April 1991 ISSN 0956-9979

VIRUS BULLETIN

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

CONTENTS

EDITORIAL

The Long Arm of The Law

On 7th March 1991, the *Metropolitan & City Police Computer Crimes Unit* based at New Scotland Yard hosted the inaugural meeting of the British *National Computer Virus Strategy Group*. The objective of the initial meeting was to lay the foundations for a common strategy to combat computer viruses. In attendence were virus researchers, corporate security specialists and representatives from a number of software companies specialising in virus countermeasures.

The meeting was held in closed session and certain details of what was discussed will not be relayed here; suffice it to say that the British police are currently engaged in the investigation of computer virus incidents and have the powers to prosecute any individual engaged in distributing or deliberately introducing virus code into computer systems within the United Kingdom. The *Computer Crimes Unit* will also seek the extradition of any individual engaged in these activities in any country with bilateral extradition treaties with the United Kingdom. Should there be any doubt about the determination to enforce the law in this respect, virus writers should be aware that Dr. Joseph Lewis Popp, a U.S. citizen, is currently in the United Kingdom to face charges of blackmail for his alleged participation in the AIDS Information Diskette extortion bid (*VB*, January 1990).

Specifically, Section 3 of the *Computer Misuse Act 1990* renders the unauthorised modification of computer material a criminal offence and the law provides for a five-year prison sentence upon successful conviction. This section of the act is interpreted to entail computer virus code which, by necessity, modifies programs and/or boot sectors.

It was made abundantly clear to the specialists present at the meeting that the lawful management of live virus code entails a number of responsibilities and obligations - the most important of which apply to the collection and submission of evidence. Anti-virus software manufacturers, computer virus investigators and researchers are now *bound* to inform victims reporting a virus incident within the UK that a criminal offence has taken place. The victim (be it a company, an individual, a client or otherwise) will then be informed that the crime should be reported to the *Computer Crimes Unit* and contact telephone and facsimile numbers are to be provided. In the event that a victim chooses *not* to report the incident to the police, the researcher, investigator or software company, once informed of the incident, is requested to report the incident to the police but the identity of the victim may be witheld. This national reporting system has been designed to enable the police to chart the progress of virus outbreaks in order to provide the necessary data for empirical analysis of the virus problem in the UK.

Professional doubts abound regarding the *Computer Misuse Act*, particularly regarding those areas of the Act which cover incitement and intent to commit computer crime - by its very nature *VB*, for instance, could be construed as an 'incitement' regardless of its actual intent. Question marks also arise as to whether or not certain provisions of the Act are practically enforceable. However, in the light of its implications, UK based organisations or overseas multinationals with offices in the UK are strongly advised to obtain a copy of the Act from *Her Majesty's Stationery Office*.

From the virus investigator's viewpoint, perhaps the most important eventual outcome of the *Computer Virus Strategy Group Meeting* is that guidelines for crime scene investigation and the collection and submission of evidence have been issued. Moreover, should the specialist parties present at the inaugural meeting remain committed to the police initiative, valuable progress *may* be made in the areas of virus classification, identification and collaborative research.

Computer Misuse Act 1990 (£2.90) HMSO Publications Centre,(mail and telephone orders only) PO Box 276, London SW8 5DT.

Telephone orders 071 873 9090, Enquiries 071 873 0011

Computer Crimes Unit Metropolitan & City Police Company Fraud Department, 2 Richbell Place, London WC1X 8SD.

Telephone 071 725 2409 (soon to be changed to 071 230 1177) Fax 071 831 8845

EICAR

The *European Institute for Computer Anti-Virus Research* (EICAR) has been established in the wake of the 'expert meeting' on computer viruses which took place in Hamburg in December 1990. The organisation will enable the fast exchange of information on computer viruses and other rogue programs in similar fashion to *CERT* in the United States. A specialist research off-shoot body named *CARO* (*Computer Anti-Virus Research Organisation*) has been established to ease the exchange of data, diagnostic tools and binary code between virus researchers in Europe. *CAROnet, EICAR's* secure database of virus samples, has been placed under the control of Professor Klaus Brunnstein at the *University of Hamburg*. The founding members of *EICAR* are: Vesselin Bontchev/Sofia, Prof. Klaus Brunnstein/Hamburg, Christoph Fischer/Karlsruhe, S. Fischer-Huebner/ Hamburg, Fridrik Skulason/Reykjavik, Dr. Alan Solomon/UK, Morton Swimmer/ Hamburg, Michael Weiner/Vienna.

Information from: *Virus Test Center, Faculty for Informatics, University of Hamburg, Schlueterstr. 70, D2000 Hamburg 13, FR Germany*

Tel 40 4123 4158/ 40 4123 4162 e-mail (EARN/BITNET): *Brunnstein@RZ.Informatik.Uni-Hamburg.dbp.de*

TECHNICAL NOTES

Randomised Code

Much of this month's edition of *VB* is dedicated to analysing the probable impact of self-modifying encryption as utilised in virus code. This is a significant development which will affect the development of scanning software in the coming months. (For background information on this subject, readers are directed to *VB*, March 1990, p.12 and *VB*, April 1990, p.10.)

There is the possibility that inexperienced or unscrupulous anti-virus software developers will either fail to realise the implications of this 'dynamic' code or will simply ignore them. For instance a software package which claims to detect the Whale virus should detect it in *all* of its thirty guises. Similarly, a program which claims to detect V2P6 should detect it in *any* of its countless thousands of generations. Detecting one generation of a randomly-encrypting virus is a simple task - it can be done by conventional means using a hexadecimal search pattern - but this is **not** equivalent to detecting the virus itself; that can only be achieved by employing far more complex algorithmic search methods. Caveat Emptor!

Mainframe Viruses

Virus code can be written to function on *any* operating system and *any* processor. Mainframe viruses are perfectly feasible and indeed have already been written - 'rabbit' programs (so called because they bred like rabbits) which accidentally ran amok on mainframes in the 1950s and '60s and '70s were effectively viruses by another name.

There appears to be a widespread and totally irrational fear of mainframe viruses. Factors which militate against the development of malicious mainframe viruses include:

Software development and distribution: Mainframe software is usually strictly controlled and rarely exchanged between user organisations. There is no mainframe equivalent of shareware, and programs directed at mainframes are not generally exchanged on bulletin boards! There is very little opportunity for accidental software contamination. Software implementation is vetted with programs usually being supplied as source code prior to compilation. 'Demo' MVS programs are not supplied on the cover of computing magazines!

System configuration: Viruses need a common platform by which to spread. Personal computers are more or less compatible - they all run under the same operating system (albeit with different version numbers), the hardware configuration is standardised and there are some 36 million machines worldwide. Mainframe operating systems are configured in a site-specific manner and there are hardware-embedded security measures. There is very little commonality between mainframe operating system configurations, which presents a very limited standard platform to a potential virus writer.

Working environment: In total contrast to the world of 'PC anarchy', mainframe computers are policed, audited and regularly monitored.

This is not to say that mainframe viruses cannot be developed, it is just that there is far less incentive to develop such programs because the chance of their spreading is so limited. If the attacker is inclined to destroy programs or data, there are quicker, easier and more insidious techniques available to him. Mainframe operations are *far* more vulnerable to an attack by the corrupt programmer who plants a logic bomb (less detectable than a virus) or the hacker who discovers or installs a trap-door. It should be noted, however, that the careless distribution of *DOS software* using mainframe/WAN links could spread a PC virus globally within minutes.

Virus Evolution

Do computer viruses evolve? No virus which can evolve by its internal resources has yet been discovered, although such a program is theoretically possible. However, when an entire family of related viruses is studied, a process similar to evolution can be observed. This "evolution" starts with one ancestral form, from which numerous variants may be derived, either by deliberate changes introduced into the code or by *extremely* rare random bit errors.

The variants will not all replicate efficiently and some of them will "die out", (apart, that is, from laboratory samples). This process is comparable to the Darwinian theory of natural selection - the "survival of the fittest". New programming techniques may increase the chances of "survival" - the encryption used by V2P2, for example, makes it harder to detect than the Vienna virus, its ancestor.

Just as living organisms have their natural enemies, a virus has enemies in the form of virus detection and eradication programs. Encoded defences which reduce the chance of detection will improve the 'survivability' of the virus.

Minimalism

In most families of viruses, the trend has been towards an increase in complexity - new variants generally comprise additional functions to older variants. However, there are exceptions - most of them from Bulgaria, where local virus writers are participating in a contest - the objective being to write the smallest virus possible.This trend became apparent soon after the discovery in Bulgaria of the original 648-byte variant of Vienna. Within a short period of time several new variants appeared - each smaller than the previous one. The smallest of these viruses was only 348 bytes.

Then a Danish programmer removed all but the most essential code from his 333 byte Kennedy virus, and created the 163 byte 'Tiny' virus. For a period, it was the smallest known specimen, but then a series of highly optimised viruses appeared in Bulgaria. This 'Bulgarian Tiny' family comprised eleven members, the smallest of which was 132 bytes.

For a while it seemed that this record would not be broken, but then a new virus appeared in Bulgaria, which is currently the smallest memory-resident virus known. As the name implies, the Micro-128 virus only occupies 128 bytes. It is not likely to become a serious menace because it has several limitations, but nevertheless it is a fully functioning virus.

As the instructions to make the program memory-resident are not essential to create a functioning virus, it is possible to write an even smaller non-resident sample. It came as no surprise when yet another Bulgarian virus writer proved this to be the case. The resulting virus was incredibly small - only 45 bytes. But would this 'Minimal-45' virus remain the smallest known sample? A cursory examination revealed that there was ample scope for further optimisation without impairing the functionality of the virus.

Predictably, the 'Minimal-45' virus has no side-effects other than replicating by overwriting the first .COM file in the current directory - not a sophisticated method, but one that is crudely functional.

Taiwanese Confusion

As no central authority exists for assigning names to viruses, naming conflicts are common. Perhaps, the greatest current confusion surrounds the AntiCAD/Plastique group of viruses from Taiwan.This is a group of seven viruses, all related to a common ancestor, the Jerusalem virus.

Some researchers prefer the name AntiCAD, as the majority of the variants are targeted against the popular *AutoCAD* ™ design program. The effects were described in the January edition of *VB* - essentially the virus will overwrite the entire contents of the hard disk when the user attempts to execute ACAD.EXE. Other researchers prefer the name 'Plastique', which is contained within some of the variants. The name also refers to the "explosion" sounds some of the variants may make through the computer's speaker.

