punctuation marks and accented letters should not be used. Characters such as ! ? & / should be avoided.
7) Short names are preferable to long names. Sixteen characters should be considered a maximum, eight or fewer are preferred. If the short name is just an abbreviation of a longer name, use of the full name may be preferable.
8) Designations such as V845 should be avoided if possible. They should never be used as family names, as the members of the family may have different lengths.
9) Names such as Friday 13th or September 22nd should not be used as family names, as members of the family may have different activation dates.
10) Geographic names which are based on the location of the discovery site should not be used - the same virus might appear simultaneously in several different places.

'General'

11) When a virus first arrives without a name it may be assigned a temporary name such as 875, where 875 bytes is the estimated virus length. Other methods of assigning temporary names may be used, but it should be possible to distinguish between temporary designations and permanent names.

12) Any short overwriting virus of less than 60 bytes is designated 'Trivial-nn', where 'nn' is the virus length.

13) If numerous acceptable names exist, the original name, the name used by the majority of existing anti-virus programs or the more descriptive name should be ascribed.

Several of the names used by the *Virus Bulletin* in the past have violated one or more of these guidelines, so some changes are unavoidable. The results of the standardisation effort will be published in full in March. It is to be hoped that this standardisation will lessen the confusion amongst users of anti-virus products, literature and services.

Top and Tail Scanning

A number of virus scanning packages conduct a 'top and tail' (or 'turbo') search of program files in order to reduce scan run times. These searches usually entail checking the first and last 10-15K of an executable file on the assumption that virus code will reside within these limited areas of the infected program.

The Einstein virus (*VB*, January 1992, pp. 16-17), the Brainy virus (*VB*, December 1991, p. 2) and a virus called Leapfrog (*Virus News International*, January 1991, p. 13) operate in a manner that may be missed by scanners run in turbo mode. These specimens infect by inserting their code *into* a file, and their exact location within an infected file varies - one method by which this is done involves placing a second JMP instruction pointing to the virus code at the location to which the first JMP instruction (i.e. the first three bytes of most COM files) points. If this location extends beyond the parameters of the scanner's designated search area, the virus infection will not be detected.

This new tactic is probably an intentional measure to subvert virus scanning programs. Some scanner developers only offer a 'top and tail' or 'turbo' search mode - it is probable that the search engines employed will need to be fine-tuned (or massively overhauled as the case may be) in order that they conduct a thorough byte-by-byte search.

Manufacturers which include *VB* search patterns in their scanners should be aware of this development. The 24 byte patterns which *VB* publishes are intended for use in a *secure* scanning mode whereby the entire file is examined.

Multiple file infection, the infection of overlay files and viruses which insert themselves into files at variable locations, provide ammunition to the proponents of secure scanning. Scanners, such as that used in the British *Eliminator* package from *PC Security Ltd* (*VB*, August 1990, pp. 21-22), which gain their blistering speed by searching in precise locations for known virus code, will also have to compromise on such surgical precision in the face of this new class of virus. This 'scalpel' approach, which reduces run-times enormously, is even more area-specific than a top and tail scan.

Some checksumming programs also 'top and tail' protected files in order to reduce run-times - this practice is fundamentally mistaken and many poorly implemented checksumming programs fail to detect the modification to files infected by this sort of virus. As previously stated in *Virus Bulletin*, secure cryptographic checksumming dictates that protected files are checksummed in their entirety.

Warning: SuperStor and Automatic Disinfection

If you are using a scanner which contains disinfection capabilities - either fully or semi-automatically invoked - you should exercise considerable care when scanning compressed partitions created by *Addstor's SuperStor* program - including the version shipped with DR-DOS 6 from *Digital Research*.

If you inadvertantly allow the scanner to check the *SuperStor* fake partition file called SSPARTSS.SWP you might well discover a virus signature as it has shown a propensity to trigger false positives on a number of current scanners. This file is highly compressed and its integrity is essential. You should not allow any write or deletion operation on this file (as would be the case if an automatic disinfector were used) since this could result in a loss of or damage to existing files.

Safe Formatting

MS-DOS 5 contains as a default a 'safe format' mode whereby a disk may be recovered in the event that it is accidentally formatted. In contrast to the trusted DOS FORMAT command under earlier releases of DOS, this 'safe format' mode overwrites only the boot sector, FAT and root directory of the disk. These three components are themselves rewritten to other sectors on disk, the rationale behind this being that the essential initialisation and configuration code of the disk can be restored.

The safe format option will destroy virus code on disk because the boot sector is overwritten and program files are no longer invokable due to the formatting of the FAT. However, if the disk is subsequently recovered using the unformat option, any virus code present on disk will be restored (if it is has not been overwritten). The command line option FORMAT /U enables a full format of the disk whereby all tracks, heads and sectors are overwritten, and the use of this option is recommended when disinfecting disks.

The dangers of formatting a disk unintentionally were reduced when the nice people at *Microsoft* added the immortal phrase 'Are you sure?' to the format request sequence.

Freddy - A New Infection Technique

A new virus called Freddy (as yet unseen by *VB*) has been reported in Israel. It reputedly uses a novel way of hiding its presence from anti-virus programs. The first report of Freddy was received in November 1991. Almost simultaneously Vesselin Bontchev publicised the fact that the same DOS weakness had been discussed on virus bulletin boards in Bulgaria as a possible virus infection method (*NCSA Anti-Virus Product Developers' Conference*, Washington DC).

During bootstrapping, the program in the DOS boot sector loads the contents of the first file present on disk (usually called IBMBIO.SYS, IBMBIO.COM, IO.SYS or similar) on a sector-by-sector basis, while almost all anti-virus programs check its contents on a file basis (i.e. as a chain of clusters). This virus is reported to copy the first cluster of this file, overwrite the original with its own code and modify the appropriate entries in the File Allocation Table (FAT) and the root directory in such a way that the first cluster of the file points to the copy of the original cluster. When the system loads the file during bootstrapping, virus code is loaded and executed, while an anti-virus program checking the file (by relying on the information in the FAT) will find it unmodified.

Floppy disks are the most common carrier of virus code at present. A virus which uses such a trick can only infect and spread via system floppy disks. The vast majority of disks which are passed around are not system disks, which should be a crucial factor in slowing the spread of any such virus.

Virus Bulletin Conference 1992

Call For Papers

Abstracts of between 300 and 1,000 words are invited for proposed papers to be presented at the *Second International VB Conference* which will take place in Edinburgh, 2nd-3rd September 1992.

The conference will be in two streams: stream one will address the management of the virus threat within the corporate environment, while the second stream will concentrate on technical developments including virus disassembly, detection and classification.

Abstracts are welcomed from individuals or groups active in research, software or hardware development, quality assurance, the law, corporate security management, or any other field related to countering computer viruses and malicious software.

Abstracts, which should be completed by February 15th 1992, should be sent to The Editor, Virus Bulletin, 21 The Quadrant, Abingdon Science Park, Abingdon, Oxon OX14 3YS, UK. Fax 0235 559935.

Virus Prevalence

The following tables provide only a partial insight into the extent of virus penetration in the UK as many other agencies involved in virus control do not maintain statistics or do not make such data available.

Virus Prevalence Table

The following table is a breakdown of virus infections in the UK reported to *VB* during November 1991.

Virus Name	Reports	% Total Infections
New Zealand II	8	23.5%
Form	8	23.5%
Tequila	3	8.8%
Michelangelo	3	8.8%
Cascade	2	5.8%
Joshi	2	5.8%
Spanish Telecom	2	5.8%
Eddie II	1	2.9%
Nomenklatura	1	2.9%
2100	1	2.9%
Maltese Amoeba	1	2.9%
Other	-	0%
Total	32	100%

During December 1991 and January 1992 Form superseded the ubiquitous New Zealand II virus to become the most common virus reported to *Virus Bulletin*. A full table of statistics showing monthly virus infections between January 1991 and January 1992 will appear in March.

Virus Prevalence Table

The following table is a breakdown of virus infections in the UK reported to *VB* during December 1991.

Virus Name	Reports	% Total Infections
Form	7	31.8%
New Zealand II	5	22.7%
Tequila	2	9%
Flip	2	9%
Joshi	1	4.5%
Eddie II	1	4.5%
Yankee	1	4.5%
1575	1	4.5%
Keypress	1	4.5%
Maltese Amoeba	1	4.5%
Other	-	0%
Total	22	100%

CASE STUDY

The *PC Fun* Debacle

Cases of mass duplication of virus infected disks and software have been much in the news recently. A recent incident in the United Kingdom may result in the tightening of standards in the publishing industry.

CIX - The CNN of Personal Computing

The *Compulink Information Exchange* (commonly referred to as *CIX*, 071 399 5252, any modem speed) is often the first forum to post alerts about major computer virus incidents in the United Kingdom. The virus conferences, of which there are three, are often inaccurate and ill-informed but make entertaining reading, punctuated as they are by occasional outbreaks of hostilities between the 'experts'.

A valid argument to justify the relatively modest expense of joining and using the *CIX* virus conferences is that amidst the more nauseating examples of technical muscle-flexing, backslapping and mutual sycophancy (characteristics, be warned, which seem to predominate on this forum), there can be found the occasional nuggets of hard intelligence.

Those battle-hardened veterans who can recall *PC Cyborg Corporation's* direct mail campaign of December 1989, will recall that it was *CIX* which emerged as the electronic sorting house of information about the immediate impact and extent of the infamous AIDS Information diskette (see *VB*, January 1990, p. 7). The inaccuracies at the time were legion - certain individuals kept screaming 'virus' long after the actual functioning of the Trojan had been reported, while one person even insinuated that UK researcher Jim Bates had perpetrated the crime! However, tin-pot theorists apart, the majority of contributions to *CIX* at the time enabled a reasonably clear electronic jigsaw picture of events to develop.

Similarly, it was on *CIX* that messages were first posted concerning *PC Today's* mishap in August 1990 when some 56,000 disks were duplicated containing Disk Killer virus code (*VB*, August 1990, p. 3, September 1990, p. 2). Fortunately, in that instance, the virus code was inoperative but the incident caused shockwaves to reverberate through the hallowed editorial offices of many a PC magazine. Editors of many hobbyist magazines started investing in anti-virus software. Some magazines even solicited the personal services of Dr. Solomon in an attempt to restore the credibility of that most potent of marketing devices - the 'free' cover disk.

A Salutary Lesson?

One might be forgiven for thinking that the *PC Today* debacle would have taught a salutary and unforgettable lesson to editors and software publishers throughout the UK. Alas, it was not to be; postings to *CIX* on January 4th 1992 indicated that yet another UK computer hobbyist magazine had distributed a virus infected cover disk and this time the virus code was functioning. Once again, *CIX* contributors, rather like *CNN*, were first with the news.

Crazy Sue's Gets A Nasty Infection

PC Fun published by *MC Publications Ltd*., Unit 29, Riverside Business Centre, Victoria Street, High Wycombe, Bucks HP11 2LT, a monthly magazine reporting on games software for use on ST, PC, Amiga, Commodore 64 and Console platforms had reportedly distributed a diskette infected by the ubiquitous New Zealand 2 virus (aka Stoned).

Disks (containing the intriguingly titled 'Crazy Sue' software) purchased with the magazines were inspected on January 5th. The two scanners used indicated that New Zealand virus code was present in the boot sectors of both 5.25 inch disks - closer inspection with *Norton* showed the familiar 'Your PC is now Stoned' text in the boot sector with the genuine boot sector in Track 0, Head 1, Sector 3 of the diskette, as would be expected. A quick test on a 'dirty PC' (an Amstrad 1640) showed that the virus was functioning - once the virus was active in memory it infected fixed disks and diskettes indiscriminately. The *PC Fun* disks were not write-protected.

According to *PC Fun's* editor, Mr. Adrian Pumphrey, the print run of the magazine was approximately 20,000 copies (the actual duplication run of the diskette itself was later established at 18,000 copies) of which the company expected to sell in the region of 7,000 copies.

It appeared that the production master disk, bearing a games program called Crazy Sue and supplied by a sister company in Germany, had been infected. No virus checking was conducted at any stage during the preparation, duplication or predespatch stages of manufacture and distribution. The diskettes had been duplicated by *Copytec Software Solutions* which is based in the same building as *PC Fun*.

Liability?

Despite its discovery on 4th January, *PC Fun* and the infected disk on its front cover were still available on high street retail shelves on 12th January 1992. Neither the publisher, *MC Publications*, nor the distributor, *Comag Magazine Marketing* of West Drayton, withdrew it and the magazine remained available at both *WH Smith & Son* and *John Menzies* outlets for many days after an official press release warning from Scotland Yard's *Computer Crime Unit*. *Virus Bulletin* warned both *WH Smith & Son* and *John Menzies* as to the existence of the infected disks by telephone on 7th January and by first-class letter to arrive on the morning of 8th January. Both *Menzies* and *Smiths* immediately instructed that the magazine be withdrawn from sale - its subsequent appearance on shelves was due to supervisory difficulties.

Questions of criminal and civil liability have arisen as a result of this incident. The magazine was not recalled by the publisher (*MC Publications*), nor by the distributor (*Comag Magazine Marketing*) although the latter organisation did contact the major retail chains to inform them about the infected disks. There has been some debate as to whether prosecution of either organisation under *Section 3* of the *Computer Misuse Act 1990* would be feasible should a public complaint be registered with the police. Unauthorised modification of a computer system, such as that caused by any virus, is an offence under the terms of this Act and in this instance the offending code was still on sale with these organisations' full knowledge. At the very least a charge of negligence could be made by an aggrieved party against the publisher, the distributor, or any retailer knowingly selling virus infected software.

Enforcing Standards

In the light of this incident, the most positive approach would appear to be to apply pressure to those commercial organisations with sufficient clout to enforce standards.

The two largest UK retailers (*WH Smith & Son* and *John Menzies*), upon notification of the problem, acted promptly and responsibly in their attempts to safeguard their customers. As one wag commented one week after its official withdrawal from sale 'copies of *PC Fun* are now as rare as rocking horse droppings.' However, if these major retailers of computer magazine titles were to demand conformance with basic QA procedures from their suppliers, the risk of future debacles such as this would decrease dramatically.

With this in mind, we recommend that *VB's* UK readership (which comprises hundreds of organisations with significant clout) writes to the two major UK retailers and requests that both companies impose software quality assurance standards with which computer magazine publishers and/or distributors must comply. The rather depressing rationale behind this campaign is that many publishers will only introduce basic software QA if they find their titles denied lucrative shelf space in the high streets.

Useful addresses:

Customer Services
John Menzies Holdings
Hanover Buildings
Rose Street
Edinburgh EH2 2YD

Customer Services
WH Smith & Son
Greenbridge Road,
Swindon SN3 3LD

Postscript

The January 1992 edition of *PC Fun* may become a collectors' item as it is the last to be published - a decision (unrelated to the virus infected cover disk) was taken at the end of last year to drop the title from publication.

QA TUTORIAL

Richard Jacobs

Virus-Free Software Manufacture

Mass distribution of infected software is a potential nightmare for any software company or disk duplication house. This tutorial provides a brief introduction to the tools and techniques which can be deployed to prevent such debacles.

Basics

Viruses are executable code just like ordinary programs - however, viruses use covert methods to ensure that they run. There are only two methods by which code, other than ROM, is executed on a PC. The first is by loading and executing files stored on disk. The second is through the bootstrap process.

When a PC is switched on it performs internal checks of its memory and hardware, it then attempts to read the first sector of the disk in the first floppy disk drive. If there is no disk in the drive, the PC will try to read the first sector of the first fixed disk. Once one or other of these bootstrap sectors has been successfully loaded into memory, execution is passed to the code in that sector. On floppy disks this code loads MS-DOS and transfers control to it. On fixed disks there is one extra stage. A fixed disk can be divided into several partitions which can be used for different operating systems such as UNIX, or as multiple DOS drives (C:, D:, etc.). The function of the first sector on the fixed disk (the Master Boot Sector) is to locate the bootable partition and this is the partition which will be used by default. Once the bootable partition has been located, its first sector is loaded and execution is transferred to it. (This active partition boot sector is directly comparable to the floppy disk boot sector.) MS-DOS is loaded and control is passed to it.

A virus can attack this boot process by replacing the code in a boot sector with its own code. Normally the contents of the original boot sector are copied somewhere else, so that the next time the PC is booted, the virus executes *before* the boot code. Once the virus is memory-resident it normally loads the original boot sector and returns control to it. The extra time taken for the PC to boot is not normally apparent to the user. Once MS-DOS is running, the only way that a virus can be executed is if an infected file is executed. When this happens the virus again executes before the normal program and either makes itself memory-resident or infects other files, before returning control to the normal program. Again, the extra time taken for the program to execute is not normally noticeable.

In general, safe computing dictates that PCs should not be allowed to boot from untrusted floppy disks and untrusted programs should not be used.

Access Control

Access to PCs used for software development should be strictly controlled to prevent accidental or intentional virus infection.

Measures might include the use of power-on passwords and mechanical locks which make it difficult for unauthorised people to use the PC. This should prevent data being read from either the internal fixed disk or floppy disks, until the correct password, or physical key, has been used. It should be borne in mind that access control software cannot prevent a boot sector virus becoming active.

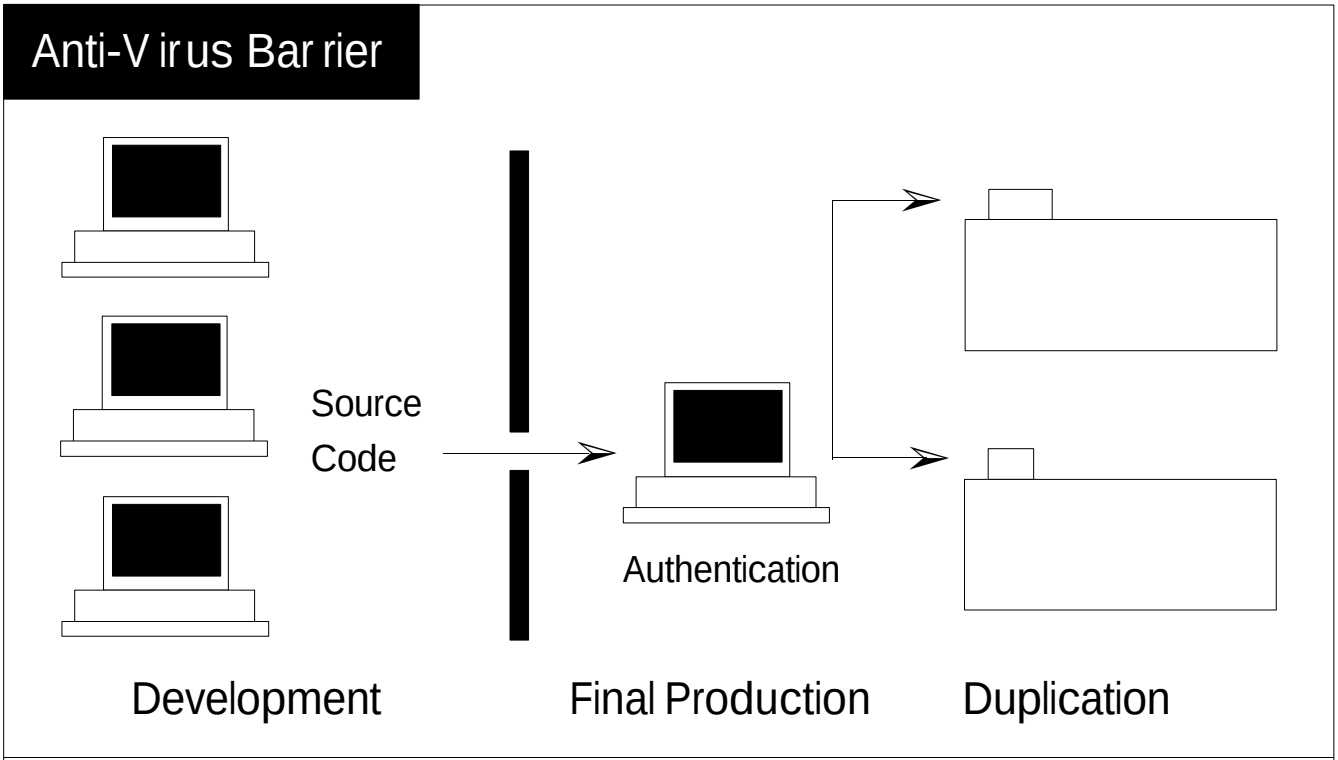
Any disk can carry a boot sector virus; there is no such thing as a non-bootable disk. Even non-system disks have a boot sector which will be loaded and executed if the disk is in the first floppy drive when the PC is booted. It is therefore possible to infect a PC with a boot sector virus by booting it from a non-system disk. This often occurs when a data disk is left in the disk drive when the PC is switched off. When the user next switches the PC on (without checking the disk drive) and the PC attempts to boot from the floppy disk, it becomes infected. Floppy disks should thus be left in disk drives for the minimum time necessary and should be removed as soon as the current operation has been completed.

Untrusted Software

It is usually necessary to run some commercial software on development PCs, such as compilers, assemblers, editors and diagnostic tools. These should be obtained from reputable suppliers and should be loaded from the original master floppy disks. Programs should be tested on other computers for an extended time prior to installation.

It is not sufficient just to check the master disks with virus scanning software; there are two major shortcomings of virus scanning software. Firstly, it cannot detect unknown viruses. Secondly, many new software applications are supplied in a compressed format and can only be installed using a dedicated installation program that decompresses the main executables. Any viruses attached to files that have been compressed will not be detectable until the files have been decompressed. The software should be installed on another, non-critical, PC and checked for viruses.

Unknown viruses can be detected reliably using checksumming software that identifies changes in executables. Viruses can be designed so that they become infectious only after a specific trigger condition has been met, such as after a particular time or date. Such a virus would not be detectable until it infected another executable.



Source code is transferred from the development area to the production area where it is compiled on an authenticated machine. The compiled code is delivered for duplication on clean write-protected master disk

New applications should be left on a non-critical PC for as long as possible and tested extensively there. It is impossible to be 100% certain that any such executable does not contain an unknown virus, if the virus is not active. Non-essential software should not be used on development computers.

All master disks should be write-protected as soon as the packaging is opened, before they are placed in any disk drive. Once the software is known to be clean it should be loaded onto all development PCs directly from the master disks. Applications should not be copied between machines.

After all applications have been loaded, development PCs should be 'fingerprinted' using a cryptographic checksumming package. This should calculate checksums of all the fixed executables on the PC's fixed disk, including the Master and DOS boot Sectors. For reliable virus detection the checksumming program and checksums should be stored on a system floppy disk, which should be write-protected after the checksums have been calculated. This disk should also contain a virus scanning program to check transient executables under development, that have not been fingerprinted. Development PCs should always be booted from this system floppy disk, thus providing absolutely reliable detection of any virus that attacks any of the fingerprinted executables on the system.

Networks

Development PCs should not be connected to networks, unless this is absolutely necessary for access to printers or backup devices.

If a network connection is allowed then extreme care should be taken. While it is possible to implement good anti-virus measures on a carefully managed network running an operating system such as *Novell NetWare*, it is very easy to leave or create loopholes that enable rapid and widespread virus propagation. Network software such as login programs and printing commands should be copied from the network master disks and checked for viruses. Once installed on the development machine, these files should be included in the list of fixed executables that are fingerprinted. Care should be taken that automatic sequences, such as login scripts, do not execute other software stored on the file server. For example, COMMAND.COM should always be loaded off a local disk, not the network. The network should be configured so that only certain users can login from development PCs. These users should not be able to read any executable files stored on the file server. (See *Secure Accessing of NetWare 3.11*, *VB*, August 1991, p. 17)

Transfer

Once software has been developed and undergone initial testing on development PCs it must be transferred to other PCs for testing. It is most important that no executable code is transferred at this stage. Viruses can only move from PC to PC with the transfer of executable code. Only source code files should be transferred from the development PC. Test PCs should have their own copies of compilers, assemblers and diagnostics copied directly from the master disks as described above. Transfer disks should be write-protected as soon as they have been written to from the development PC, so that nothing can change their contents.

Once the source files have been transferred, the executables should be generated on the test PC. These should be compared to those generated on development PCs, preferably using a cryptographic checksumming program.

Any changes (bug fixes, patches, or other improvements or corrections) that must be made to the software at this stage should be done on the development PCs, followed by a repeat of the transfer process until the software is fully functional. At this stage the original transfer disk generated on the development PC containing only source code should be transferred to the production department.

Never transfer executables between development, test and production areas!

Production

One PC should be dedicated to the task of generating duplication masters. This PC should be physically isolated from the development and test areas. It must not be connected to a network and should have some form of access control, similar to that prescribed for development PCs.

This PC is only used for the generation of software masters and is never used for any other purpose. Physical access should be rigorously enforced and limited to as few people as possible. This PC generates the final executable files that will be distributed and it **must** be free of viruses.

The only executables stored on the production PC should be compilers, assemblers and any other software needed to generate the final product. As before, these should be thoroughly tested for viruses and then copied directly from the write-protected master disks.

Once the required executables have been copied to the production machine, its contents should be fingerprinted using cryptographic checksumming. The checksums should be stored on a write-protected system floppy disk, which also contains the checksumming program. The production PC must only be booted from this write-protected floppy disk.

The production PC should regularly be checked for viruses using the copy of the checksumming program stored on the floppy disk. Any reports of any changes should be investigated immediately.

Once the production PC is known to be clear of viruses, the source files from the transfer disk may be copied onto the production PC and the software executables generated. These executables should be compared to those generated on the test

PC using a cryptographic checksumming package. Once it has been confirmed that they are identical, a software master disk may be generated using these executables. This master disk should be write-protected as soon as it has been generated. It can then be used for duplication purposes.