The Plastique/AntiCAD family contains the following variants:

- **Plastique/AntiCAD-2576**: This appears to be the original virus in the series, although one report indicates that a non-functioning "virus" named HM2 may represent the earliest efforts of the author. A text message inside this virus seems to corroborate that this is indeed the first in the series.

  ```
  To Whom see this: Shit!  As you can see this
  document, you may know what this program is. But
  I must tell you: DO NOT TRY to WRITE ANY ANTI-
  PROGRAM to THIS VIRUS.This is a test-program, the
  real dangerous code will implement on November. I
  use MASM to generate varius virus easily and you
  must use DEBUG aginst my virus hardly, that is
  foolish. Save your time until next month. OK?
  Your Sincerely, ABT Group., Oct 13th, 1989 at
  FCU.
  ```

- **Plastique/AntiCAD-2900**: This is the variant described in *VB*, January 1991. It contains the following encrypted message, which indicates that it is written by the same author as the previous version.

  ```
  Copyright (C) 1988, 1989 by ABT Group.
  ```

- **Plastique/AntiCAD-3012**: A 3012 byte variant, which is also known as 'Plastique 4.51'. It contains the following text:

  ```
  Program: Plastique 4.51 (plastic bomb), Copyright
  (C) 1988, 1989 by ABT Group. Thanks to: Mr. Lin
  (IECS 762??), Mr. Cheng (FCU Inf-Center)
  ```

- **Plastique/AntiCAD-3004**: This variant is closely related to the previous one and contains the same text message, but it also contains the string "COBOL".

- **Plastique/AntiCAD-4096A**: The 4096-byte variants contain code to infect the boot sector as well as program files. The text message reads:

  ```
  PLASTIQUE 5.21 (plastic bomb)  Copyright (C) 1988-
  1990 by ABT Group (in association with Hammer LAB.)

  WARNING: DON'T RUN ACAD.EXE!
  ```

- **Plastique/AntiCAD-4096B**: This variant is functionally similar to the first 4096 byte variant, but is also known as 'Invader', as it contains the following encrypted text:

  ```
  by Invader, Feng Chia U., Warning: Don't run
  ACAD.EXE!
  ```

- **Plastique/AntiCAD-4096C**: This variant is closely related to the previous one, and contains the same encrypted text. A few minor modifications have been made to the code.

Aside from the confusion created by the names already mentioned - Plastique, AntiCAD and Invader, one popular scanning program added to the confusion by referring to the 2900 and 2576 byte variants as 'Taiwan 3' and 'Taiwan 4' respectively. The names were chosen because two genuine Taiwanese viruses had been named 'Taiwan' and 'Taiwan 2'.

The genuine Taiwan family is **not** related to the Plastique/AntiCAD family.

The Taiwan family comprises:

Taiwan-A (Taiwan):	708 bytes
Taiwan-B (Taiwan 2):	743 bytes
Taiwan-C:	752 bytes
Taiwan-D:	677 bytes

All these variants are simple .COM file infectors and add their code in front of the host program. The Taiwan family viruses overwrite the root directory and FAT on drives C: and D:. Trigger conditions are currently being analysed.

COMPARATIVE REVIEW

A Comparative Review of Twelve Virus Scanning Programs

Introduction

The following eleven pages contain a comparative product review of twelve virus scanning programs from manufacturers in the United States, the UK, Iceland and the Netherlands.

The main objective of the review is to provide *some* insight into the comparative speed and accuracy of these programs in detecting computer viruses on disk. A secondary objective is to describe each scanner, comment on its performance, ease of use, documentation and other features which will be of interest to potential users.

The testing protocol is published on pages 6 and 7 so that the readership of *VB* and the manufacturers of the products can see the essential criteria which have been chosen in an attempt to discover the capability of the programs. Readers will notice that the protocol is uncomplicated - this reflects the relatively straightforward software task of searching for computer viruses on disk.

Questions About Objectivity

Having stated these objectives it is necessary to add a number of provisos.

Objective comparative reviews of virus scanning software entail a number of difficulties, one of which is that there is no agreed definition of what differentiates a 'good' or 'acceptable' scanner from a 'poor' or 'unacceptable' one.

It is generally accepted that virus-scanning software of this sort tends to be a compromise between speed and security. Speed does not necessarily reflect accuracy; accuracy, under test conditions, does not necessarily reflect security. It is, for instance, more than probable that a scanner which uses highly-specific search techniques will be less secure but faster than a scanner which conducts a byte-by-byte search.

A discussion of the major factors which determine the development of a scanning program was published in last month's edition of *VB* (*Developing a Virus Scanner*, pp. 7-9) and readers are advised to read that article as supplementary information to this review.

The important point is that the fastest and most accurate scanner in a simple test such as this may not *necessarily* be the most suitable for certain types of diagnostic work. This review is, by necessity, limited. It will not, for instance, be able to tell prospective purchasers whether or not any of the packages tested is actually suitable for use within a particular working environment (other than its being network compatible).

Issues such as the provision of technical support, regular updates, ease-of-use, price, manufacturers' quality assurance procedures and a host of other factors are of critical importance in choosing security software.

In informal consultation with various product manufacturers, one of the most vocal protests against a *VB* comparative review was that the *VB* test-set is not representative because it contains viruses to which some manufacturers have not had access while others have. Moreover, many manufacturers claim to maintain virus libraries against which products rated as satisfactory here would fail.

These criticisms have some validity. Obviously, software developers who have access to the viruses in the test-set have a distinct advantage over those which do not. Therefore, those products which are known to have access to the viruses which now comprise the test-set, or which make use of *VB* hexadecimal search patterns published in *VB*, are clearly indicated in the *features table* which appears on pages 10 and 11.

Regarding the second complaint, all that can be said is that the *VB* test-set contains working computer viruses collected from around the world which have been widely circulated within the research community. The reader should decide whether it is presumptious to suggest that a properly maintained and supported virus scanner, particularly one developed for use in Europe, should detect a high proportion of these viruses.

The overriding premise behind this review is that some comparison between these products is long overdue and that, at the very least, tests conducted against the *VB* test-set will help to identify the 'poor performers' - those products which demonstrably fail to detect a significant percentage of known viruses. The *VB* test-set is, admittedly, too limited to identify those scanners with an exceptionally high accuracy rating against virus collections at present unknown to *VB*. The test-set appears in *Appendix A* on page 8.

Finally, the descriptive section of this comparative review comprises information and comment on the products. As is the case with any review, some statements will result from opinion and observation. Every care has been taken to exclude statements which might prove prejudicial.

Declarations of Interest:

Sophos Ltd. and *Virus Bulletin Ltd.* are under the same ownership.

Frisk Software is directed by Fridrik Skulason, Technical Editor of *Virus Bulletin.*

Jim Bates (*Viscan*), Jan Hruska (*SWEEP*), Ross Greenberg (developer but not distributor of *VPSCAN*), and Ray Glath (*Vi-Spy*) are editorial advisors to *Virus Bulletin*.

TESTING PROTOCOL - VIRUS SCANNERS

VB PRODUCT EVALUATION

IMPORTANT: The evaluator should read this form in its entirity before evaluation proceeds. Any questions should be directed to the Editor, Tel 0235 555139.

Product Category: Virus-specific scanning software.

Objective: To provide the essential criteria by which to judge the relative speed and accuracy of virus scanning programs.

Component Tested: Non-memory resident scanner. TSR interception facilities will not be tested. Testing will be done in a clean DOS environment i.e. having booted from a clean-system DOS diskette. No testing will be undertaken with viruses active in memory. All testing will be conducted from the floppy disk drive.

Hardware: The hardware used should be specified. All comparative testing should be conducted on the same machine with exactly the same file configuration. Full details should be provided about hard disk capacity, drives, clock speed etc. There should be no disk caching. The test machine should have a minimum 20 Mbyte hard disk storage capacity.

Hard Disk Directory Structure: There should be a minimum of 20 sub-directories, organised with at least 3 levels of nesting. Directory configuration should be stated.

Hard Disk Executables: There should be at least 5 Mbytes of clean (uninfected) COM and EXE files spread across the sub-directories. The exact volume of clean executables in megabytes stored on the hard disk should be stated.

Virus Test Set: The virus test set will be supplied by *VB*. It will consist of computer viruses with a proven ability to replicate. No other type of malicious program will be included in the test set. 1 EXE file and/or 1 COM file (if applicable) will be produced for each parasitic (program) virus. Samples of boot sector viruses will be supplied individually on genuinely infected floppy disks. A table showing the viruses used for testing appears in Appendix A.

Note: 1. To facilitate evaluation conduct each test separately against the products - thus all products go through TEST 1, then all products go through TEST 2 etc.

Note: 2. If more than one virus is detected in a single file it should count as only one infection detected.

Note: 3. If any false positive indications occur they should be reported in the evaluation.

Note: 4. Multiply-encrypting viruses such as Whale and V2P6 are present in the test set. Only 1 instance of infection for each such virus chosen at random is included for testing.

TEST 1: TIME TO SCAN AN UNINFECTED HARD DISK

This is a test of the speed (in seconds) with which each scanner can search the entire uninfected hard disk. No viruses should be present in any sub-directory at this stage of testing.

i) Speed with the program undertaking a 'turbo' search, (i.e. the fastest mode) if offered.

ii) Speed with the program undertaking a high security search, (i.e. the most secure mode) if offered.

TEST 2: TIME TO SCAN AN UNINFECTED DISKETTE

This is a test of the speed (in seconds) with which each scanner can search an uninfected diskette containing at least 3 executable files. The evaluator should state the diskette and content densities.

i) Speed with the program undertaking a 'turbo' search, (i.e. the fastest mode) if offered.

ii) Speed with the program undertaking a high security search, (i.e. the most secure mode) if offered.

TEST 3. SCANNER ACCURACY - PARASITIC VIRUSES

A sub-directory will be created into which the parasitic virus test set will be loaded using COPY a:*.* from the four test-set disks supplied. No clean programs or other materials should be present in this sub-directory.

The total number of infected files in the parasitic test set will be recorded (A). A = 306.

The scanner will search the entire hard disk in its 'fast' search speed. A note will be taken of the number of files in which an infection is reported by the scanner (B).

The parasitic virus accuracy test will be repeated but with the scanner in its 'high security' mode and the number of files in which an infection is reported (C) will be recorded.

TEST 4. SCANNER ACCURACY - BOOT SECTOR VIRUSES

The accuracy test will then be conducted against boot sector viruses. The scanner will be run at its 'fast' search speed setting against each infected floppy disk.

The total number of infected disks in the test set (D) will be recorded. D = 7.

The number of infected floppy disks reported by the scanner (E) will be recorded.

The boot sector virus accuracy test will be repeated but with the scanner in its 'high security mode' and the number of infected floppy disks reported by the scanner (F) will be recorded.

TABULATION OF RESULTS

Product	**Version Number**	**TEST 1(i) Speed ('Turbo')**	**TEST 1(ii) Speed ('Secure')**
		TEST 2 (i) Speed('Turbo')	**TEST 2 (ii) Speed ('Secure')**

TEST 3/4 Accuracy % ('Turbo')

$$\frac{(B + E)}{313} \times 100 = \quad \times 100 =$$

TEST 3/4 Accuracy % ('Secure')

$$\frac{(C + F)}{313}$$

ADDITIONAL INFORMATION

Upgrades (Frequency) **Network Compatible** Can the scanner search in memory? Y/N

Y/N Y/N Is there a virus removal facility? Y/N

Does the software overwrite or disinfect infected files? OVERWRITE/DISINFECT

Does the documentation instruct the user to boot from a clean system disk? Y/N

Does the scanner have a user-updatable virus pattern library? Y/N

Is the scanner available by subscription? Y/N Is technical support readily available? Y/N

Does the developer have access to VB virus test set? Y/N Other information/observations.

TEST CONDITIONS

(See Testing Protocol, pp. 6-7)

The scanners were executed from a 3.5 inch diskette. Where timing measurements were taken, the times included the time required to load the program from the diskette, perform any initialisations and (where applicable) automatic memory scans. Disk caching software was disabled.

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

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

VIRUS TEST-SET (Appendix A)

In the following list of 306 parasitic infections, the letters C and E inside brackets refer to COM and EXE file infections and where a number is also given, this refers to the number of infective variant samples:

1049 (CE), 1067 (C), 1260 (C), 1600 (CE), Eddie-2 (CE), 2144 (CE), 2480 (C), 405 (C), 417 (C), 440 (C), 492(C), 4K (CE), 5120 (CE), 516 (C), 600 (C), 696 (C), 707 (C), 717 (C), 800 (C), 8 Tunes (CE), 905 (E), 948 (CE), Agiplan (C), Aids (C), Aids II (C), Alabama (E), Ambulance (C), Amoeba (CE), Amstrad - V847 (2 C), Jerusalem - Anarkia Variant (2 C), Anthrax (CE), Plastique 2 (CE), Anti-Pascal 1 (2 C), Anti-Pascal 2 (3 C), Armagedon (C), Attention (C), Bebe (C), Best Wishes (C), Blood (C), Black Monday (CE), Burger 1 (C), Burger 2 (C), Burger 3 (C), Cancer (C), Carioca (C), Cascade (1) 01 (C), Cascade (1) 04 (C), Cascade (1) Y4 (C), Cascade Format (C), Casper (C), Christmas in Japan (C), Christmas Tree (C), Cookie (E), Dark Avenger (CE), Datacrime 1 (C), Datacrime 2 (C), Datacrime II (C), Datacrime IIB (E), Datalock (CE), dBase (C), DBF Blank (CE), December 24th (E), Destructor (CE), Diamond A (CE), Diamond B (C), V2000 (C), Dir (C), Diskjeb (CE), Dot Killer (C), Durban (CE), Dyslexia (C), Eddie (C), Eddie- 2 (CE), Evil (C), Fellowship (2 E), Fish-6 (CE), Flash (CE), Flip (CE), Fu Manchu (CE), Ghostballs (C), Jerusalem Groen Links variant (CE), Guppy (C), Hallochen (E), Hymn (CE), Icelandic 1 (E), Icelandic 2 (E), Icelandic 3 (E), Internal (E), Itavir (E), Jerusalem B (CE), Jocker (E), Jo-Jo (C), Joker 01 (C), July 13th (E), Kamikaze (E), Kemerovo (C), Kennedy (C), Keypress (CE), Lehigh (C), Leprosy B (CE), Liberty (CE), Love Child (C), Lozinsky (C), Machosoft (CE), Jerusalem Mendoza variant (C), MG (C), MG 3 (C), MGTU (C), Mix1 (E), Mix1-2 (E), MLTI (C), Monxla (C), Murphy 1 (CE), Murphy 2 (CE), Nina (C), Nomenklatura (CE), Nothing (C), Number of the Beast, variants A to F (C), Ontario (CE), Oropax (C), Parity (C), Perfume (C), Phoenix (C), Piter (C), Pixel 1 (C), Pixel 2 (C), Pixel 3 (C), Plastique 1 (CE), Plastique 2 (CE), Polimer (C), Polish 217 (C), Proud (C), Prudents (E), Jerusalem PSQR variant (C), Rat (E), Russian Mirror (C), Saddam (C), Scott's Valley (CE), Shake (C), Slow (CE), South African 1 (2 C), South African 2 (2 C), Spanish (CE), Spanish Telecom (C), Spyer (C), Stupid (C), Subliminal (C), Sunday (CE), Suomi (C), Suriv 1.01 (C), Suriv 2.01 (E), Suriv 3.00 (CE), SVC Version 4 (CE), Sverdlov (CE), Svir (E), Sylvia (C), Syslock (C), Taiwan (C), Taiwan 2 (C), Tenbyte (CE), Terror (C), Tiny (C), Tiny Family 2 (T-133) (C), Tiny Family 2 (T-134) (C), Tiny Family 2 (T-138) (C), Tiny Family 2 (T-143) (C), Tiny Family 1 (T-154) (C), Tiny Family 1 (T-156) (C), Tiny Family 1 (T-158) (C), Tiny Family 1 (T-159) (C), Tiny Family 1 (T-160) (C), Tiny Family 1 (T-167) (C), Tiny Family 1 (T- 198) (C), Trackback (CE), TUQ (C), Turbo 488 (C), Turbo Kukac (C), Typo (C), V-1 (C), V2000 (Die Young) (C), V2P2 (C), V2P6 (2 C), Vacsina - TP04 (C), Vacsina - TP05 (C), Vacsina - TP06 (C), Vacsina - TP16 (C), Vascina - TP23 (C), Vacsina - TP24 (C), Vacsina - TP25 (C), Vacsina - TP05 (C), V-Alert (C), Vcomm (CE), VFSI (C), Victor (CE), Vienna 644 (C), Vienna 1 (C), Vienna 2 (2 C), Vienna 3 (C), Vienna 4 (C), Vienna 5 (2 C), Vienna 6 (2 C), Violator (C), Virdem Gen (C), Virdem 1 (C), Virus 90 (C), Virus B (C), Virus 101 (CE), Voronezh (CE), VP (C), W13-A (C), W13-B (C), Westwood (CE), Whale (C), Wisconsin (C), XA-1 (1) (C), XA-2 (2) (C), Yankee - TP33 (2C 1E), Yankee - TP34 (CE), Yankee - TP38 (CE), Yankee - TP41 (CE), Yankee - TP42 (CE), Yankee - TP44 (CE), Yankee - TP45 (CE), Yankee - TP46 (CE), Old Yankee 1 (E), Old Yankee 2 (E), and, Zero Bug (C).

The following boot sector viruses were also used:Aircop, Brain, Disk Killer, Italian, Joshi, Korea, New Zealand 2.

VIRUS BULLETIN

THE PRODUCTS

Mark Hamilton

The Quest For The Perfect Scanner

The purpose of this comparison is to transcend the marketing hyperbole in order to ascertain *some* indications of the actual relative performance of twelve aspiring contenders. For this review, a very large suite of 313 infected files was selected from over 200 virus variants. Of these, 306 were parasitic viruses and seven were boot sector viruses. All the parasitic virus samples were contained in either .COM or .EXE files.

The 'Acceptable' Criteria

Scanning accuracy was my primary concern. Provided the scanner correctly identified an infection, it passed the test; it did not matter, for instance, which name a scanner ascribed to a particular virus. A 90 percent accuracy rating suggests that a virus scanner is being well maintained and this percentage has been selected as a benchmark for the acceptable products. Scanning speeds for a clean 20 Mb (full) hard drive and floppy disk should be within reasonable bounds - I consider the acceptable times to be less than 4 minutes for a hard disk and less than 90 seconds for a floppy disk. Many other factors influenced my assessment of the package, one of the more important of which concerned the documentation which should clearly instruct the user to boot from a clean system floppy disk prior to using the scanner.

The packages were tested in turn and the results are produced in tabular form on page 15.

Individual Evaluations

F-PROT Version 1.14a

Supplier	*Frisk Software International* (Iceland)
Country of Origin	Iceland
Telephone	+354 (0)1 694749
Price	Free (non commercial), US $1.00 per PC (commercial)
Update Frequency	Monthly
Pros	High detection rate at a moderately fast speed.
Cons	Poor quality documentation.

F-PROT is an extensive package which will be the subject of an in-depth review for next month's *Virus Bulletin*. Its author, Fridrik Skulason, has elected to provide separate programs to check memory, boot sectors and files for viruses, known as F-SYSCHK, F-DISINF and F-FCHK respectively and it is these three programs which have been examined for this review. F-PROT was developed while Skulason was working at the *University of Reykjavik* and the version supplied for review is shortly to be superseded by version 2.

While not being the fastest scanner, it is one of the more secure, finding 301 of the 306 infections. It was interesting to note that Skulason does not adhere to the *Virus Bulletin* naming convention, despite the fact that he is the journal's Technical Editor. I can only assume that this is a naming convention adopted for the US market.

F-FCHK (the file scanner) is capable of disinfecting a large number of viruses; files which it can not disinfect, it offers to delete. F-DISINF (the oddly-named boot sector scanner) correctly detected six of the seven boot-sector viruses. The seventh (Aircop), was reported as "This boot sector is not a usual DOS boot sector. It may be infected with an unknown virus" - this demonstrates competent programming but, unfortunately, did not count as an 'identified' virus in the accuracy percentage results. I personally doubt the wisdom of separating the memory, boot sector and file virus scanners into three separate programs. I believe that such a scheme is user-*un*friendly.

Printed documentation did not accompany the review software, but there were a few disk-based text files which explain the use of the software. The lack of printed documentation lets down an otherwise competent, strong package.

Dr. Solomon's Anti-Virus Toolkit Version 4.26

Supplier	*S&S International*
Country of Origin	UK
Telephone	UK +44 (0) 442 877877 (US distributor) +1 612 937 1107
Price	£84 including quarterly updates
Update Frequency	Quarterly. More frequently to subscribers of *S&S's Virus Fax International*
Pros	Fastest scanner for uninfected disks.
Cons	Only updated quarterly.