Disk Duplication

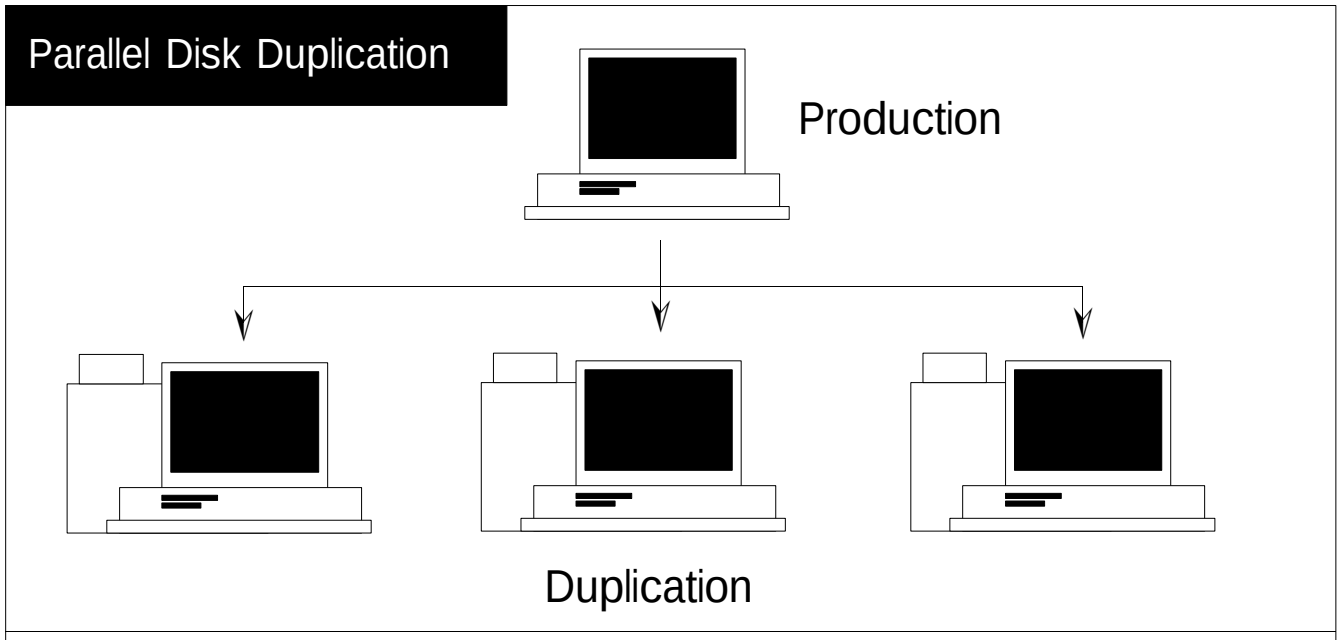
The disks finally produced by the duplication process can carry viruses for one of two reasons. Either the master disk may be infected, which can be prevented by following the guidelines already given, or the duplication equipment can itself carry an infection which spreads to the disks as they are duplicated. To prevent this second possibility, it is important to understand how disk duplicators work.

There are two main types of PC disk duplication equipment: PC-driven autoloaders and free standing duplication units. The first type is a standard PC disk drive with automatic loading and unloading of disks, attached to a conventional PC, while the second is a self-contained unit. The difference between these two types is not as clear as it might seem. Some free standing disk duplication machines essentially comprise a disk autoloader with a built-in PC containing normal processing facilities and a fixed disk (even if they do not have a keyboard or screen). This type of machine should be treated in the same way as a separate PC-and-autoloader combination. Those self-contained duplication machines which do not run software stored on disk, but only execute code stored in ROM are not susceptible to virus attack, even if a fixed disk is used to store disk images. Extreme care should be taken to distinguish these machines from those that load software from disk.

Any machine that loads its software from disk is a potential virus carrier, whether it is a normal PC or one internal to a duplicating machine. All software should be checked for viruses as described for development PCs. The executables of the machine should then be fingerprinted using cryptographic checksumming. As before, the checksums should be stored on a write-protected system disk and this disk should be stored in a secure place. The duplicating machine must always be booted from this system-disk before any duplication occurs and should only be used if it is confirmed as virus-free.

The most likely type of virus to affect the system at this stage is a boot sector virus. Duplication equipment normally operates by reading a complete floppy disk and either storing it on an internal fixed disk or holding it in memory and then writing it as a complete disk image to floppy disks. This image copying process has no concept of the files on the floppy disk and cannot infect them. However, a boot sector virus is capable of infecting the floppy disks as they are written to. It is essential that the machine is always booted from the write-protected system disk, removing the possibility of infection by a boot sector virus.

Where several duplicating machines are being used in parallel to produce identical floppy disks, either the same master disk disk should be used for all the machines, or a master for each machine should be generated on the production PC. The



Software should be loaded onto the duplication machines from the authenticated master disk. Software should never be transferred from one duplication machine to another. If duplication is to take place at more than one site, authenticated write-protected master disks should be supplied to each.

output from one machine should not be used as the master for the others. Regular samples should be taken from the output of each of the duplication machines and compared with the master generated on the production PC.

Diskettes

Software should be duplicated directly onto permanently write-protected disks, to avoid the possibility of accidental infection after the disks have been generated. This involves making a modification to the disk duplicating equipment, which is normally easily done. If possible, the PC should be booted from a disk drive other than that built into the autoloader. The 'boot' drive should have normal functional write-protection to protect the boot floppy disk.

Do not use pre-formatted disks, as these could already be virus infected. Many disk duplication systems do not write to areas of the disk which are unused on the master disk. This could conceivably result in a virus on a pre-formatted disk surviving the duplication process.

Duplication Houses

Many software producers rely on duplication houses to do their software duplication and packaging. Only reputable companies should be used and it should be made sure that they are aware of the dangers of viruses and are taking suitable precautions to prevent infection. Software producers should ensure that random samples of duplicated disks are returned for comparison with their master copy.

Programming Subcontractors

Many software developers rely on specialist programming subcontractors to develop software. There are two major dangers associated with this approach to software development. The first is the threat of accidentally infected files being supplied by the subcontractor. The second is the remote but equally unnerving possibility that malicious, undocumented features have been introduced into supplied code. Both threats can be averted by insisting that subcontractors supply code in fully documented source form.

As with all other methods of preventing virus infection the simple approach is not to allow any transfer of executable code. Subcontractors should be instructed to use the same compilers and assemblers as the main software house, so that source code can be brought in and compiled on the normal development PCs using identical compilers. All source code should be inspected prior to compilation.

'Reflections on Trusting Trust'

Ken Thompson's famous dissertation[†] described the ultimate nightmare of any software developer: a situation in which the compiler compiles source code differently to what is documented, and subsequent examination of the compiler source code fails to reveal any anomaly.

A compiler is a program, often written in its own language. (for example a *C* compiler is usually written in *C*). If a compiler generated from altered source code was distributed as a standard item, then it could add extra code (e.g. virus code) to any program compiled with it. However, compiler-introduced virus code of this sort would be unlikely to do much damage as it could be spotted by examining the compiler source code. Any anomalies become much more difficult to spot if a slightly more sophisticated approach is used. In this case a *second* modification is made to the source code of the compiler. This second modification has no function unless the compiler is being used to *compile a copy of itself*. In this case it incorporates *both* modifications into the compiler being compiled. Once this version of the compiler has been generated, it is installed to replace the original compiler.

At this stage the compiler source code can be changed back to its original form, leaving no trace that it was ever altered. However, whenever a copy of the compiler is compiled, it is recognised by the new version of the compiler and the Trojan horse code is incorporated into it. This process could have dire consequences if such a Trojanised compiler was used by a software house to generate copies of its 'in house' compiler. The software house would subsequently distribute copies of the compiler to all its customers, unaware that it had been Trojanised. Even taking a clean copy of the compiler source code and recompiling would generate a new copy of the Trojanised compiler.

The only effective way of preventing this type of Trojan horse is for producers of such software to use strong cryptographic checksumming software to ensure that changes to their compiler cannot go unnoticed. This applies to all compilers, assemblers, loaders and other such software. As we all use these tools, we are totally reliant on the manufacturers (*Microsoft*, *Borland*, *Zortech* et al.) to take appropriate precautions. It is essential that the checksumming algorithm is based on strong cryptographic principles. Anyone capable of subverting a compiler is likely to be able to reverse engineer a simple checksumming algorithm, nullifying the effectiveness of the checksums.

Mr. Thompson's rather dismal message is that you cannot trust *any* software unless you write it yourself, or inspect every single instruction of its coding. Clearly, this is not a practical proposition for the majority of business users. However, it does emphasise the need only to use software provided by reliable manufacturers and to check even this software as rigorously as possible.

[†] *Ken Thompson, Reflections on Trusting Trust, Communications of the ACM Vol 27, No 8, August 1984, Association for Computing Machinery.*

IBM PC VIRUSES (UPDATE)

Amendments and updates to the *Virus Bulletin Table of Known IBM PC Viruses* as of January 22nd 1992. Hexadecimal patterns may be used to detect the presence of the virus with a disk utility or preferably a dedicated scanner.

Type Codes

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

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

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

Seen Viruses

Albania - CN: This is a group of 4 viruses, which all contain the word Albania, but they are believed to be written in Bulgaria. The variants are 429, 506, 575 and 606 bytes long.

```
Albania          83F9 0074 0C80 7CFE 3B74 06AA E803 000E 1FC3 5651 1E06 0E1F
Albania-429      83F9 0074 0826 807D FE00 7405 41AA E80F 000E 1FBA 8000 B41A
```

Anto - CN: A small virus, only 129 bytes long, which does nothing other than replicate.

```
Anto             B800 425A 87CF CD21 B440 5A87 CFCD 21B4 3ECD 21B4 4FCD 2173
```

Beware, Monday 1st - CN: This 442 byte virus activates on the first day of the month, provided it is Monday, and then overwrites the first track of diskettes in drive A: It contains the text 'BEWARE ME - 0.01, Copr (c) DarkGraveSoft - Moscow 1990'.

```
Beware           C3B4 3ECD 21C3 8DB5 8402 57B9 3100 8BFE AC34 80AA E2FA 5FC3
```

Black Monday-Borderline - CR: This virus is detected by the Black Monday pattern, but it appears to be an older variant, as it lacks the ability to infect EXE files. It is also shorter, only 781 bytes.

Checksum - CR: Version 1.00 of this Russian virus is 1233 bytes long and version 1.01 is 1232 bytes long, with only minor differences. As the name implies the virus calculates a checksum for itself, and if changed it will not activate. The virus is designed to replace older versions of itself.

```
Checksum         832E 0300 4F83 2E02 004F 0BC9 740B 508C C040 8EC0 B449 CD21
```

Crazy Imp - CR: A 1445 byte virus, which is almost fully stealth. It was received from Minsk. It uses several tricks to hide from debuggers but has no effects other than replication.

```
Crazy Imp        B413 CD2F 33C0 8ED8 832E 1304 048C C88E D848 8EC0 2681 2E03
```

CSL-V4 - CR: A 517 byte variant of the CSL (or Microelephant) virus reported in the December edition and probably written by the same author. Not yet analysed. The CSL-V5 is another new variant of the same virus, but it is only 457 bytes long.

```
CSL-V4           5152 1E06 8BF0 0590 008B D88C C88E D8BF 0001 8B47 FD89 058A
CSL-V5           5152 1E06 8BF0 0592 008B D88C C88E D8BF 0001 8B47 FC89 058A
```

Dada - ER: A Russian virus which contains the text 'da,da' - Russian for 'yes, yes'. Awaiting analysis.

```
Dada             CB50 8CC0 2603 0603 0040 8EC0 58C3 33C0 8EC0 2680 3E00 004D
```

Diskspoiler, 1308 - CN: A 1308 byte Russian virus, which uses very simple encryption. The virus searches the FAT for free clusters and marks them as bad, slowly eating up the entire disk.

```
Diskspoiler      E800 005E 8BFE B90B 0580 750E FF90 47E2
```

DM-310 - CR: Probably an older and more primitive version of the DM-400 virus. It does not seem to do anything but replicate.

```
DM-310           F7C1 FEFF 7405 B801 43CD 63C3 E800 005D 061E 33C0 8EC0 2680
```

DM-400 (1.01) - CR: A slightly improved version of the DM-400 virus reported earlier, with extra encryption added. It too is 400 bytes long. The virus corrupts files that fit the *.TP? pattern - overwriting the first 8 bytes.

```
DM-400 1.01      56B9 2401 3024 46E2 FB5E C3E8 0100 CF5D 0633 C08E C0BB 0600
```

Feist - CER: A 670 byte Russian virus, awaiting analysis.

```
Feist            B10C D3E2 5233 D2B9 1000 F7F1 8BCA 5A03 D02B 16A6 0383 EA10
```

Hafenstrasse - EN: An 809 byte virus, probably from Germany. Awaiting analysis..

```
Hafenstrasse    F607 FF74 1E8A 170A D274 0743 B402 CD21 EBF3 B20D B402 CD21
```

Hungarian-473 - CR: Closely related to the Hungarian-482 virus, this 372 byte virus activates on June 13th and then overwrites the Master Boot Sector of the hard disk. Detected by the Hungarian-482 pattern.