Dr. Alan Solomon is a seasoned anti-virus campaigner and has done more than any other individual to make corporate UK aware of the threat of PC viruses. This latest version of his software was issued on 13th January 1991.

FINDVIRU is the *Toolkit's* virus scanner which can be launched either from the DOS prompt or from within the *Toolkit's* menu system (TOOLKIT). The same program detects viruses in memory as well as on disk and within files. *FINDVIRU* has been optimised to scan extremely quickly when

it detects no viral activity - indeed this was the fastest scanner when looking at non-infected hard disks - but it slows considerably when it finds a virus. In addition to virus and Trojan detection. The developer has also included search patterns for two benign joke programs (BUGS and BUGSRES) from the Soviet Union.

The Toolkit documentation has not changed appreciably for a number of months, except that it is now delivered in smart new livery complete with a cardboard slipcase. The documentation needs redesigning - users should be able to find detailed installation and program invocation instructions at the beginning of the manual, not more than three-quarters of the way through at chapters 5 and 6 respectively. Full marks to the developer for pointing out the need to boot from a write-protected diskette before running this program in the very first sentence of the manual.

If this software is used in conjunction with others, its naming convention (at times unique to *S&S*) could cause confusion. This underlines the need for a standard nomenclature and classification scheme for computer viruses.

In common with the *Norton Anti-Virus*, *FINDVIRU* detected 1260 and V2P2 as if they were the same virus. It failed to detect V2P6.

HTSCAN Version 1.12

Supplier	Harry Thijssen
Address	Zeskant 85, 6412 DV Heerlen, The Netherlands
Country of Origin	The Netherlands
Telephone	(not available)
Price Shareware	Fl 2.50 per PC
Update Frequency	(not available)
Pros	Affordable.
Cons	Distributed by Bulletin Boards only.

HTSCAN is one of two Dutch packages, the other being *TBSCAN*, and it shares with it a common virus pattern file format. This is, incidentally, the same format that IBM uses with its virus scanner. Both *HTSCAN* and *TBSCAN* recom-

VIRUS-SPECIFIC SCANNING SOFTWARE/FEATURES

Product	Developer	No. of Viruses in documentation	Memory Checks Conventional	Memory Checks Upper	Network Aware	Single File Check
F-FCHK 1.14a	Fridrik Skulason	244	No [1]	No	Yes	No
FINDVIRUS 4.26	S&S International	357 [2]	Yes	No	Yes	No
HTSCAN 1.12	Harry Thijssen	233 [3]	Yes	No	Yes	No
Norton NAV 1.01	Symantec	150 [4]	Yes	No	Yes	Yes
PC-EYE 2.0b	PC Enhancements	254	No [1]	No	Yes	No
SCAN V74-B	McAfee Associates	475	Yes	No	No [5]	No
SWEEP 2.23	Sophos	302	Yes [6]	No	Yes	Yes
TBSCAN 2.0	ESaSS	233 [3]	Yes	No	Yes	Yes
VIRFIND 1.4	Visionsoft	143	Yes	No	No [7]	Yes
VISCAN 3.03	Bates Associates	357	Yes	Yes	Yes	Yes
VI-SPY 5.0	RG Software	238	Yes	No	Yes	No
VPCSCAN 1.1a	Microcom	137	No	No	Yes	No

Notes

[1] There is a separate memory check program included with the package.

[2] S&S says this program finds a total of 357 strains, but only lists 263.

[3] HTSCAN and TBSCAN both use a virus definition file kept up to date by Jan Terpstra (IBM Holland).

[4] Applies when files WHALE.DEF and UPDATE01.DEF are read-into *Norton Anti-Virus*.

[5] A network-specific version called NETSCAN is available, but was not tested. [6] Instructions to scan memory must be put in the SWEEP.ARE file.

[7] There appears to be a serious bug in this program which makes it unusable on a network. See text.

mend the use of VIRSCAN.DAT, which contains virus patterns prepared by Jan Terpstra, which is distributed on several bulletin boards within Europe and the US - though not on either *Compulink* (CIX) or *Compuserve* (CIS).

HTSCAN also recognises a pattern file called HTSCAN.DAT which can be prepared using the virus patterns published in *Virus Bulletin*. It even accepts wildcards so that recognition patterns for most of the known viruses can be entered and subsequently scanned for. The downside is that the pattern file is straight ASCII and entries are not checksummed. This means that inadvertant mistakes or deliberate tampering with the pattern file would render it useless. For the purposes of this review, I relied solely on Terpstra's patterns.

Regarding boot sector viruses, surprisingly *HTSCAN* found the Aircop infection but *missed* Brain. Interestingly, despite the fact that both *HTSCAN* and *TBSCAN* both use the same pattern file, *HTSCAN* detected more viruses. Its display is straightforward and shows the number of directories, files and bytes scanned. It can also produce a report file which details which file(s) are infected. *HTSCAN* will optionally delete or rename infected files and you are prompted for your consent before this process commences. For what is virtually free software, this is a reasonably competent effort but is not one of the high-flyers in terms of either speed or accuracy.

Norton Anti-Virus Version 1.01

Supplier	*Symantec Corporation*
Country of Origin	USA
Telephone	(USA)+1 408 253 9600 (UK)+44 (0) 628 776343
Price	£149 including 1 year subscription
Update Frequency	Monthly
Pros	Norton name. Nice user interface.
Cons	Relatively poor detection rate.

Norton Anti-Virus was reviewed in the January 1991 edition of *Virus Bulletin* since which time the company has issued the first update disk (dated 13th February 1991). This update adds patterns for "12 Tricks Trojan", "Plastique 5.21", "Un-

Definition Format [8]	**Virus Removal** (Disinfect/Overwrite/Delete)	**VB Test Set** [10]	**Pattern Library** [11]	**Resident Scanner/Monitor**
Proprietary	All Methods	Yes	Yes	Yes - Device Driver
Proprietary	Delete[9]	No	Yes	No
IBM/VB	None	No	No	No
Proprietary	Disinfect/Delete	No	No	Yes - Device Driver
Proprietary/VB	Delete	No	No	No
N/A	None [9]	No	No	Yes - TSR Program
VB	Overwrite/Delete	Yes	Yes	No
IBM/VB	None	No	No	Yes - Device Driver and TSR
VB (Abbreviated)	None	No	No	No
Proprietary/VB	None	Yes	Yes	Yes - Device Driver/TSR
N/A	Overwrite/Delete	Yes	Yes	No
N/A	None	No	No	Yes - non-virus-specific monitor

[8] "Proprietary Format" indicates that you must obtain virus patterns from supplier; ''Proprietary/VB indicates that VB patterns are used in addition to the manufacturer's; proprietary patterns; "IBM/VB" means that definitions are compatible with those published by IBM and VB; and "VB" indicates that definitions are directly compatible with those published by Virus Bulletin.

[9] Certain viruses can be disinfected using other utilities from this supplier.

[10] Direct access to VB virus collection.

[11] States whether there is a user upgradeable pattern library.

known Plastique", "Plastique", "Keypress", "Aids 2" and a further three Whale variants. This brings the product's portfolio of virus patterns and identities to 150.

The virus patterns are stored, in compressed form, as part of the device driver NAV.SYS which now has a disk size of 33818 bytes, which reduces to around 28 Kb in memory. If you are using a 386-class PC and DR-DOS 5.x or a memory manager which can relocate device drivers, Norton's device driver can safely be tucked away in high memory, where its large memory footprint causes less of a problem.

However, it is no match against either *Bates'* or *S & S's* device drivers, in terms of memory usage. Having the device driver present in memory does impact heavily on file-based operations, typically by adding an overhead ranging from 25 percent up to a higher limit of 300 percent for file write or program execution operating system calls.

When reading a known-clean hard disk, *Norton Anti-Virus* did not produce any false positives, but, in separate tests, it did fail to detect one infection of WHALE even though it found three other WHALE infections

PC-EYE Version 2.0b

Supplier	*PC Enhancements*
Country of Origin	UK
Telephone	+44 (0) 707 59016
Price	£79 + £45 per annum update fee
Update Frequency	Monthly
Pros	A good secure scanner. Good documentation.
Cons	Does not detect encrypted viruses (Casper, V2P2, Proud etc.).

I reviewed this package last October, as part of *PC Business World's* comparative review of anti-virus software, at which time it claimed to detect 68 viral strains and achieved a 43 percent detection capability (version 1.17 9A). Since this time, *PC Enhancements* have expended much energy in research and development, consequently expanding the scanner's library to 254 strains. This effort is reflected in the scanner's results - it now achieves a very respectable accuracy rating.

This places it firmly among the front runners and, if *PC Enhancements* add the capability to detect self-modifying and highly encrypted viruses, it looks set to be a force to be reckoned with. The authors have also enhanced the product's scanning speed: in October, it took 8 minutes to scan a 20 Mb hard disk, this has been reduced to under four minutes.

PC Enhancements supply a separate memory scanner as part of the *PC-EYE* package which scans all memory, including extended and expanded memory.

The documentation is brief, to the point and less patronising than most. The manual gives precise instructions on how to install the package - including a prominent warning to boot from a clean system floppy disk.

This package is testament to a positive attitude adopted by its authors in the face of constructive criticism.

SCAN Version 6.3V74-B

Supplier	*McAfee Associates*
Country of Origin	USA
Telephone	(USA)+1 408 988 3832
Price	Shareware (?) Available on Bulletin Boards including CIX
Update Frequency	Monthly
Pros	Useful documentation, acceptable detection accuracy.
Cons	May only be available commercially.

McAfee's scanners are often considered to be the *de facto* industry standard anti-virus products but he is now facing strong competition in his home market, particulary from *Symantec*, *Microcom* and, more recently *S&S* who have recently appointed *Ontrack Systems* of Minneapolis as a US distributor for *Dr. Solomon's Toolkit.* Faced with the test-set used for this review, *SCAN* fared acceptably well and is very much on a par with the *S&S* product, in terms of its detection capabilites. Documentation is disk-based and terse but does provide a useful list of the viruses which *SCAN* detects, the names of McAfee's companion products which disinfect files and details of various viral infection characteristics. This is arguably the most useful aspect of this ubiquitous product.

I have received an as yet unconfirmed report that McAfee is pulling out of the shareware market and that his product portfolio is now available on a formal commercial basis.

SWEEP Version 2.23

Supplier	*Sophos Ltd.*
Country of Origin	UK
Telephone	+44 (0)235 559933
Price	£295 for 12 issues
Update Frequency	Monthly - annual subscription
Pros	Reliable package, well documented.
Cons	More expensive than its principal competitors.

Since I reviewed *SWEEP* for *Virus Bulletin* in December, *Sophos* has added a menu shell program which is designed to make the program easier to use. However, this shell does not overcome one of my major criticisms of the product - the need

to create a special file to check files (or memory) that do not form part of the default settings of the program. Just about every other supplier manages without this special file, why can't *Sophos*?

SWEEP detected all the viruses in the test suite, which is not surprising because the product developers had access to it. It is not, however, among the faster scanners for speed. It also has a propensity to report false positives in infected files - by this I mean that it often finds patterns for more than one virus in files that are known to be infected by a single virus.

Sophos' product is the first anti-virus package to have been granted a UKL1 certification by *CESG* and while this test is by no means exhaustive or infallible, it does provide a certain assurance that it complies to a defined standard. *Sophos'* documentation is of a high standard and well presented in a linen-cloth binder and slip-case.

TBSCAN Version 2.0

Supplier	*ESaSS*
Country of Origin	The Netherlands
Telephone	+31 (0)80 787771
Price	Free
Update Frequency	(not available)
Pros	Contains code to control principal interrupts.
Cons	Reliance on the presence of search patterns at the beginning of virus code.

TBSCAN is the second of the Dutch products included in this review (see *HTSCAN* above) and is written by *Novix International*, developers of the *Thunderbyte PC Immunizer* add-in card (see *Virus Bulletin*, January 1991, back page).

The company also produces a device driver version - which can also be loaded as a TSR (Terminate-Stay-Resident) program - called *TBSCANX* which shares the same pattern file as *TBSCAN* and *HTSCAN* and which is researched and published by Jan Terpstra of IBM Holland. *TBSCANX* monitors DOS file write calls and warns you if it detects that the file you are writing contains a virus - very useful for monitoring file copying operations.

If *TBSCANX* detects a virus, it asks you whether you want to continue and if you decide not to, it returns a "disk full" condition to the calling program (e.g. the DOS COPY command). It also only monitors writes to files with COM or EXE extensions which limits its usefulness.

TBSCAN, on the other hand, defaults to scanning memory, .COM, .EXE, .SYS and .OV? files as well as boot sectors and partition tables; a command line option will force it to scan all files. It seems that this program tries to analyse the file being scanned and uses one of three algorithms on each file. These are confusingly called "scanning", "tracing" and "analyzing". I say "confusingly" because these terms do not accurately describe the algorithm the program decides to adopt. For example, if the program says it is analysing the file, what it is in fact doing is scanning the entire file looking for a matching virus pattern, whereas if *TBSCAN* says it is scanning the file, it means that it has found the program's entry point and is scanning 3 Kb from that point.

I believe that this program's ingenuity (complexity?) could prove to be its downfall - the fact that its detection rate is lower than its compatriot (*HTSCAN* which uses exactly the same search patterns) is very telling. This scanner relies on the fact that the virus patterns are located within the first 3 Kb of the virus code which further reduces this package's usefulness.

Documentation is supplied on disk in Dutch and English.

VIRFIND Version 1.4

Supplier	*Visionsoft Ltd.*
Country of Origin	UK
Telephone	0800 590868 (Freephone, UK only) +44 (0)274 610503
Price	£98
Update Frequency	Unknown
Pros	Freephone to order product.
Cons	Abysmal detection rate, extremely slow, poorly documented.

This scanner is part of *Visionsoft's Immunizer* package which the company describes as "Truly the last word in virus protection" and exhorts you to "Protect yourself now, with the world's most powerful Anti-Virus system".

I noticed a serious bug with this package: the test suite of viruses was installed on logical drive L (drive M being the last drive) of the Apricot's hard drive, but *VIRFIND* would only allow me to scan drives A to J inclusive - thus, the scanner thought that there were 3 fewer drives than in fact existed. This was confirmed on the Compaq - whose drives are lettered A to I inclusive - upon which *VIRFIND* reported A to F as being available for scanning.

I could not believe how poorly this scanner performed and therefore the detection tests on this product were repeated for verification. The results were identical. *VIRFIND* has a questionable virus detection capability and it is excruciatingly slow at scanning disks.

Regardless of the merits, or otherwise, of the other components of the *Immunizer* package, it is my judgement that this scanner has absolutely no place in a corporate environment.

VI-SPY Version 5.0

Supplier	*RG Software Systems Inc.*
Country of Origin	USA
Telephone	USA +1 602 423 8000
Price	US$ 250, site licences available
Update Frequency	Monthly
Pros	Most accurate US package.
Cons	Relatively slow scanning speed.

VI-SPY (version 2.0) was reviewed in May 1990's edition of *Virus Bulletin* since when it has been radically improved. Although *VI-SPY* detected over 95 percent of the virus suite, this was at the expense of speed and was one of the slower of the scanners tested. Taking the US-authored packages as a group, *VI-SPY* comes out on top in terms of its detection rating.

Uniquely - for an American package - it identifies viruses principally by the *Virus Bulletin* name, followed in brackets by other common names for the same virus. For example, for the 4K virus, *VI-SPY* reports: "4K (4096, Frodo, IDF, 100 Years, Stealth)". Unusually for an American product, *RG Software* do not offer any specific disinfection routines. *VI-SPY* prefers the safe option, that of overwriting and then deleting infected files - this option can be disabled and, in any case, the program asks for permission to delete files.

The documentation is very good and included an A5 fold-out sheet which detailed the new options that are available with version 5 of the software.

VPSCAN Version 1.1a

Supplier	*Microcom Software Division*
Country of Origin	USA
Telephone	USA +1 919 490 1277 UK +44 (0)483 740763
Price	£85 + £55 for a year's updates
Update Frequency	(not advertised)
Pros	Fast search engine.
Cons	Inadequate update frequency.

This program appears to have been updated just once since October 1990 and it has the dubious distinction of being the only package (reviewed both then and now) to have actually worsened in terms of its detection rating. Last October, *VPCSCAN* detected 70 percent of the test suite then in use, it now detects 58.15 percent of the current test suite.

On scanning uninfected drives, this package - like several others - achieves a good turn of speed but this is countered by its poor detection rating. As well as failing to detect numerous parastic viruses, it was unable to find some of the newer boot sector varieties, most notably the Korea virus. The printed documentation for the scanner is sparse - most of the manual is given over to *VIREX-PC*, the commercial version of *Flushot+* marketed by *Microcom Software Division*. Information about the scanner component is mostly to be found in a disk READ.ME file. (A beta-test version of VPSCAN v 2.00 is currently undergoing trials. Ed.)

VISCAN Version 3.03

Supplier	*Total Control/Bates Associates*
Country of Origin	UK
Telephone	*Total Control* +44 (0) 488 685299 *Bates Associates* +44 (0) 533 883490
Price	Single Copies: £20 Updates cost £10
Update Frequency	Monthly
Pros	The most accurate scanner tested at an affordable price.
Cons	

Since *VISCAN* was reviewed in *PC Business World* last October, it has been completely redesigned. Originally written in *Quickbasic*, Bates has rewritten it in Assembler to create a program not only one tenth of its original size, but one that is considerably faster - back in October, it took over 7 minutes to scan a 20 Mb hard disk, now it takes just over 3 minutes.

Bates' attention to detail and accuracy is well-known and is exemplified by this product. For example, where another scanner identifies a particular virus as "Tiny Family (2)", VISCAN will identify the same virus as "Tiny Family 2 (T-xxx)", where "xxx" is "133", "134", "138" or "143". This attention to detail means that more accurate advice can given more quickly to users who suffer a viral infection.

This scanner routinely scans all memory up to 1 megabyte and, in common with *Norton Anti-Virus*, scans all files by default - there is a command line option to instruct *VISCAN* to scan program files only (.COM, .EXE, .OV?, .APP, .PGM, .SYS, .DLL and .PIF). Bates' virus pattern files are encoded and he includes a program (SIGEDIT) so that users can add, edit or delete their own patterns. After a pattern file has been in use for three months, a warning message suggests that you obtain an update - *Sophos* and *S&S* display similar warnings.

This scanner will also optionally display an advice screen when it detects a virus, the contents of which are tailored according to the attributes of the particular virus detected.

The documentation starts by intoning in large, bold type that you should reboot your PC from a clean, write-protected DOS diskette before running the scanner. The remaining documentation explains the program's operation and reminds you of the benefits of regular backups. (See pp. 26 -27.)

RESULTS TABLE - SCANNING SPEEDS [TESTS 1 (i), 1(ii), 2(i), 2(ii)] (See Testing Protocol, pp. 6-7)

Package	Version	Hard Disk 'Turbo'	Hard Disk Secure	Diskette 'Turbo'	Diskette Secure
F-FCHK	1.14a	6:23	11:47	0:35	1:06
FINDVIRUS	4.26	1:09	2:20	0:34	0:39
HTSCAN	1.12	2:18	3:35	0:39	0:52
NORTON ANTI-VIRUS	1.01	1:56	N/A	0:39	N/A
PC-EYE	2.0b	1:12	3:57	0:24	0:43
SCAN	V74-B	3:41	6:14	0:59	1:26
SWEEP	2.23	3:38	5:25	0:39	0:50
TBSCAN	2.0	1:25	2:53	0:14	0:32
VIRFIND	1.4	N/A	84:39	N/A	5:10
VISCAN	3.03	3:18	3:24	0:19	0:24
VI-SPY	5.0	3:01	5:00	0:30	0:54
VPCSCAN	1.1a	1:07	4:11	0:17	0:46

RESULTS TABLE - SCANNER ACCURACY [TESTS 3/4] (See Testing Protocol, pp. 6-7)

	Parasitic Viruses		Boot Sector Viruses		Accuracy Percentage	
Package	**'Turbo'**	**Secure**	**'Turbo'**	**Secure**	**'Turbo'**	**Secure**
F-FCHK 1.14a	301	301	6	6	98.08%	98.08%
FINDVIRUS 4.26	287	287	6	6	93.61%	93.61%
HTSCAN 1.12	226	226	6	6	74.12%	74.12%
NORTON NAV 1.01	216	N/A	6	N/A	70.92%	N/A
PC-EYE 2.0b	287	299	7	7	93.93%	97.76%
SCAN V74-B	285	285	7	7	93.29%	93.29%
SWEEP 2.23	306	306	7	7	100.00%	100.00%
TBSCAN 2.0	222	226	7	7	73.16%	74.44%
VIRFIND 1.4	N/A	109	N/A	5	N/A	36.42%
VISCAN 3.03	306	306	7	7	100.00%	100.00%
VI-SPY 5.0	294	294	6	6	95.85%	95.85%
VPCSCAN 1.1a	177	177	5	5	58.15%	58.15%

KNOWN IBM PC VIRUSES (UPDATES)

Amendments and additions to the *Virus Bulletin Table of Known IBM PC Viruses* as of 26 March 1991. The full table was published in the January 1991 edition of *VB*. Hexadecimal patterns can be used to detect the presence of the virus with the 'search' routine of disk utility programs or, preferably, can be added to virus scanning programs which contain pattern libraries.