Hydra - CN: A group of nine viruses, which do nothing particularly interesting. The set of signatures below can easily be reduced to one or two by using wildcards.

```
Hydra-0         B43D B002 BA53 01B0 02CD 218B D806 1FB8 003F B9FF FFBA E002
Hydra-1         B43D B002 BA53 01B0 02CD 218B D806 1FB8 003F B9FF FFBA 9301
Hydra-2         B43D B002 BA53 01B0 02CD 218B D806 1FB8 003F B9FF FFBA 5701
Hydra-3         B43D B002 BA53 01B0 02CD 218B D806 1FB8 003F B9FF FFBA 5601
Hydra-4         B43D B002 BA53 01B0 02CD 218B D806 1FB8 003F B9FF FFBA 5401
Hydra-5         B43D B002 BA53 01B0 02CD 218B D806 1FB8 003F B9FF FFBA 8701
Hydra-6         B43D B002 BA53 01CD 218B D806 1FB8 003F B9FF FFBA 7401 CD21
Hydra-7         B43D B002 BA53 01CD 218B D806 1FB8 003F B9FF FFBA 7001 CD21
Hydra-8         B43D B002 BA53 01CD 218B D806 1FB8 003F B9FF FFBA EF01 CD21
```

JD - CR: A group of four semi-stealth viruses, 356, 392, 448 and 460 bytes long. In addition there are two shorter variants, 158 and 276 bytes, with no stealth features. Not fully analysed, but do not appear to do anything but replicate.

```
JD (1)          521E B813 35CD 2106 5304 11CD 2106 53B8 2425 501E 520E 1FBA
JD (2)          5053 561E 068B F2B4 2FCD 21AC 3774 0383 C307 061F 8B47 1724
JD-158          5ABB 4300 8EDB 833D 3D74 08B4 25CD 21B1 9E8E C30E 1FF3 A458
```

Keyboard Bug - CER: This virus was received from Kiev, but has not yet been fully analysed. Analysis is complicated by the fact that the virus uses multiple layers of encryption, as well as other methods to hide from debuggers. The effects are unknown, but are assumed to be keyboard-related. The length has been reported as 1720, but the actual increase in length is variable.

```
Keyboard Bug    1E53 2EFF B597 07BB 6E06 B928 0158 2E30 0143 E2FA 5B1F E8
```

MSTU-554 - CEN: Closely related to the 532 byte variant reported last December.

```
MSTU-554        BB16 0026 8B07 3DEB 55C3 5E8B C6B1 04D3 E80E 5B83 C364 03D8
```

Murphy-Amilia - CER: This Canadian virus is based on the HIV variant, and is only slightly modified. It is 1614 bytes long, and detected by the HIV pattern.

Orion - CR: Two simple viruses, probably from Bulgaria. They contain the texts ‘Hello,boy! Im a new virus’ and ‘Orion system !’. The viruses, which are 262 and 365 bytes long contain one error - they cannot properly infect very short files.

```
Orion-262       AB33 C0AB 1616 1F07 8BC3 CB3D 004B 7406 E8A2 FFCA 0200 1E06
Orion-365       AB33 C0AB 1616 1F07 8BC3 CB3D 004B 7406 E89F FFCA 0200 1E06
```

Phalcon, Cloud - CN: A 1117 byte virus, awaiting analysis. It contains a strange text message about someone called Bob Ross.

```
Phalcon         BE15 0103 3606 018A 24B9 2304 83C6 2D90 8BFE AC32 C4AA E2FA
```

Pixel-Rosen - CN: The smallest member of the Pixel family, only 131 bytes long. Does nothing but replicate.

```
Rosen           A433 FF06 57CB 1E07 BE83 01BF 0001 1E57 B9FF FE2B CEF3 A4CB
```

Shirley - ER: A 4096 byte virus, probably from Germany, which contains several long text messages, including the string ‘IWANTSHIRLEY’. Awaiting analysis.

```
Shirley         4683 C704 8BC6 3B06 B019 7307 8B05 3946 0475 ED8B C63B 06B0
```

Sistor - CER: Two viruses from the USSR. The 2225 byte variant triggers after 16:00, displaying a familiar bouncing-ball/falling letters effect. The later variant has been improved somewhat - it is not as obvious, and includes code to bypass interrupt monitoring programs.

```
Sistor-2225     5BFA 891E 7000 8C06 7200 FB33 C08E D8B8 4953 A340 032E 80BC
Sistor-2380     5B33 C089 1E70 008C 0672 0033 C08E D8B8 4953 A340 032E 80BC
```

Smallv-115 - CN: A very small virus from Bulgaria. Does nothing of interest.

```
Smallv-115      B802 3DCD 218B D8B9 0300 8BD5 B43F CD21 B802 4299 8BCA CD21
```

Surrender, Jews - CER: A 513 byte Russian virus, containing the text ‘Jews never surrender!’. Awaiting analysis.

```
Surrender       061F B800 43CC 51B8 0143 33C9 CCB8 023D CC0E 1F8B D8B4 3FB1
```

Timeslice, 2330 - CER: A 2330 byte virus, written in the former USSR. It does not appear to do anything but replicate, but the infection mechanism is rather unusual, as the virus intercepts INT 28H and it therefore infects at irregular intervals.

```
Timeslice       1E8E C64E 8EDE C745 0108 0009 C975 0581 6D12 C300 1F8B F5BF
```

Tiny DI - CN: Four new variants of the family which was previously called Mutant. The viruses are 94, 101, 108 and 110 bytes long and do nothing but replicate. Only the 110 byte variant works correctly - the shorter variants are not able to infect most files correctly, but simply destroy them.

```
Tiny DI-94      B802 3DCD 218B D806 1F8B D749 B43F CD21 055E 0050 33C9 B800
Tiny DI-101     B802 3DCD 218B D806 1F8B D749 B43F CD21 0565 0050 33C9 B800
Tiny DI-108     B802 3DCD 218B D806 1F8B D749 B43F CD21 056C 0050 33C9 B800
Tiny DI-110     B802 3DCD 218B D806 1F8B D733 C949 B43F CD21 056E 0050 41B8
```

Tiny-Ghost - CR: This virus differs from the other members of the Tiny family in two ways. It is fairly long, 330 bytes, and it has one effect other than replicating - it will display the message 'This scan program can't find me I'm a GHOST in your machine!!', if it detects the execution of a virus scanner.

```
Ghost           9191 2687 85E0 FEAB E3F7 931E 07C3 3D00 4B74 052E FF2E A401
```

Trivial-44 - CN: Yet another uninteresting overwriting virus from Bulgaria.

```
Trivial-44      023D CD21 8BD8 B92C 00BA 0001 B440 CD21 B43E CD21 B44F EBE0
```

Tula-419 - CER: Probably a Russian virus. It is 419 bytes long and will only infect on machines with a colour display.

```
Tula-419        B43F CD21 7225 BEA0 0FAC 3C4D 7505 AC3C 5A74 5CBE 6E02 81C6
```

Tumen 1.2 - CR: A 1225 byte member of the Tumen family. Detected by the pattern previously published for the other two variants.

Vienna-625 - CN: A minor variant of Vienna. Detected by the Vienna-4 pattern.

Vienna-Kuzmitch - CN: An encrypted, variable-length variant of the Vienna virus, which contains a block of text in Russian. The base length of the virus is 810 bytes. No simple search pattern is possible, only a short one, which contains several wildcards. Second-generation copies of this virus do not always seem able to replicate.

```
Kuzmitch        BE?? ??FC 33DB B9?? ??8A 54?? 3090 ???? 43E2 F9
```

Vindicator - CR: A 734 byte virus, which is found at the beginning of infected files. Probably of Russian origin. Awaiting analysis.

```
Vindicator      FAB8 0010 F6E7 0500 B88E D831 F6B8 2000 BAA0 8031 DB38 E372
```

Voronezh-Chemist-650 - CR: A 650 byte member of the Voronezh family, reported to have originated at the Moscow State University. It contains a text string in Russian which translates to 'The Chemist & the Elephant'. The virus activates if an infected program is run at xx:03 o'clock when it displays the message 'Video mode 80x25 not supported.' and switches to 40 column mode if possible.

```
Chemist-650     0500 018B F0BF 0001 FC8A 0434 CC88 0546 47E2 F6B8 0001 50B4
```

Wordswap-1391, Wordswap-1485 - CER: Just as in the case of the 1387 and 1503 byte variants reported last November, no search pattern is possible for these two variants.

Zherkov-1882 - CER: A 1882 byte version of the Zherkov (formerly Lozinsky) virus. It uses a slightly more sophisticated encryption algorithm than the older variants, and is able to infect EXE files. The 1958, 2968 and 2970 byte variants are probably later versions. All the viruses are targeted against the AIDSTEST program, a Russian anti-virus program written by D. Lozinzky, deleting it if it is executed. The virus also attempts to corrupt data on diskettes in a unique way - it sets the byte at location 1AH in the boot sector (Number of sides) to zero - causing the DIR command to produce a 'Division by zero' error. The larger viruses have slightly different effects - the 2968 and 2970 byte variants display a striking screen display with the text 'AIDSTEST' if no key is typed for 30 seconds, and then restore the screen on the next keystroke.

```
Zherkov-1882    5051 061E E800 005E 2E8A 44F8 3C00 740F 83C6 1890 B9D9 062E
Zherkov-1915    5006 1EE8 0000 5E2E 8A44 F93C 0074 118B FE83 C71A 90B9 0007
Zherkov-2968    5706 1EE8 0000 5E2E 8A44 F53C 0074 118B FE83 C71A 90B9 EE0A
Zherkov-2970    5006 1EE8 0000 5E2E 8A44 F93C 0074 118B FE83 C71A 90B9 F40A
```

Reported Only

191 - CN: Small, 191 bytes, does nothing but replicate

1241 - CR:

Barcelona - CR: 1792 bytes - contains text in Spanish.

Beta - CN: 1117 bytes.

Involuntary - ER: Infects executable SYS files as well.

Tormentor - ER: 1024 bytes. Reported to delete *.PAS files.

HARDWARE

Dr. Keith Jackson

Chips With Everything?

Although most products which claim to detect viruses have traditionally been sold as software packages, in recent months several companies have begun to sell hardware anti-virus products. Some of these products have been reviewed in *VB* (the first review appeared in the June 1991 issue), in the main with a none too favourable outcome. This article attempts to explain the advantages and disadvantages of using hardware products to fend off viruses, and to shed some light on the main reasons why the *VB* reviewing process has found problems with such products. In many ways hardware-based anti-virus products are only reverting to methods which have been used for several years in other areas of PC security. They also seem to be following a similar evolutionary path (see below).

Advantages

A hardware-based anti-virus product can provide all of the facilities of a software anti-virus product, plus:

1. The software controlling the anti-virus product resides in a chip (usually an EPROM), and cannot be altered by an executing program. In particular there is no possibility of the anti-virus software becoming infected by a virus.

2. The anti-virus hardware can take control of the PC before the MS-DOS operating system becomes active and can provide access control facilities.

3. Booting from a floppy disk can be constrained and/or prohibited as required. Without rewriting the PC BIOS, prevention of booting from a floppy disk requires hardware and cannot be achieved by software alone.

4. Any desired tests (e.g. inspection of a hard disk) can be performed before MS-DOS becomes active, therefore it is unlikely that the 'stealth' features used by some viruses will prevent detection of the virus.

Disadvantages

The disadvantages of implementing anti-virus products in hardware are outlined below, and a user has to balance these against the perceived advantages:

1. Most anti-virus hardware is implemented as a plug-in card which has to be inserted into an available slot within the PC. However, in recent years an increasing percentage of PCs have been sold in laptop or notebook form, which do not have slots for plug-in cards.

2. Anti-virus hardware must interact with the PC hardware at a very low level and it is inevitable that many types of PC will exhibit problems.

3. If the anti-virus hardware does not incorporate its own processor, then it can only function by 'stealing' some execution time from the PC's processor. This is most visible as a decrease in system performance and an increase in boot time.

4. Without an on-board processor, the software controlling the anti-virus hardware can only function by 'hooking' into one or more of the software interrupts available on the PC. This is neither the time nor the place to explain how interrupts work on a PC using the MS-DOS operating system, suffice it to say a program can be removed from the interrupt chain so that it can never execute (see below).

5. An anti-virus hardware product with its own on-board processor will inevitably be quite expensive, and it is doubtful that such costs can be justified simply on the grounds of fighting viruses.

6. Virus-specific components are hard to upgrade regularly if search data is stored in PROM!

Operating a PC

It is an absolute necessity that hardware is added to a PC if booting from a floppy disk is to be prevented under all circumstances. The software based security products which offer such a feature do not prevent a user booting from floppy disk, they merely encrypt all the data on the hard disk.

Preventing a user booting from a floppy disk is a major defence when trying to prevent the spread of viruses, as floppy disks are one of the main means by which viruses spread from one PC to another. This feature is usually offered by manufacturers in combination with access control. Using both of these features it can therefore be established not only that a PC was booted in a particular manner, but also who used the PC at any particular time; both are important pieces of information when tracing the source of a virus infection.

Overhead

The overhead introduced by a memory-resident anti-virus product is usually simple to measure. Perform some tasks using a large amount of processing time (e.g. a spreadsheet recalculation), and a similar set of tasks using a large amount of file manipulation (e.g. copy many large files). Time how long these tasks take to perform. Repeat the tasks with the anti-virus product installed, and compare the two times.

Whenever I have done this as part of the *VB* review process, manufacturers of anti-virus products (hardware or software based) have without fail objected to my results and have often produced page upon page of their own timing tests to 'prove' that my results must be wrong. Ignore such shenanigans.

The only timings that matter are the ones on the PC that you intend to use. Insist on measuring the overhead introduced by the anti-virus product on a particular PC. All other timing tests produced by the manufacturer are irrelevant, and will no doubt be designed to show the product in a good light.