Type Codes

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

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

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

SEEN VIRUSES

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

```
10 past 3          B840 008E D8A1 1300 B106 D3E0 2D00 088E ; Offset 068
```

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

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

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

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

Azusa - DR: A short boot sector virus which may damage data on diskettes larger than 360 Kb. Upon activation the virus disables COM1: and LPT1: (*VB*, April 1991)

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

Crazy Eddie - CER: A 2721 byte virus from Bulgaria.

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

Deicide - CN: A primitive 666 byte overwriting virus. Upon triggering, the virus destroys the first 80 sectors on drive C:. According to a text message in the code, this virus was written by a person named Glenn Benton. (*Deicide*: "killer, or killing, of a god" - *Oxford Concise Dictionary*)

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

Doom II-B - CER: This variant of Doom 2 has not replicated under test conditions. Infected programs hang or overwrite the FAT and root directory on drive C: Version B uses the same encryption method as the other known variant.

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

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

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

Frere Jacques-B - CER: Variant of Jerusalem, closely related to Frere Jacques virus. Detected by Jerusalem (1) pattern.

Gergana - CN: A simple 192 byte virus which has no side-effects.

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

Grither - CN: A 774 byte variant of Vienna detected by the Vienna (2) pattern published in the January 1991 edition of *VB*.

Iraqui Warrior - CN: A 777 byte variant of Vienna in which numerous NOP instructions have been added to avoid detection by current scanners.

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

Jerusalem-1600 - CER: This variant is somewhat shorter than the standard Jerusalem virus at only 1600 bytes. The virus is detected by the Jerusalem-USA pattern (*VB*, January 1991). Awaiting analysis.

Justice - CR: A 1242 byte virus. Test computers hang when an infected program is executed.

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

Kylie - CER: A 2272 byte variant of the Jerusalem virus which plays a tune when activated.

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

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

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

Minimal-45 - CN: This Bulgarian overwriting virus is the smallest known to date with a length of 45 bytes. When executed it overwrites all .COM files in the current directory with its own code.

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

Mirror - ER: This virus is 924 bytes long but infected programs may grow by a maximum of 940 bytes. When the virus triggers it reverses the contents of the screen showing a mirror image of the original display.

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

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

PcVrsDs - CER: A destructive 1904 byte virus. The sample obtained was from an infected site in Ireland. The virus will trigger on 23rd September 1991. (*VB*, April 1991)

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

Phantom - CR: A 2201 byte virus which contains an encrypted message stating that it was written in Hungary.

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

Plastique/AntiCAD-3004 - CER: Very closely related to the 3012 byte variant of Plastique. The virus contains the string 'COBOL'. It is detected by the Plastique (1) pattern published in *VB*, January 1991.

Staf - CN: A 3083 byte "demonstration" virus which appears to have no harmful effects. The virus contains the following text: ''Virus Demo Ver.: 1.1 - Handle with care! By STAF (Tel.: (819) 595-0787).''

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

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

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

Taiwan-D - CN: Closely related to Taiwan-C, but only 677 bytes. It can be detected by the same pattern as Taiwan-C but this is located at offset 1F1.

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

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

Vienna-822 - CN: The effects of this variant have not been determined but appear to affect the boot sector. It is detected by the search pattern for the GhostBalls virus (*VB*, January 1991).

Virdem-792 - CN: A destructive variant of the Virdem virus which overwrites the first 5 sectors on all disks when it triggers.

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

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

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

REPORTED ONLY

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

Discom - CER: A 2053 byte variant of the Jerusalem virus.

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

Jeff - CN: A destructive 814 byte virus which writes garbage to the hard disk upon triggering.

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

Little Pieces - ER: A 1374 byte virus which occasionally clears the screen and displays: "One of these days I'm going to cut you into little pieces."

Plaque - CEN: An overwriting 590 byte virus which may trash disks.

Swiss-143 - CN: A small and very primitive virus which has no side-effects.

Sylvia B - CR: Reported to be a rewritten version of the Sylvia virus.

SPECIAL FEATURE

Jim Bates

Mark A. Washburn - Walking the Research Tightrope

The business of taking MS-DOS computer viruses apart so that they can be analysed and classified is done solely to provide information that will enable rapid identification and effective protection for computer users likely to be at risk from the malicious targeting of such code. Researchers worldwide are becoming far more accomplished in their dissections and analyses but all of them are still severely overworked trying to keep pace with new viruses as they are discovered.

The Virus Writers' Fallacy

The whole research effort operates under the one over-riding premise that there is no such thing as a computer virus which cannot be taken apart. Since virus code (by definition) must be totally mobile, it must also be completely self-contained - including such tricks as self-modifying code, pre-fetch queue manipulation, anti-debugging code and direct hardware access.

The particular collection of selected "tricks" used, together with their respective order and location within the program provides a recognisably unique "profile" by which a virus may be identified and dealt with. Virus writers recognised this fact some time ago and in some cases have gone to extreme lengths to hide the details of this "profile" from prying eyes by introducing various layers of encryption and randomisation of their code, even varying these from infection to infection.

The fact that virus code must be self-contained and therefore must be capable of decrypting itself before execution, seems to have escaped the restricted 'intellects' involved in virus production.

Nevertheless, some of them still persist in attempting the impossible - a truly undetectable virus which will escape detection by virtue of its anonymity.

A Bogus Researcher

One of the most stubborn of these individuals is known to researchers since he operates under the bogus guise of being a virus "researcher" and produces live virus code which contains his name and address!

I refer to Mr. Mark Washburn of the United States, who has produced V2P1 (1260) V2P2 and latterly the V2P6 virus.

That this man is allowed to write and distribute virus code with impunity is symptomatic of just how badly legislation against computer crime has fallen behind in various countries. By no stretch of the imagination can his "work" be classified as virus research since his code has produced nothing of which responsible researchers were not already aware.

What he *has* achieved is to distribute virus code of a most dangerous kind, through channels which lack any security and in such a way that there is no doubt that samples of his code are (or soon will be) in the hands of virus writers who will undoubtedly use his virus vehicles to deliver destructive trigger routines.

Reports of virus analyses produced for public information must necessarily be carefully examined before publication to ensure that they do not provide technical details which could be of use to virus writers.

(Editor's note: the encryption methods used by V2P6 will not be analysed in detail here, but a discussion of the simple structure and infection method of this virus follows and will prove informative. Anti-virus software developers and bona-fide researchers requiring information on the algorithmic methods to detect V2P6 should contact *VB*, *Bates Associates*, UK (0533 883490) or Fridrik Skulason at the *University of Iceland* (+35 4 1 694749).

V2P6 - The ''Patternless Monster''

In the case of the V2P6 virus, the technical details are quite sparse and completely innocuous. In the original sample there is no trigger routine, the virus does **not** become memory-resident and only COM files are effected. The infective length is between 1801 and 2350 bytes and no attempt is made to hide the increase in length from normal DOS operations.

A single COM file is infected each time the virus code is executed (the 'one-shot' replication method), first in the current directory, and then by searching along the designated PATH as specified within the machine environment area.

Infected files are marked with the ubiquitous 62 second marker in the date/time field of the file's directory entry and this is used as a recognition flag by the virus itself. There are several bugs within the code, some of which affect how the virus selects files to infect. For example, it is obvious that file lengths of 10 and 63746 respectively were intended to be minimum and maximum limits but careless coding has resulted in the virus infecting all COM files *except* these two file sizes.

The internal V2P6 code is unremarkable. From a researcher's point of view, this virus must be classified as "armoured" because as well as primary encryption (and randomisation), it contains a primitive routine which is supposedly designed to make disassembly difficult.

This is a linked INT 03H/INT 01H handler which decrypts and recrypts certain sections of the virus code "on the fly". Such routines have already been observed in other virus code and present only a minor irritation to experienced researchers.

Self-Modifying Encryption

Washburn's main effort (as in his other viruses) has been directed at randomising the primary decryption routine in such a way as to nullify the normal pattern recognition techniques used in most virus scanners.

More than half of the virus code is taken up with the convoluted calculations and bitmapping gymnastics needed to generate a randomised decryptor for each infection of the virus. This renders V2P6 capable of producing hundreds of millions of possible combinations for the decryption routine. All of the viruses that Washburn has produced seem designed to impress the researcher with just how "clever" he is at producing randomised encryption/decryption routines.

Unfortunately for him, simple pattern recognition is only a small part of the armoury of good scanning software. His approach produces a different kind of detection profile which is paradoxically even easier to recognise than a straightforward hex pattern.

Who Has Benefited?

It is therefore apparent that Washburn's efforts have added **nothing** to existing knowledge about MS-DOS computer viruses other than to increase the already heavy workload of dedicated researchers around the world who must necessarily disassemble his nonsense. Continued production of such "research" viruses can only be detrimental to the research effort and his masquerade should be stopped forthwith. If he had not already demonstrated his irresponsible attitude to the virus problem, he might be better employed in helping the rest of us in a positive way by analysing existing virus programs for the general benefit of computer users everywhere.

As it is, there can be little doubt that eventually one of his programs (or a recognisable derivative) will appear as a vehicle for a malicious trigger routine. As will be seen, evidence is accumulating which suggests that this has already happened - the destructive Casper virus (which *VB* has obtained as a source code listing and which includes Washburn's name, address and copyright notice!) and the anonymous Violator virus reveal an uncanny resemblance to Washburn's V2P1 (1260) program. (Mr. Washburn denies having developed the Casper virus and claims that this is a 'hacked' version of V2P1. Ed.)

In the United Kingdom, there is a substantial body of opinion which maintains that Mr. Washburn should be held personally responsible should his code (or, indeed, modified versions of it) infect personal computers in this country.

Virus Attribute Summary

Name: V2P6

Origin: U.S.A. (Mark Washburn)

This is a non-resident, 'one-shot' COM file infector (including COMMAND.COM) which uses multiple encryption and randomisation. No static code exists between generations of V2P6, therefore it is not possible to extract a hexadecimal search pattern for this virus. There is no trigger routine. All COM files, except those with lengths of 10 bytes and 63746 bytes, are infected. Infected files are marked with a 62 seconds marker in the directory entry Time field; this is the virus' self-recognition signature.

Washburn's Legacy - The Threat of Randomised Code

```
Hello, all anti-virus "researchers" who are
reading this message...

I am glad to inform you that my friends and I are
developing a new virus, that will mutate in 1 of
4,000,000,000 different ways! It will not contain
any constant information, so no virus scanner
could be detecting it...

The virus will have many other new features that
will make it completely undetectable and very
destructive!

the Dark Avenger
```

This typically infantile message, purportedly from the Bulgarian virus writer calling himself 'Dark Avenger' was uploaded to Bulgarian BBSs in March 1991. It subsequently appeared on *Fidonet* and we are grateful to Michael Weiner, the Austrian virus researcher, for forwarding this transcript.

Self-modifying encryption, first identified in Washburn's 1260 virus, is now being adopted elsewhere and the threat that this method will be employed by the Bulgarian 'virus factory' should be taken seriously. Virus scanning software will be somewhat impeded by the appearance of such code - the development of search algorithms for each such specimen is both painstaking and time-consuming. However, **no** virus will ever be 'undetectable' - CRC and cryptographic checksums will remain effective long after virus-scanning has ceased to be practicable. Meanwhile, researchers are confident that virus-specific detection will remain viable for the foreseeable future.

Mr. Washburn's Explanation

The existence of numerous 'Lab' viruses (code written for experimental purposes to 'assist' the development of anti-virus software) presents both technical and ethical dilemmas to anti-virus investigators. Mark Washburn's 'experimental' viruses, which are reported in this month's edition, present particular difficulties as the programs do not appear to have been written with malicious intent. The publication or open discussion of the encryption methods employed would be unwise because these viruses effectively invalidate the hexadecimal search pattern as a reliable means to detection. Worse still, his initial methods were made available in the public domain which accounts for the 'hacked' Casper virus which *VB* reported in January 1991 (p. 24). In view of our intention to report his activities, it was decided that Mr. Washburn should have the opportunity to explain himself; with this in mind a letter was sent to him at his address in the United States.

8/2/91

Dear Mr. Washburn,

We are currently analysing the 1260, V2P2 and V2P6 computer viruses, as well as a destructive virus called Casper which formats track 0 of the hard disk on an infected PC.

From reading Patricia Hoffman's listing, it would appear that these viruses were written for experimental purposes and their distribution has been carefully limited. VB will publish a technical analysis of this series of viruses (albeit with some sensitive information removed) in the March 1991 edition.

It would be helpful to publish a statement by the author of these programs providing the rationale for their development and an insight as to how and to whom the programs were sent for analysis. The appearance, in the 'wild', of source code for the Casper virus has caused much concern - any clues as to how this source code came to be in circulation would be most welcome.

Thank you in advance for your cooperation.

Yours faithfully,

Edward Wilding Editor

February 21, 1991

Dear Mr. Wilding

I originally created V2P1 (the 1260) as a demonstration of programming technique. Specifically, my intent was to exhibit a problem of relying upon fixed scan strings as the sole method of detection. The 1260 (object) was labeled as a demonstration virus and publicly offered. V2P2 and subsequent experiments have restricted access.

I do not have a copy of the 'Casper' virus; however, it is my understanding that the object code is derived from a disassembly of the V2P1 demonstration object code.

Generally, for virus code of this type, the decryption routine is the primary target for the scanning pattern. The basic principle behing V2P1 is to pseudorandomly generate a decryption routine that is used to mask the effectual virus code. The total effect is that it appears as if every byte of code changes.

I believe the V2P6 experiments created the first true patternless viruses. To this date, I have not received contrary evidence. For example, the V2P6 derivatives can generate thousands of 4-byte (more than 6500 5-byte) GREP patterns; in contrast, the 'Ontario' virus can be detected with one 5-byte pattern.

Because my experiments have created the patternless 'monster', so to speak, I've developed a TSR monitor that effectively stops all executable file infectors. SECURE v2.22 also warns of boot sector viruses and offers basic Trojan protection.

I look forward to a transcript of your review of SECURE or my virus experiments.

Sincerely,

Mark A. Washburn

VIRUS ANALYSIS 1

The Violator Virus - Burger's Continuing Legacy

The technical competence of virus writers varies considerably, from abysmally poor to reasonably proficient but this is not usually a consideration which affects the actual functioning of virus code (apart, of course, from programming bugs).

Over a period of time, a researcher will develop a "feel" for the style and structure of particular viruses and may even be able to link apparently dissimilar programs and reasonably ascribe them to the same original author. Such stylistic analyses have little value to computer users but they may become extremely useful as computer misuse legislation is adopted worldwide and law enforcement agencies begin to home in on the criminals responsible for the problem.

One of the most obvious links discovered to date concerns the origins of the Violator virus and it highlights the undoubted advantages of detailed disassembly of virus code over the faster (but less effective) sparse analysis technique. Before examining the conclusions of a stylistic analysis, I will first describe Violator.

Brief Description

This is a non-resident virus which infects only COM files of between 10 and 64000 bytes. Infection takes place on a 'one-shot' basis (i.e. one file is infected each time the virus is executed). Files in the current directory are attacked first and when they are all infected, the search continues by accessing files within directories listed in the system PATH setting. A date controlled destructive trigger routine is incorporated and described below. The code is not encrypted and responds readily to automatic disassembly.

Operation

From the initial jump instruction at the head of the host COM file, the virus first collects an offset value which is subsequently used throughout the code to address various data items. This value is modified during the infection routine to reflect the length of the new host file. Once this offset has been collected, it is used to access the original three bytes of the host header and these are replaced at the top of the file.

A check is then made on the current DOS version and processing returns to the host program if this is earlier than version 2.00. If the DOS version is acceptable, the virus sets up its own Disk Transfer Area and then checks the current setting of the system date to see whether the trigger routine should be executed.

The code to check the date is extremely clumsy but the criteria are as follows:

If the date is before 15th August 1990 then the trigger is not executed. If the month is January to July (inclusive - any year) the trigger is not executed. If the date is the 1st to the 14th (inclusive - any month) the trigger is not executed. This selection of dates for the trigger routine does not affect the infection routines which are processed every time the code is executed. Once the trigger routine has run, processing continues with the normal infection routines.

Trigger

The trigger routine consists of a small loop which uses the BIOS INT 13H call to attempt to format the first track of all floppy drives from A to Z. This will obviously destroy the boot sector of any unprotected floppy disks in those drives. The virus does not install a special critical error handler and no check is made for error conditions. This means that unless there is a write-enabled disk in every floppy drive, the DOS error handler will report either "Sector not found" or "Drive not ready" errors to the screen. No attempt is made to initialise the format instruction correctly.

Infection

The infection routine begins by accessing the Environment Segment belonging to the host program and searching for the "PATH=" command. Once this is found, its position is stored for later use.

A search mask of "*.COM" is then used with a call to Function 4EH of INT 21H to find the first matching file. Attributes are set to include System and Read Only files. Once a file is found, the time field is checked for a value of 1FH (31 = 62 seconds) in the seconds field. If this is found, the file is assumed to be infected and the search continues with a Function 4FH (Find Next) call. If no matching (uninfected) file is found in the current directory, processing collects the first parameter in the "PATH=" statement and continues the search there. This process continues until all of the directories (delimited with a semi-colon) noted in the path statement have been searched.

Once a suitable file is found, the usual processes of collecting and storing the attributes and the date/time field are executed and the file is then opened for write access. **Files which were set to Read Only access are still at risk since the virus resets these temporarily during infection to allow write privileges.**

The next phase collects the first three bytes of the new host and stores them within the virus code. The 1055 bytes of the virus code are then written to the end of the host file and a new offset is calculated for the initial jump. The new jump instruction is written to the beginning of the file and the file date and time field is restored to its original value but with the seconds field set to 1FH (62 seconds). The file is then closed and the

attributes restored to their original value before the virus passes control back to the original host program. A recognition pattern for Violator has already been published (*VB*, January 1991) and this analysis has confirmed this string as accurate and effective.

The Washburn-Burger Connection

The operations described above are unremarkable and are similar to those found in most parasitic viruses. What *is* interesting is when a stylistic analysis is conducted and considerable similarity is revealed between large sections of the code in the Violator, Casper and V1 viruses.

Casper is a 'hacked' development of the 1260 virus (V2P1) written by Mark Washburn in the U.S. and V1 is listed in Ralf Burger's book *Computer Viruses - A High Tech Disease* (*VB*, October 1989, p.19) as a version of the Vienna virus.

There is no equivocation in this comparison; the similarities are numerous (even to the duplication of NOP instructions and bugs). The temptation to *speculate* upon the original derivation of Violator is irresistible:

Given three viruses from (apparently) three different sources, the first question is which came first. In this case there is no doubt that the original Vienna virus was first since it is a disassembly of this which appears as the V1 listing in Burger's book. The book was originally published in Germany in 1987 (the English translation appeared around a year later), so we can place Vienna at pre-1987.

Dating the other two is less easy. File dates are not reliable since they can be changed so easily, but in this case there are other indications concerning the original dates of Casper and Violator. The earliest report that I can find concerning Violator appears in the Patricia Hoffman listing from the United States, dated November 1990. The Hoffman listing is a first class initiative and it deserves success. Unfortunately it seems to be plagued with many inaccuracies in the virus reports which add to the confusion concerning exactly how particular viruses operate. In this case for example, Violator is reported as follows:

> *When a program infected by the Violator virus is executed, what happens depends on what the system date is set to. If the date is prior to August 15, 1990, the virus will infect 1 .COM file located in the current directory, adding 1,055 bytes to the program. If the date is August 15, 1990 or after, the virus will not affect any files.*

This is plainly at variance with my observation of the current sample which is infective regardless of the date and triggers as described above. However, the reported text strings and other details match exactly and do seem to indicate that we are referring to the same virus. The same entry reports that: *"The Violator virus was submitted in August, 1990 by an anonymous user of the HomeBase BBS*". This places Violator no later than August 1990 so we only need to date Casper to complete the timescale. The source listing of Casper (which includes Washburn's name and address) contains the message "*Copyright (C) Mark Washburn, 1990. All Rights Reserved*". Assuming that this 'copyright' message is correct, this enables us to date Casper to 1990, but the exact month of its development is unknown.