Possible Problems

Without exception, all of the anti-virus hardware products reviewed in recent months for *VB*, have caused me to have grave reservations about their design. I believe that to provide a level of security that cannot easily be circumvented, it is imperative that a plug-in anti-virus hardware product has its own processor on the card. If you are not going to do this then the advantages of using hardware seem at best minimal. Most anti-virus hardware products seem to have only an EPROM chip, a RAM chip, and some logic chips to 'glue' things together. They function by ensuring that one or more of the MS-DOS interrupt vectors are set so that software contained within the EPROM is activated at the correct time.

"No matter what manufacturers may claim, it makes no difference that some of the software is stored in EPROM on a plug-in card. The interrupt vectors can still be reset."

However, on PCs any program can have access to any part of the computer's memory. Therefore 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 program. To prevent the anti-virus hardware product from having any effect, a virus can simply re-vector the interrupts as desired.

No matter what manufacturers may claim, it makes no difference that some of the software is stored in EPROM on a plug-in card. The interrupt vectors can still be reset.

The developers of some of the anti-virus hardware products available claim that they ensure that their software must be active by changing the operation of the PC's BIOS so that it can only function properly using information residing on the card. Typically encryption is used to enforce this constraint. This worries me immensely, as altering the way that a PC operates at such a low level in the software hierarchy is bound to be fraught with compatibility problems when a product is used on PCs made by many different manufacturers. With the best will in the world, developers can only test products on a few different PCs. There is no guarantee that reviewers or *users* will own one of the PCs that have been tested with the particular product.

A processor which executes its own program on a plug-in card can ensure that software executing on the PC is not permitted to access the security features contained on the plug-in card. This partitioning of the software is a major advantage, and the plug-in card can therefore monitor the operation of the PC with no fear that events on the PC can interfere with this monitoring. For instance, in the case of viruses a plug-in card could monitor the PC bus to look for viral activity without the ongoing worry that a 'stealth' virus was concealing itself.

Varying Cost

Cost is the real reason why most hardware security products do not use their own on-board processor. The design costs, the manufacturing costs, the component costs, and the extra software development costs combine to ensure that the price of such products remains high.

As with all things in life, *'you pays your money and you takes your choice'*. However, I would introduce a note of caution. The development of hardware specifically tailored to detecting viruses is following the same path that general PC security products followed some four or five years ago. Many lessons were learnt then, and as the manufacturers of anti-virus security hardware have so far tended to be different from the manufacturers of more general purpose security products, these lessons are now being relearned. I would caution against purchase of anti-virus hardware until the developers have moved somewhat further along this learning curve. Perhaps a few months delay will be sufficient.

Conclusions

I would like to believe that the manufacturers of anti-virus products are in business to prevent the spread of viruses. However, I must be cynical and point out that they are actually in business to maximise their profits by selling anti-virus products. This goal is often best achieved by increasing the price of each individual unit. Therefore, security products with increased functionality tend to be the more expensive products - this is beginning to happen with anti-virus products. This process of stratification will inevitably continue as it merely reflects the differing value of the information contained on various PCs.

Hardware anti-virus products can offer access control features, and can prevent a PC from being booted from a floppy disk. These are excellent security features, but they are not virus-specific. I believe that virus-specific features will be reflected back into general purpose security products (indeed this process is already afoot). If you require security at the highest level(s), then given the architecture of the PC, additional hardware is inevitable, and it is likely that you will require more security features than mere anti-virus features.

DIRTY MACS

David Ferbrache

1991 - A Quiet Year For The Mac

In stark contrast to the seemingly exponential explosion of virus strains and variants on the IBM PC platform, 1991 has been an exceptionally quiet year on the *Apple Macintosh* platform. The year saw one new virus for Hypercard (dubbed the 3-tunes virus), three strains of existing viruses (MDEF D, ZUC C and nVIR C) and a further nVIR clone (nCAM). A number of issues relating to the security of the new *System 7* release from *Apple* have also arisen, in particular with regard to network file sharing facilities and the restructured desktop.

MDEF D

The year started with a new reported strain of the MDEF virus. This virus belongs to the family of MDEF viruses discovered in Ithaca, New York and includes MDEF A (May 1990, Garfield), MDEF B (August 1990, Top Cat) and MDEF C (October 1990). The MDEF D strain infects applications by adding a viral MDEF resource:

```
MDEF id=8375 size=506
```

The application's MENU resource is modified to point to the viral MDEF resource rather than the standard system MDEF resource (or application dependent MDEF resource). When the viral MDEF resource is activated by an infected application it will search the current directory for an uninfected application (file type = APPL). The current directory is defined as the most recently referenced directory in the standard file selection dialog. A single application will be infected on each call of the viral MDEF resource. The system file is not modified or infected.

Disinfection of applications modified by MDEF D is normally possible (by patching the application MENU resource). The virus does not, however, store the original MDEF reference from the MENU resource, and thus applications using resources other than MDEF 0 (custom menu definitions) may have to be corrected after disinfection.

MDEF D is detected by *Disinfectant* 2.5 while *Virus Detective* can be modified to detect the virus by adding the string:

```
File type=APPL & Resource MDEF & WData 4546#8A9AB
#84E55
```

ZUC C

The C strain of the ZUC virus was detected in June 1991 in Italy. The virus infects applications under all system releases later than 4.1. The virus appends itself to the first CODE resource in the application jump table.

Two strategies are adopted by the virus to select an application for infection:

1) With a probability of 15 in 16, the virus will search the old *System 6* desktop file for all resources of type 'APPL'. The first uninfected application with such a resource entry is infected.

2) Otherwise the virus will begin a full recursive search of the hierarchical directory structure on all mounted volumes for an uninfected application.

One consequence of this search strategy is that the virus takes a linearly increasing time to locate an application to be infected, thus on heavily infected systems a major reduction in system performance may be detected.

The virus will infect applications whose code segment is not numbered CODE 1, unlike ZUC A and B which are restricted to being able to infect only applications whose main code segment is numbered CODE 1. Non-standard jump table entries will also prevent ZUC A and B infection (such as *Think PASCAL*), ZUC C will infect such applications.

The virus will trigger if the system clock is set to a date/time after 13 August 1990 13:13:13, and the application has been infected for between 13 and 26 days. On triggering the virus will insert a vertical blanking queue (VBL) task. VBL tasks will be triggered during the vertical retrace period when the monitor is returning to top of screen. The VBL task will cause the cursor to begin to bounce whenever the mouse key is depressed. This behaviour will begin one minute after the virus has activated.

Spread will only occur on HFS volumes. A number of executables are excluded from being infected, including those applications with creator signature:

SpDo, XPRS, DFCT, VGDt, VlRy, OMEG, FEVr, PLUS, VlCM

These signatures include a number of commercial anti-virus products, presumably to avoid detection by the self-checksum routines incorporated in such products.

The virus will also bypass software locks on volumes, read only and protected resource flags. Code has also been included (as in ZUC A and ZUC B) to attempt bypass of memory-resident anti-virus monitor software.

This strain can be detected by *Disinfectant* 2.5, *Virus Detective* 4.0.4. Older versions of *Virus Detective* can be modified to detect the virus by adding the search string:

```
Filetype = APPL & Resource Start & WData
A746*A033*A042*A9F7
```

Visual inspection of an infected file will show the main code resource to have 4 bytes at the end containing CO $BA $BB (the last two bytes being hex values). This is a trivial encryption of the original ZUC 'CODE' signature.

nVIR C

In July 1991 a new strain of nVIR was discovered in the USA. This strain infects both applications and the system file. The virus adds the following resources to the files:

System File	Application	Common to both
INIT 32 416b	CODE 256 788b	nVIR 1 428b
nVIR 0 2b	nVIR 2 8b	nVIR 6 66b
nVIR 4 788b	nVIR 3 416b	nVIR 7 2106b
nVIR 5 8b		

The virus is similar to the standard nVIR strains in that it infects the system file by adding an INIT 32 resource (a copy of the nVIR 3 resource from an infected application). On reboot the INIT 32 will cause the TEinit (text edit package initialisation) trap to be patched to point to the memor-resident virus code.

Applications which make use of this package (most non-trivial applications) will be infected. The virus includes an nVIR 0 generation counter in the system file which is set to 1000, and decremented by 1 on reboot, 2 on the launch of an infected application. When the counter reaches 0 the virus will begin to 'beep' on 1 in 8 reboots or 1 in 4 application launches.

The virus can be recognised by its distinctive auxiliary code (nVIR) resources, or by a CODE 0 jump table entry of the form '0000 3F3C 0100 A9F0'.

nCAM

In true tradition a further nVIR clone has appeared in which each auxilliary resource has been named 'nCAM' rather than 'nVIR'. It is expected that many other such resource editor products will appear. This strain is detected by all major anti-viral products (which now employ generic nVIR clone detection code).

3-Tunes

Finally, in March 1991 a new hypercard virus was detected in the Benelux. This virus named '3-Tunes' (also known as the HC virus) follows in the footsteps of the Dukakis virus reported in 1989. The 3-tunes virus is 3359 bytes in length an is designed to infect hypertalk stacks.

When a stack infected by 3-Tunes is loaded, the virus will be activated. Closing the infected stack will then infect any uninfected home stacks which are open at the time. Replication of the virus will continue until 10th November 1991 (when a date is specified in European format dd.mm.yyyy. In systems using other date formats (e.g. US mm\dd\yy) the virus will not stop replicating.

Damage caused by the virus include five separate effects:

1) After 17 seconds the message 'Hey, what are you doing?' is displayed, and a sound generated.
2) After 2 minutes a German folk song '*Muss I Denn*' is played and repeated at 4 minute intervals.
3) After 4 minutes the song 'From the Blue Mountains' is played and repeated at 4 minute intervals.
4) After 5 minutes the pop-up menus for toolbox and pattern will be displayed and, if closed, they will reopen each minute.
5) After 15 minutes the text 'Don't panic' scrolls across the screen accompanied by a sound effect.

Damage is triggered when the trigger condition

> Date character 1-2 < '11' OR
> Date character 4-5 < '11' OR
> Date character 9-10 < '91'

is False. The *University of Hamburg Virus Test Center* has detected system hangups using an infected hypertalk stack which are presumably attributable to heap/stack overflow.

General Security Issues

The new *System 7* operating system release includes a variety of changes and new features. The *Computer Incident Advisory Capability* (*CIAC*) at the *US Department of Energy* has issued a warning bulletin concerning a number of issues. Of particular concern is the ability to provide remote access to the local file system via file sharing. The default file sharing setup permits global access to all files on the local host. It is therefore crucial that the 'Users & Groups' control panel device be used to modify guest access. The access rights for the <Guest> user should be carefully setup to restrict access to public files. Additional network users can be created with specified passwords with appropriate access rights. Care should also be taken to control volume access using the 'Sharing...' choice from the file menu.

A number of significant security issues are raised from the provision of world writable folders. No folder should be made world writable under any circumstances.

The desktop file in *System 7* has been significantly restructured to include a new desktop database. The original flat file may still exist on many *System 6* disks. Thus the virus strains (such as ZUC) which search the desktop file as a means of locating uninfected applications will be limited in the range of applications infected, although in the case of ZUC the exhaustive directory search will still occur. *System 7* is also immune to infection by the desktop viruses (CDEF and WDEF) due to restructuring of the desktop file.

PC VIRUS ANALYSES

Jim Bates

1. Hallöchen

The Hallöchen virus is one of the 'also rans' in the ongoing list of viruses currently in the wild in Europe. First reported from several sites in Germany during late 1989, occasional reports of it are still received from time to time. However, accurate information about what the virus actually does and how it works has been difficult to find, so a full disassembly has now been completed and the analysis is presented here. 'Hallöchen' is a twee German idiom meaning 'hello'.

Structure

This is a resident, parasitic virus which infects COM and EXE files as they are invoked through the DOS Load and Execute function request (4BH). The virus contains a keypress corruption routine as well as a time-wasting routine, both of which are installed on a progressive basis dependent upon the number of a current infection count. The code is unremarkable but provides yet more insights into the basic design flaws so characteristic of virus writers. In this instance, the writer has left portions of code lying around which contribute nothing to the function of the virus but do indicate that various ideas were tried and discarded.

Installation

This virus attaches its code at the end of files and modifies either the initial instructions or the file header parameters to ensure that the virus code is executed first.

By its very nature, virus code needs to be mobile and relocatable. A problem associated with this approach is how the subsequent code can determine exactly where it (and its associated data) are located in memory once loaded for execution. In this case the first routine completes various calculations to ensure that the code is addressed via a zero offset. Then the DOS Memory Control areas are searched to see whether the virus is already resident. This is different from the more usual 'are you there?' calls which some viruses use, but the method is nonetheless effective. The test consists of a simple search for a value of 5555H at offset 167H of the relevant memory block.

If the virus is found in memory, a check is made of the resident infection counter and if this is less than the counter within the current file, it is updated before processing reverts to the original program. Otherwise the memory blocks are manipulated to place the virus code permanently into memory so as not leave an obvious 'hole' or significant reduction of available memory.

Monitor Subversion?

Once installed, the code is hooked into the system but once again the method is the less obvious one of overwriting part of the interrupt service routine. This is similar to the system used within the 4K/Frodo virus and is probably a primitive attempt to avoid simple anti-virus software which monitors changes to the Interrupt Table. The process involves copying the early part of the service routine into a data area and replacing it with a jump into the virus code. This necessitates cumbersome routines to swap the two sets of information back and forth at appropriate times. When first installed, only the DOS interrupt is hooked and only function 4BH (Load and Execute) is intercepted.

File Infection

Each request for function 4BH is held while the target file is checked. The first check attempts to determine whether the target file is located on a disk in drives A: or B: (usually floppy disks). If so, the resident infection counter is reset to zero before continuing. The checking sequence is unusual in that it continues by unhooking the virus code from the system before going on to open the target file for inspection.

Once the file has been opened, its last date of access is checked and if the month and year match that currently set within the system, the file is closed and processing continues with the original system request the infection. Otherwise, the first two bytes of the file are checked to see whether the common 'MZ' signature of a standard EXE file is there.

Processing branches at this point depending upon the results of the check but each branch immediately tests the file to see whether it is already infected. Within EXE type files, the checksum field (offset 12H) in the header will contain a value of 5555H on infected files, while non-EXE types will have the same marker at offset 4 (5th and 6th bytes) in the file. If the target file is not infected, a further check is made on the file size. As a result of incorrect coding, EXE type files are infected regardless of their size but other types are only infected if they are longer than 4999 bytes and more than 2026 bytes below a 64 Kbyte boundary (63509 bytes, 129045 bytes, etc.).

Once accepted for infection, the file length is padded to expand it to a 16 byte boundary before the 2011 bytes of virus code are added. Thus the actual length by which infected files will increase varies between 2011 and 2026 bytes. Since this virus has no stealth capability, increases in file size will be readily apparent within an ordinary directory listing.

Trigger Routines

As each file is infected, a counter is incremented and this value is used within the virus code to trigger various undesirable effects. The first of these, encountered when the counter passes fifty infections, installs an additional routine into the INT 08H service. This service is the Timer Tick routine and is usually executed around 18 times per second during normal machine operation.

As the virus installs this, a subsidiary counter value is calculated which introduces a specific delay into the routine. Thereafter, as each new file is infected, the INT 08H routine is de-installed and then re-installed with an increased value in the delay. The net result is that the machine becomes slower and slower as the infection count increases.

Also keyed to the infection counter is an equally insidious routine which the virus hooks into the system keyboard services as the infection counter passes seventy. In this case a similar progressive effect is applied to a simple corruption routine which adds a value of 1 at random to single keystrokes. Thus 1 may become 2, A may become B and so on. The progressive effect is applied to the chances of this happening so that after 72 infections, there will be a 1:256 chance of corruption, after 74 - a 2:256 chance, after 76 - a 3:256 chance and so on.

Also contained within the virus code (at offset 101H) are the plain text messages:

```
Hallöchen!!!!!!, Here I'm
Acrivate Level 1
```

The 'ö' is ASCII character code 148 decimal. The virus contains code to display messages on screen but it is not in the processing path and the messages are not displayed.

Detection and Disinfection

A reliable recognition string for this virus was published in *VB* in July 1991, but note that the associated report erroneously indicated that infection only occurred on files matching the current month and year. A reliable search pattern is repeated here:

```
Hallöchen       EB8C C903 D98E D3BC DB08 53BB
                2E00 53CB
```

Infected files should be deleted (using the DOS DEL command) and replaced from clean masters after a clean system reboot. For maximum security infected files should be overwritten before being deleted (see below.)

2. SPANZ

Another virus which appears occasionally in the wild is SPANZ. This is a primitive, non-resident virus which attacks only COM files (including COMMAND.COM). It was first reported in Europe in late 1991 and is not particularly widespread.

Operation

The virus code is extremely simple. When an infected program is executed, the virus collects the system date and calculates the number of months elapsed since January 1980. This number is used later as a pseudo-random indicator to determine whether a file deletion routine is invoked. A search commences of the current directory for uninfected COM files. The virus' self-recognition signature is the last three bits of the seconds field, which are set to zero, in an infected file's directory entry.

If no suitable files are found, the search continues by examining the PATH setting within the machine environment looking for COM files within each specified directory in turn. Once a suitable file is found, the first three bytes of the file are overwritten with a calculated jump instruction, and 627 bytes of virus code are appended to the end of the file. The file seconds field is then altered as noted above before processing returns to the original program code.

Unusually, this virus makes no attempt to alter file attributes and so if a file is set to Read Only, the virus will ignore it and continue the search. A check is made on file size and anything over 63488 bytes in length is ignored.

This is effectively a 'one shot' virus in that each time the code is executed, one more file will be infected. The file deletion routine referred to above is only invoked when the virus can find no suitable files to infect. When tests were conducted, the routine did not cause file deletion or corruption but differing environments might produce different results. Since this virus contains virtually no error handling routines, there is a risk of the machine hanging during virus execution.

Detection

No stealth or encryption methods feature in the code and as a result this virus is easy to detect. A reliable search pattern is published here:

```
SPANZ   807D 043D 7506 83C7 051F EB0F B9FF
        7F33 COF2 AE80 3D00 75DB
```

A plain text message:

```
INFECTED! * SPANZ *
```

may be also be seen as the final 20 bytes of an infected file.

Removal

Simple deletion of infected files and replacement with clean master copies is the safest method to remove this virus. In order to prevent someone from intentionally 'resuscitating' the deleted virus code it is advisable to overwrite infected files (using a positive erasure utility) prior to deleting them.

Infected files may be reparable by some generic cure programs. A number of virus-specific scanner/disinfector programs can remove this virus - both *McAfee's CLEAN* and *Frisk's F-PROT* contain disinfection routines. As always, tests should be conducted with any disinfection program to establish that files are disinfected correctly.

PRODUCT REVIEW

Mark Hamilton

Untouchable

Fifth Generation Systems released its anti-virus product last month and, in common with *Xtree* (see *VB*, January 1992, pp. 19-23), the company has contracted with an Israeli anti-virus software developer. *Fifth Generation Systems* is marketing *BRM's V-ANALYST* but in a modified and updated form. [*The original review of V-ANALYST appeared in VB, October 1990, pp. 15-17 and readers are directed to this review as a source of supplementary information. Ed.*]

Fifth Generation Systems regards *Untouchable* as a logical extension of its product range of data security products. The company has stated that it would only enter the anti-virus market once it could 'market a product that would not become obsolete as soon as it hit the streets'. The company thus emphasises the generic detection and disinfection capabilities of the product.

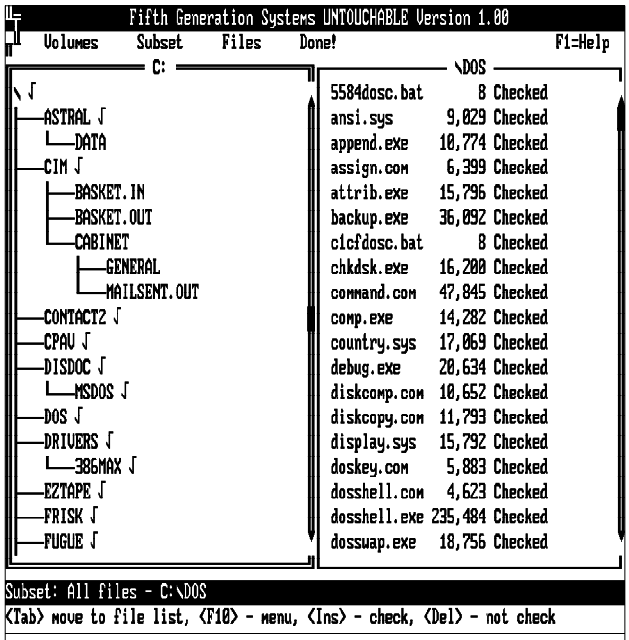
Components

Untouchable consists of a slim manual, a quick reference card and software on both 5.25- and 3.5-inch diskettes. The larger diskette was permanently write-protected, but the 3.5-inch diskette was write-enabled. An installation program is provided; its use is necessary to install *UT*, the package's generic checker. Both the TSR monitor (*UTRES*) and the virus scanner program (*UTSCAN*) can be run uninstalled. Strangely, the distribution disk contains a file named UNVIRUS.DOC which reads 'This is a text file used by unvirus (sic) for more information.' *UNVIRUS*, which is not a component of *Untouchable*, was a previous virus scanning and disinfection program developed by *BRM Systems*. According to the manufacturer, this anomaly in the documentation was corrected by the time that *Untouchable Network*, the fully network compatible version was released.

Generic Defence

Fifth Generation Systems has adopted the policy that virus-specific protection is short-lived and the real answer lies in generic checking. The company's stance is adequately laid out on page 9 of its manual, in a section entitled 'Unique product features': 'No need for updates ... since *Untouchable* uses reliable file signature information as a way to thwart future, unknown viruses, it does not burden you with frequent virus signature updates.' Generic checking (which *Fifth Generation Systems* call 'differential detection') is not a new technology - until recently *Fifth Generation Systems* marketed an established integrity checking program called *Mace Vaccine* (see *VB*, December 1989, pp. 13-14).

The company does not state which checksumming algorithms are employed, but simply states that they are 'mathematically proven



The user can configure *UT* to checksum selected files. Authentication times are dramatically reduced if only system files are checksummed

to be undefeatable'. In fact, the package employs a 32-bit CRC algorithm which, according to some experts, offers the optimum balance between speed and security [see *Checksumming Techniques for Anti-Viral Purposes*, *Proceedings of the First International Virus Bulletin Conference*, *September 1991. The discussion in this paper covers the relative merits of CRC versus the approved ISO standards 8731/2 and ANSI X9.9. and concludes that the CRC algorithm is adequately secure for anti-virus purposes. Ed.*]

Fifth Generation Systems's packaging boasts that its generic checker is patented although the software itself is patent pending. Whether *Fifth Generation Systems* will be granted its patent remains to be seen, given that similar technology has been available in the public domain for many years.

UT can be driven either from a full-screen menu or by the use of command line options but, in common with the other components of *Untouchable*, no allowance is made for mono-displays attached to colour adaptors (as found on laptops) to run the programs in mono mode. On these machines, the products' screens are difficult to read - so difficult as to be a positive disincentive to use these programs at all.

Speed of Execution

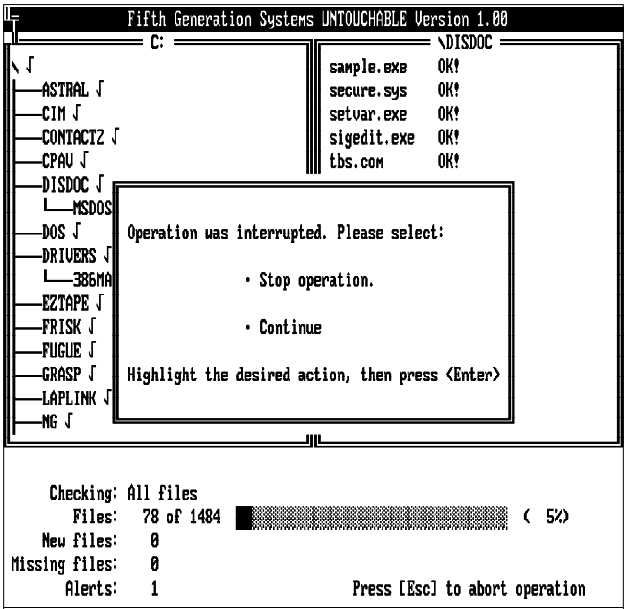
UT runs reasonably quickly, both when it is creating a database of checked files and when it is checking files thereafter. The test machine (Compaq DeskPro 386/16) had 502 executable files occupying 20,240,198 bytes and *UT* created the database file of these in 3 minutes 40 seconds. Subsequent checks of these files

took 2 minutes 49 seconds. Checks can be limited to system files which speeds authentication to a few seconds.

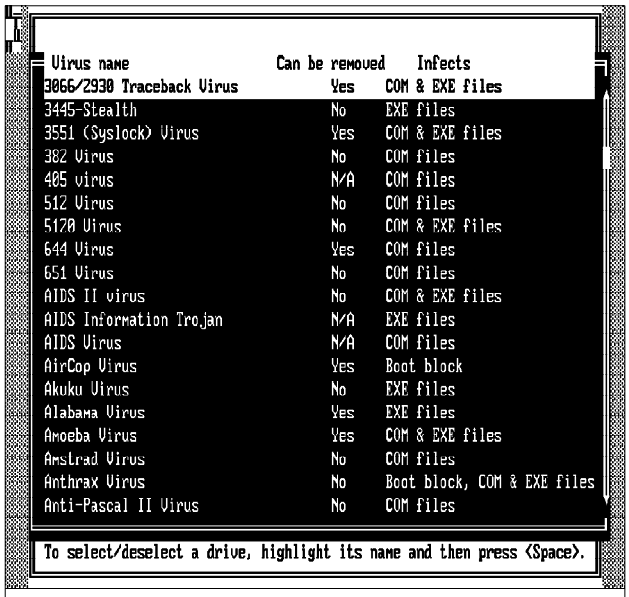
There are a number of ways in which a file may become corrupt - date and/or time stamps may change, a file's length may be altered or its contents tampered with. If *UT* discovers a change, it states that the file has changed while information on the precise cause or nature of the alteration is available by pressing <Enter>. Some changes it can repair since it stores the first 40 bytes of each checked file in its database in plain, unencrypted format as a part of its 50-byte record per file. *Fifth Generation Systems* claims that this repair facility restores 90% of infected files.

Since *Untouchable's* claim to strength lies in its implementation of its generic file checker with its attendant cure capabilities, I put this to the test using *The Norton Utilities*. I modified a byte at the very beginning of a previously recorded file. *UT* correctly identified this modification and correctly reconstructed the file. Then, to simulate the action of an inserting virus (e.g. Einstein), I modified the body of a previously recorded file. In this instance, *UT* could not effect a repair but did report that the file had been modified. Finally, to simulate the actions of a parasitic virus, I increased a previously checked file's length. *UT* should have both reported an error and been able to effect a repair - in fact it only reported an error and did not restore the file.

In tests, *UT* was found to be unaware of stealth viruses and reported no changes to files when Tequila, Haifa, 4K and a host of other stealth viruses were memory-resident. Files previously uninfected and added to the *UT* database were subsequently infected with a variety of such viruses and the resulting changes were not detected by *UT* with these viruses present in memory.



The checking process commences. Operation is interrupted to copy this screen-dump (note progress bar). *UT* will warn of the appearance of new files or the removal of files.



UTSCAN displays some of the viruses which it is capable of detecting and removing. However, a number of viruses in *VB's* critical group 1 are not detected by this scanner.

This shortcoming is by no means unique to *Untouchable*; checksumming programs are notoriously vulnerable to this subversion by stealth viruses. The golden rule when comparing the files on disk with their checksums is to boot from a clean write-protected system disk and run a clean copy of the check-summing program from floppy. The developer is fully aware of this fact which is clearly stated in the documentation. The installation program provides the option to create a 'Safe Disk' (i.e. a working copy of *UT* and its checksum database on a clean bootable disk) so that the user can perform a safe test in this way. By booting the PC from the 'Safe Disk', the user can perform a secure integrity check in a known clean DOS environment. Note that no 'Safe Disk' can be prepared in this way when running under DR-DOS 6.00 with compressed drives.

Scanner Performance

Fifth Generation Systems has placed great emphasis on *Untouchable's* generic virus detection which appears adequate. However, the virus-specific performance of the product is below average. The most positive thing that can be said about the scanner is that it is fast. Scanning the same 502 files as before, *UTSCAN* took just over one minute to complete.

Using the *Virus Bulletin* standard test set (see *Virus Bulletin* September 1991, p. 18 for details), *UTSCAN* found 263 of the 364 parasitic infections and seven of the eight boot sector viruses. Using a larger test set of 748 infections, *UTSCAN* found 429 infections. In subsequent testing against the 'in the wild' test set (see *VB*, January 1992, p. 19), *UTSCAN* found 27 of the 34 parasitic and 11 of the 13 boot sector infections. Notably it missed Spanish Telecom 2, Flip, PcVrsDs, Spanz, Tequila,

Whale, and Spanish Telecom boot. The results render *UTSCAN* currently the poorest performer of the fifteen scanners upon which *VB* regularly reports.

There is a nasty bug in both the *UT* and *UTSCAN* front-end such that on my Dell 316LT laptops, any attempt to enter the 'Options' menu option caused the PC to freeze absolutely solid, requiring a full power-down to clear. The Options menu requires a Control-Enter key sequence to complete it, an example of this product's non-standard user interface.

The memory-resident program, *UTRES*, appears to be entirely virus-specific but the company claims that it also incorporates tests that provide an early indication of suspicious activity in memory. No significant overhead is imposed by *UTRES* on program load and execute but the program also automatically checks boot sectors the first time a floppy disk is accessed and this imposes a noticeable delay in accessing the disk.

Manual

The manual explains clearly the operation of the various programs. The appendix gives descriptions of nine common PC viruses - Brain, Cascade, Dark Avenger, Disk Killer, Jerusalem, Joshi, Italian (which it calls Ping Pong), New Zealand (Stoned) and Sunday. Unfortunately, the manual is marred by several inaccuracies such as defining a partition as 'a small section of a hard disk physically divided from other sections of the disk' whereas, to be precise, it is a logical division, not a physical one.

In Conclusion

No details are to hand regarding technical support for users outside the United States - US users get an 800 number to call when problems arise. Quarterly updates to the virus-specific components (both memory-resident and standalone) will be supplied by BBS. *Fifth Generation Systems* may regret its apparent reticence to supply more frequent updates. With the escalating appearance of new viruses, I would strongly recommend monthly updates to virus-scanning products.

Fifth Generation Systems argue that scanning technology is obsolescent and the company is right in its contention that integrity checking offers the long term defence against the virus threat. However, most PC users still require virus-specific scanners as a convenient method by which to check disks and that requirement will not disappear overnight. The virus-specific performance of this package is undeniably disappointing.

Since *UT* specifically calls *UTSCAN* when building its initial checksum database it is imperative that the latter program should provide a high degree of assurance of detecting any existing virus infection on the system. Given *UTSCAN's* relatively poor performance, the user is recommended to use a more comprehensive and up-to-date scanning program prior to installing *UT* itself. Regarding *UT*, there are just as good, if not better, generic file checkers available from a variety of sources. It should be remembered that generic checking alone never detects the sources of infection but only its effects.

UNTOUCHABLE (UTSCAN V19.04)

Product :	Untouchable
Version :	1.00 (Scanner v19.04)
Developer :	BRM Technologies, Israel
Marketed by :	Fifth Generation Systems, USA, Tel +1 504 291 7221

Scanning Speeds [1]

Test 1 Hard Disk - Turbo	1 min 01 secs
Test 1 Hard Disk - Secure	2 min 05 secs
Test 2 Diskette - Turbo	12 secs
Test 2 Diskette - Secure	26 secs

Scanner Accuracy [2]

Parasitic Viruses	- Turbo	263 out of 364
	- Secure	263 out of 364
Boot Viruses	- Turbo	7 out of 8
	- Secure	7 out of 8
Accuracy Percentage		72.58%

Stamina Test - Encrypting Viruses [3]

Multiple Test: Flip	Fail
Multiple Test: Suomi	Fail
Multiple Test: Tequila	Fail
Multiple Test: Spanish Telecom 1	Pass
Multiple Test: Spanish Telecom 2	Fail
Multiple Test: Group II	Fail
Multiple Test: Group III	Fail

The Acid Test - Viruses 'in the wild' [4]

Parasitic	27 out of 34
Boot Sector viruses	11 out of 13
Detection Percentage	80.85%

[1] The speed test is outlined in the test protocol described in *VB*, April 1991, pp. 6-7.

[2] The test-set is outlined in *VB*, September 1991, p. 18.

[3] This test to determine a scanner's ability to detect encrypted viruses was described in *VB*, October 1991, pp. 7-11.

[4] This test to determine a scanner's ability to detect viruses found in the wild was described in *VB*, January 1992, pp. 18-19.

Technical Details

Test Conditions: The testing for this review was conducted on three PCs. The first, a Compaq DeskPro 386/16 running under DR-DOS 6 was used for the speed tests. There are 1,036 files occupying 31,778,036 bytes of which 502 files occupying 20,240,198 bytes are executable. For the floppy read test, the 360-Kbyte Setup Disk for Microsoft C version 5.01 was used. This contains a total of 12 files totalling 354,804 bytes of which 4 (238,913 bytes) are executable. The virus identification tests were performed on an Apricot 486/25 which houses the test libraries. The third PC is a Dell 316LT laptop which was used to check the readability of the various screens.

PRODUCT REVIEW

Dr. Keith Jackson

ViruGuard

ViruGuard is a half length plug-in card for the IBM PC (ISA bus) which claims to be a 'sophisticated and unique hardware solution for extensive computer virus protection and access control'. The half length card came with a 43 page manual, and a 5.25 inch disk containing various software utilities. The instructions in the manual which explain how to install *ViruGuard* are quite clear. In short, you switch off the PC, find a vacant slot, and insert the plug-in card. When the PC is next rebooted, *ViruGuard* takes control ahead of the BIOS.

It is obviously possible for such an installation process to go awry, and the developers of *ViruGuard* have thoughtfully provided a utility (*PRECHECK*), which copies information from the Partition Table of the hard disk onto a floppy disk. One access control facility in *ViruGuard* encrypts the Partition Table to prevent booting from a floppy disk. If this facility malfunctioned, the information gleaned from the Partition Table would prove essential to recovery.

Utilities are also provided which give technical details about the installed hard disks, and memory usage beyond 640 Kbytes. The latter is important when looking for a suitable memory location at which to install the *ViruGuard* plug-in card. I'm not sure that these utilities provide correct information. One of the utilities (*DRIVSIZE*) reported that the hard disk on my *Toshiba* laptop portable was only 13 Mbytes, whereas it is in fact a 40 Mbyte drive!

Boot Disk

During the installation process, I was amazed to discover that the floppy disk supplied was a bootable disk, version 5.0 of MS-DOS. The documentation recommends that you boot from this floppy disk. Creating a known virus-free bootable floppy disk using the version of DOS currently in use on the PC will in all probability be far less error-prone than using a bootable floppy disk supplied by the manufacturer of *ViruGuard*.

When *ViruGuard* is operating, the messages C1, W1 and R1 are seen to flicker on and off in the middle of the screen whenever a disk access is made. I tracked this down to a 'feature' which the developers of *ViruGuard* call 'Show Drive Status'. It is described in the manual (hidden away in the middle), and the first letter of the message refers to the drive activity (R=Read, W=Write etc), the second letter describes which drive is being used. Flickering letters in the middle of the screen are most disconcerting, and I was relieved to discover how to disable this feature.

System Preferences

ViruGuard offers tailorable options (called system preferences), to suit the particular PC in use. These cover matters such as scanning programs for viruses before execution, ensuring that disk storage and the list of interrupt vectors is correct, enabling password access, and execution (or not) of 'barred' programs.

Doomed To Failure

Program facilities and privileges are controlled by a utility called *DOOM*. There's obviously a joke here, but I'm not quite sure what it is! *DOOM* is a key part of *ViruGuard*. I was disappointed, therefore, to find that on my test PC the *DOOM* utility loaded, displayed its opening screen, and then locked up so thoroughly that a power down was necessary. Perhaps *DOOM* is an apt name after all! The problem is obviously linked to an interaction between the hardware of my test PC and the *ViruGuard* plug-in card. *DOOM* ran happily on my laptop computer but as this has no slots for the *ViruGuard* plug-in card, it was not much use for further testing. *DOOM* executed correctly on the test PC when the *ViruGuard* plug-in card was removed.

Program Load and Execution

With *ViruGuard* operating on my test PC, I tested half a dozen of my most commonly used software packages to see whether they were affected in any way. All except one worked correctly. The *Odyssey* communications package caused *ViruGuard* to issue the message 'Virus pre-scan not complete - System Halted'. The message does not lie, the only way out was by powering the PC down. I have no idea why *Odyssey* failed to execute, but it was definitely consistent (I tried it four times). The warning message issued by *ViruGuard* is not discussed in the manual.

Using the test programs that did operate, and also a reboot, I measured the overhead added by *ViruGuard* to the time taken to load a program (Table 1.). All tests were carried out using *ViruGuard* in its default state.

Program	Normal Load (seconds)	*ViruGuard* (seconds)	Increase
WordStar v6	6.7	10.7	59%
Norton Commander	2.3	6.1	165%
Norton Change Directory	1.4	5.3	278%
Reboot	20.7	53.4	158%
Procomm v2.4.2	4.8	8.4	36%
		Average =	139%

Table 1. Program load overhead

Given that the *ViruGuard* hardware does not have its own CPU, such overhead is inevitable, as some of the processor power must be utilised by *ViruGuard*. The increases in load time shown in table 1 are considerable. I would contend that more than doubling the time taken to reboot a computer will strain most users' patience to the limit. Remember that the time taken by a user to enter his password must be added to the above time (to keep the comparison fair I began the measurements when password entry was complete). Presumably this overhead comes from *ViruGuard* scanning executable files before they are loaded, monitoring files that have changed, and protecting against unwanted changes to the operation of the PC. The *ViruGuard* documentation does not discuss program overhead.

I then measured the overhead imposed on the processing carried out by a program, as opposed to merely loading the program (Table 2.).

Activity	Normal Time (seconds)	*ViruGuard* (seconds)	Increase
Decompress large file	50.4	55.1	9%
Compress a file set	3.3	6.9	109%
Execute *CHKDSK*	15.9	20.4	28%
COPY a directory	12.9	15.0	16%
		Average = 32%	

Table 2. Program execution overhead

ViruGuard thus affects the execution of a program far less than the loading of that program. The manual does not explain the large variation in overhead between loading and execution.

Access Control

The access control facilities of *ViruGuard* centre upon the fact that users must enter a password when the PC is booted before they can use it. Note that there is no user ID or user identifier involved in this process.

A PC with *ViruGuard* installed displays a screen which insists upon just the correct password before the system is allowed to boot. Nothing else. As there is no entry of an ID, it is vital that *ViruGuard* checks for identical passwords. Unbelievably, it fails to do this. *ViruGuard* is happy for a user to change his password to be the same as another user's password. This includes *ViruGuard*'s privileged user (User 0). If this happens, when the PC is next rebooted, *ViruGuard* checks the list of users to see whether the password is valid. Bingo, a mere user has just gained the right to reconfigure the system and to reset everybody else's password (facilities supposedly only available to User 0). The irony of this is that a user may not even know that this happened. User 0 would eventually find out from the strange user access statistics that would result.

I have to admit that I lost faith with *ViruGuard*'*s* basic design at this stage. This last point about identical passwords is so serious as to cast doubt on all the access control features claimed for *ViruGuard.*

Disk Drive Selection

I attempted to test *ViruGuard*'*s* claim that it has knowledge of 142 unique viruses, can detect their presence, and prevent them activating.

To avoid unnecessary infection of my hard disk I wished to do this by executing floppy disks containing the virus infected files. The *ViruGuard* manual claims that it is possible to set up a user's privileges so that they can only access drive A. I attempted this, in fact I tried for the better part of a day, but I failed. No matter what I did, *ViruGuard* refused to boot from drive A, and unsurprisingly when a user is defined with access only to drive A, failure to boot from this drive prevents the PC from booting altogether.

These problems also prevented me from testing *ViruGuard*'*s* virtual drive capabilities, whereby users are constrained to their own (perhaps overlapping) areas of a hard disk. This is a shame, as such a facility has much to offer, for instance if a virus infection was introduced by one particular user, *ViruGuard* should constrain the infection to that particular user's portion of the hard disk.

Minor Points

There are a number of glaring inconsistencies in the documentation for this product, some of which I itemise below:

1. The manual states that the default value of 'Show Drive Status' is disabled. It is actually enabled by default.

2. A utility called *CNFSTRIP* is provided. Nothing explains what it is. I deduced that it removes configuration details from an executable file, but the documentation should explain this.

3. The utility called *HISEARCH* needs a section in the documentation explaining its use. It should be explained better than merely stating that it searches for ROMs in high memory.

4. Text within the *PRECHECK* utility states 'Use the *RECOVER* utility to restore all critical sections'. What *RECOVER* utility?

5. The description of the Partition Table in the glossary states 'Use the *SAVESYS.COM* utility to make a copy of your partition table'. What *SAVESYS* utility?

Observations

I must take issue with some of the claims made in the manual provided with *ViruGuard*. Claims such as '*ViruGuard* traps all present and future viruses, directly or indirectly, and provides complete security against unauthorized computer access' are simply untrue. Nothing is capable of providing infallible protection against all future viruses, and nothing provides complete security. There are only varying degrees of *lack* of *insecurity*.

The access control claim is particularly galling when users can inadvertently (or maliciously) end up all using the same password and logging in as each other. Being able to choose a password that is already in use is weak even when it is associated with a particular ID. It is *ludicrous* when the 'protection' in use allows one user to transmute magically into another! The manual tells us that 'Password protection offers infallible security for private and shared drives'. Hmmm. If readers think that I've been harsh about the manual, the sample sales literature continues in this vein: '... complete peace of mind is guaranteed', '100 % effective against all viruses'. I could go on.

Distributing the *ViruGuard* files on a bootable DOS disk is a bad idea, as this version of DOS may be incompatible with that installed on many PCs (see *Technical Notes*, *VB*, December 1991, p. 3). I imagine *Microsoft* would have something to say about a company giving away free copies of bootable MS-DOS disks, especially as it's the latest version of its software.

The failure of DOOM to execute properly when the *ViruGuard* plug-in card was installed was unfortunate. It removed much of the functionality offered by *ViruGuard*. However, such profound failures simply should not happen on a product that is released for sale.

ADDENDUM

The preceding review was contentious as far as the *ViruGuard* developers were concerned, even though it contains no factual inaccuracies. It is very rare that *VB* agrees to extend a review, however as the anti-virus features were not even examined (they did not work on the test PC), *VB* agreed to examine the anti-virus features of *ViruGuard* using a second version of the product.

On receipt of this second plug-in card, it was obvious that the developers of *ViruGuard* had been listening to the reviewer's previous comments, as some of the problematic features had been corrected. Users now have to enter both an ID and a password, thus preventing the problem whereby all users could inadvertently end up being treated as one and the same. The new plug-in card also worked correctly with an 8088 processor. The covering letter stated that a new README file on the accompanying disk would explain this new version, but unfortunately the README file provided was identical to the previous version; somebody sent the wrong disk.

We were promised the correct disk, but are still awaiting this even though a couple of weeks have now elapsed.

The developers claim that the problems encountered with the *DOOM* utility, and the refusal to boot from drive A, were caused by *ViruGuard* being originally available in 8088 and 80286 specific versions. *VB* had been sent the 80286 version, hence its refusal to operate on an 8088 processor. The two versions have since been combined, but there was nothing in the documentation which discussed this potentially catastrophic problem.

Hard Disk Requirements

Even though *ViruGuard* can prevent a user from accessing a hard disk drive, a hard disk must be present otherwise it will not boot. The reviewer discussed this with the developers, who explained that *ViruGuard* uses some fixed sectors on the hard disk to store its own data, and needs to retrieve this information at boot time. The developers assure us that the sectors chosen are not used on any of the hard disks that they have tested. However, we are reluctant to share their optimism, such tactics seem inherently dangerous.

No utility was provided to remove *ViruGuard* from the hard disk, and on inquiring how to achieve this, the developers provided details of the absolute sectors that had to be erased manually. These sectors contain no plain text as markers and the data contained within them is encrypted. *VB* does not intend to publish details of which sectors of the hard disk are involved; asking users to preform such actions is fraught with danger. What if the wrong sector is erased?

Anti-Virus Features

ViruGuard anti-virus features were tested using the standard test set of 183 virus samples (114 different viruses). Tests were carried out with the viruses resident on a diskette and *ViruGuard* set to prevent any access to the hard disk.

All *ViruGuard* virus detection features were activated. Of the 183 individual virus samples, *ViruGuard* detected 96 as infected prior to loading (52%). This represented 47 of the 114 different viruses used for testing (41%).

ViruGuard also spotted when a program was attempting to become memory-resident, and/or trying to modify either the interrupt vectors held in memory, or an executable file. Note that for these problems to be reported, the virus must already be active in memory. A user is totally reliant on *ViruGuard* preventing any action that the virus may then care to make: a task of Herculean proportions. When these measures of virus activity were taken into account, *ViruGuard* detected some dubious activity with 171 of the 183 test samples (93%).

This still left 12 test samples where *ViruGuard* did precisely nothing and the virus infected program was allowed to execute.

Conclusions

ViruGuard can detect viruses. It can do this before execution commences for about 50% of viruses. It can detect over 90% of viruses when infected files are allowed to execute. However, the vendor's original claim that '*ViruGuard* traps all present and future viruses' is shown to be untrue.

Technical Details

Viruses used for testing purposes : This suite of 114 unique viruses (according to the virus naming convention employed by *VB*), spread across 183 individual virus samples, is the standard *VB* test set. It comprises two boot sector viruses (Brain and Italian), and 112 parasitic viruses. There is more than one example of many of the viruses, ranging up to 12 different variants in the case of the Tiny virus. The actual viruses used for testing are listed below. Where more than one variant of a virus is available, the number of examples of each virus is shown in brackets. For a complete explanation of each virus, and the nomenclature used, please refer to the list of PC viruses published regularly in VB:

1049, 1260, 12 Tricks (Trojan), 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: *ViruGuard*

Developer/Vendor: Ports of Trade, 6 Alcis Str., Newlands, Cape Town 7700, South Africa, Tel: 686-8215, Fax: 685-1807.

Availability: IBM PC/XT/AT or 100 compatible with at least one floppy disk drive, one hard disk, 256K of RAM using DOS version 3.0 or higher.

Version evaluated: 1.45

Serial number: None visible

Price: £125

Hardware used: An ITT XTRA (a PC clone) with three floppy disk drives (1x3.5, 2x5.25), a 32 Mbyte Western Digital hard card, 640 Kbytes of RAM, and an 8088 processor running at 4.77MHz.

A Toshiba 3100SX laptop with a 40 Mbyte hard disk, 5 Mbytes RAM, a 16 MHz 80386 processor and a single 3.5" floppy disk.

FOLKLORE

Offensive Action or Offensive Smell?

Has the *NSA* ('*No Such Agency*') unleashed its first virus in anger? According to *ABC News*, *Sky TV*, and even the *New Scientist*, the boys and girls at *Fort George Meade*, Maryland, managed to implant a 'virus' into an Iraqi mainframe computer which subsequently wreaked havoc on the Iraqi air defence network.

The story has it that a western intelligence agency managed to intercept a shipment of computer equipment destined for Iraq in the months leading up to the outbreak of the Gulf War in January of last year. A 'chip' containing a 'virus' is said to have been implanted into a printer (yes, a printer) which was subsequently connected to a vital mainframe. Upon triggering, the 'virus' is said to have overwhelmed computer resources and blinded Iraqi radar screens.

The technicalities behind this attack are not divulged. Just how a virus could replicate from a printer to a computer is not revealed (the story might be more credible had associated driver software been mentioned) and this has led many observers to dismiss the whole story as pure fabrication.

Journalists are stymied when dealing with spook agencies - a common rumour still in circulation is that *GCHQ* monitors all intercontinental telephone calls made from the United Kingdom and uses voice traffic analysis to intercept key words. Terrorist 'experts' writing for the *Sunday Times* were convinced of the story's truth stating that Nassir Hindawi (the man who valiantly assisted his unsuspecting pregnant girlfriend onto a 747, accompanied by a timebomb) was caught because he said 'Semtex' over the public telephone network! Then there was the (unnamed) French weapons manufacturer which allegedly Trojanised flight control chips in its missiles in case friends turned into foes. The *NSA* 'virus' appears to be the latest invention off the conveyor belt at *Rumor Mill Inc*.

Most seasoned observers have decided that this latest outbreak of media speculation is indeed the result of a hoax - an April 1st edition of US magazine *Infoworld* is understood to have started the rumour. As one wit commented, ''This information was disclosed on a 'need to bullshit' basis.''

Given the remarkable performance of stealth technology (as in *bomber*, not *virus*), laser-guided 'bunker busters' and the awesome panoply of high-tech weaponry which Uncle Sam unleashed against Saddam, such a relatively impotent and unreliable device as a mainframe virus does appear a tad silly.

If spook protocol is maintained, the *National Security Agency* will neither *confirm* or *deny* this story until the year 4000. In the meantime, there's a strong pong of red herring emanating from somewhere.

END-NOTES & NEWS

The ***Little Black Book of Computer Viruses*** (ISBN 0-929408-02-0) by Mark A. Ludwig was published in January 1992. This is the most explicit technical treatise which *VB* has yet seen. Full code listings, both source in assembler and binary in hex are featured, as well as extensive technical instructions to develop a variety of parasitic and boot sector computer viruses. This book is the computing equivalent of the notorious *Improvised Munitions Black Books* which contain detailed instructions on bomb making. The *Little Black Book of Computer Viruses* is part one of a trilogy - the second and third books detail methods to subvert anti-virus programs and the military uses of computer virus code. Interestingly, the copyright for the book says that it was written in 1990 suggesting that it was rejected by many publishers prior to its acceptance by *American Eagle Publications Inc*. *VB* will publish a report in March along with suitable search patterns from the source code listings. Anti-virus software developers should be aware of this book which costs US$14.95 and is available from *American Eagle Publications*, PO Box 41401, Tucson, Arizona 85717, USA.

Springer Verlag has published ***A Pathology of Computer Viruses*** (ISBN 3-540-19610-2) by David Ferbrache which provides a comprehensive and well written account of virus developments in the PC, Mac and Unix arenas. This *Meisterwerk* is essential reading for dedicated virologists everywhere. Recommended retailed price is £24.95.

Taiwanese PC manufacturer ***Mitac*** **plans to bundle £250 worth of software with every desktop computer**. The move is aimed to improve the performance of its *Mistation* PCs and prevent damage or loss due to virus infection. Two anti-virus programs '*Antivirus*' and *Mitac's* proprietary '*MSCAN*' are included in the software sets supplied with the computers.

Leading hotel chain ***Hilton International*** **has purchased 1,000 copies of *Central Point Anti-Virus*** for use in its hotels, sales offices and administration centres worldwide. *Central Point Software UK*, Tel 081 848 1414.

Bull **has introduced three levels of anti-virus service** for its customers to include consultancy, on-site inspection and post-attack recovery. For more information contact Laura Hotham, *Bull HN Information Systems UK*. Tel 081 479 2751.

Symantec **has released *Norton Anti-Virus 2.00*** which claims to detect over 1,000 viruses. The product is *Windows 3* compatible and can be run with *Novell NetWare 286* or *386*, *3Com* and *OS2 Lan Manager* network software. *Symantec UK's* General Manager Jeremy Brown reportedly 'sees approximately 20 new viruses appearing every week'.

CAUTION! Two highly destructive viruses which are in the wild trigger next month. **Michelangelo** triggers on **6th March** (*VB*, January 1992, pp.13-14) while the **Maltese Amoeba** virus triggers on **15th March** (*VB*, December 1991, pp 15-16). Does *your* scanning software detect these viruses?

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