Unfortunately, it is impossible to draw absolutely firm conclusions from the above speculation but the alternatives are interesting in themselves. Violator and Casper could *both* have been written by the same hand or both could have been *copied* from the Burger book, but independently.

It is also possible that sections of Violator could have been copied from Casper (and, less likely, vice versa). It should be remembered that source code for the Casper virus has been widely distributed.The presence of certain incorrect checks and the position of some of the NOP instructions leads me to suspect that Violator was probably copied from the Burger book, as was Casper. The impression gained during disassembly of Violator is that it was written by someone with virtually no knowledge of PCs who had access to some virus source code and a rather poor reference book to DOS. It is impossible to determine whether the same author was involved in both cases, even though Violator contains text claiming *"Copyright (c) 1990 RABID!*" and Washburn has certainly demonstrated his desire to corner the 'market' through claiming copyright.

More importantly, this examination highlights once again the fact that virus source code is immensely more dangerous than its assembled equivalent because source code will continue to spawn modified strains. Burger's publication of source code to the Vienna virus has spawned more viruses and variants than any other single action. Washburn's V2P1, V2P2 and V2P6 are *all* based on the Burger listing. (Even disregarding his public dissemination of virus code, the existence of the destructive Casper virus which is derived from 1260, has served to discredit Washburn as a responsible researcher.) Ultimately Violator, which is *clearly* related to the V1 source code, is another damning indictment of Burger.

Technical Editor's comment: Analysis of the Casper virus assembly listing indicates that is **not** developed from a disassembly of the V2P1 executable, as Washburn claims (see page 20). Rather it is created by modifying the virus' *source code* which indicates that V2P1 (1260) source code is in circulation.

It is my opinion that Casper and Violator were developed *independently* but that they share a common ancestor; namely the original Vienna virus. It is unlikely that the author of Violator had access to Casper (or V2P1) as Violator contains none of the special code which makes V2P1 different to Burger's published Vienna variant, V1. It is equally unlikely that the author of Casper had access to Violator as its code contains none of the mistakes found in the latter virus.

There is one seemingly indisputable connection - Washburn used Burger's published source code to create his V2P1 (1260) virus and the source code to 1260 is now in circulation. The publication and/or distribution of source code represent a greater threat than the distribution of binary virus code and are acts of gross irresponsibility.

VIRUS ANALYSIS 2

Fridrik Skulason

Azusa - Complicating the Recovery Process

One of the "new" viruses listed in this edition has been named 'Azusa' for reasons unknown. Several 'real world' outbreaks of this virus have been reported in the USA. In some respects Azusa resembles the New Zealand (Stoned) virus. It only occupies one sector and infects the Master Boot Sector (MBS) of hard disks, as well as diskette boot sectors.

Operation

When a computer is booted from a diskette infected with the Azusa virus, the virus will reserve 1K of RAM, copy itself into this area and redirect INT 13H. This is typical behaviour for a boot sector virus. However, the next step Azusa performs has not been used by any previously known virus.

As sectors are normally 512 bytes long, Azusa (which is 368 bytes long) does not occupy the last 144 bytes of the sectors it infects. These 144 unoccupied bytes are left intact.

The implications of this are two-fold.

A diskette boot sector usually contains various text messages in this area, such as:

```
Non-System disk or disk error
Replace and strike any key when ready
Disk Boot failure
```

A boot sector infected by the Azusa virus will, therefore, contain *exactly* the same system text messages. This could possibly mislead or confuse an investigation of the boot sector using disk utilities such as the *Norton Utilities ˜* or *PC Tools ˜*.

Contained at the end of the MBS is a 64 byte table called the Partition Record which describes the partitioning of the hard disk (by FDISK) and the location of the bootable DOS Boot Sector. Each DOS partition has its own DOS Boot Sector. However, only one of these sectors (usually allocated to drive C:) is booted when the machine is switched on.

Unlike the New Zealand virus which stores the entire MBS (including the Partition Record) and overwrites all 512 bytes of the sector with its own code, Azusa does not store the original MBS anywhere. Instead the virus itself fulfills the most important function of the MBS by examining the Partition Record and locating the bootable partition.

This is done by checking if the first byte in any of the table entries contains the value 80H. If Azusa finds no indication of a bootable partition, in the sector containing the virus, it assumes the computer was booted from an infected diskette, not a hard disk. In this case, it will attempt to infect the MBS of the first hard disk in the system.

Azusa then loads and executes the original boot sector of the infected diskette and stores it on Track 39, Head 1, Sector 8.

If a bootable partition is found, the computer must have been booted from an infected MBS and the virus checks an internal counter and increments it, unless it has reached 32. When this value is reached, which happens when the computer has been booted 32 times from an infected hard disk, the virus will disable LPT1: and COM1: by altering the port addresses located at 0040:0000 and 0040:0008.

When an attempt is made to read from a diskette or write to it, the virus checks whether the diskette motor is running. If not, the boot sector is read and checked for an existing infection. If no infection is indicated, the boot sector is stored on Track 39, Head 1, Sector 8. Azusa then attempts to camouflage itself, by incorporating parts of the boot sector *into* itself. It will copy an 8-byte area located at offset 3, which usually indicates a text string such as "MSDOS3.3". It also copies the last 144 bytes of the *original* boot sector. The camouflaged boot sector is then written back to the diskette.

Damage

The virus may destroy data on diskettes larger than 360 Kb. Just like the Den Zuk virus, Azusa may cause loss of data on 3.5 inch or 1.2 Mb diskettes. The location used by the virus (39,1,8) is at the very end of 360 Kb diskettes which will not be used unless the disk is nearly full. The higher-capacity diskettes have more tracks and Track 39 is right in the middle of the diskette. However, Azusa is less destructive than Den Zuk as it only occupies a single sector, not the entire track.

Disinfection and Recovery

Removal of Azusa from a diskette involves moving Track 39, Head 1, Sector 8 to the boot sector, thus overwriting the virus. Disinfecting the MBS is more complex as the original boot sector is not stored *anywhere*.

Possible approaches include:

- Restoring the MBS from a backup copy. Unfortunately, most PC users do not make a backup copy of this critical area, although it only takes a couple of minutes using *NU* or a similar utility.
- Backing up the entire hard disk, verifying the backups, reformatting the hard disk and partitioning it with FDISK. This is the 'brute force' method.
- Zeroing out the MBS and using a program such as *Norton Disk Doctor ˜* to reconstruct it.
- Writing down the data in the intact Partition Record and then overwriting the virus with "generic" MBS code from a similar computer. The critical location data can then be re-entered into the uninitialised Partition Record.

The last two methods have one possible drawback due to the *slight* possiblility of differences between boot sector code on different machines.

VIRUS ANALYSIS 3

Richard Jacobs

PcVrsDs - A Sleeping Bomb

PcVrsDs (PC Virus DOS?) is a destructive parasitic memory-resident virus that infects .COM and .EXE files, increasing their length by 1904 bytes. Unlike the majority of viruses, which are circulated first within the research community, it was reported to *VB* by a member of the public, in this case in the Republic of Ireland. The temporary name of this virus comes from a text string contained within its code; since this string has no set pronunciation, I would suggest that a new and more manageable name be allocated to this code and published in the May edition of *VB*.

Compared to many new viruses the measures taken by this virus to avoid detection are relatively simple. The bulk of the virus is encrypted using the simple technique of subtracting a randomly chosen key from each byte. The virus starts with a 27 byte decryption routine which adds this key back on to the next 1871 bytes.

A Common Self-Recognition String

The virus recognises itself using the 5 byte string "PcDos" at the end of the infected file. This string is not encrypted and enables the virus to detect reliably whether or not files have been infected. However, ''PcDos'' is of no use as a detection pattern because this is a text string which appears in many legitimate DOS programs; its inclusion in a scanner would cause an embarrassing number of false positive alarms!

File Infection

The virus infects .EXE files by the fairly common technique of adding itself to the end of the file and altering the file header so that the virus code executes when the host program is run. After the virus has finished, execution jumps to the normal entry point of the file.

The way in which .COM files are infected is more unusual. Rather than writing to the end of the file and altering the initial JMP instruction to point to the virus, PcVrsDs writes itself in front of the normal file, to be loaded at offset 100H. When an infected .COM file is executed the virus is immediately run, with no need for a JMP instruction. After the virus has made itself memory-resident, it moves the original file down to offset 100H and jumps to that address, returning control to the host program. When an infected program is executed, the virus decrypts itself and then checks whether or not it is already memory-resident. If it is, control is immediately returned to the host program, otherwise the virus copies itself to a temporary location and transfers control to this copy. The procedure from this point depends on the date.

Destructive Trigger Routine

If it is Monday the 23rd of any month **not** in 1990, the virus will reformat head 0 of the first 32 cylinders of the fixed disk, using an undefined table of descriptive bytes. This table provides the cylinder number, head number, sector number and number of bytes in each sector. Should these values be undefined (as will be the case when this virus triggers) the data in these areas of the disk will be completely unreadable. **During this formatting operation the critical error handler INT 24H is disabled, so that no errors will be reported until the process has finished.**

Once the disk has been formatted the following message is displayed, and execution is terminated.

```
PcVrsDs Version 1.00
Copyright (c) VirOP 1990
```

It should be noted that there is a second destructive routine (albeit very much less pernicious) contained within this virus which is described later.

Keyboard Services Interception

On any other date, INT 21H is reset to point to a routine within the virus and the file is reloaded and executed, using INT 21H function 4BH (Load & Execute). When the program has finished, the virus terminates, leaving itself memory-resident.

On any Monday which is **not** the 23rd of the month and **not** in 1990, INT 16H (BIOS Keyboard services) is intercepted.

The INT 16H routine monitors which INT 16H function is called. Unless the "Read Next Keyboard Character" function is called, control is returned immediately to the normal INT 16H routine.

If this function is called, the keyboard is read, using the normal routine and then a counter is checked. This counter is initially set to 255 by the virus. While this counter is greater than zero, the only function of the routine is to decrement the counter. Once the counter reaches zero and if the character read is a printable character, the ASCII value returned is incremented and the counter is reset to 13. So any routine that uses the ASCII value, rather than the keyboard scan code, will read one key in thirteen incorrectly.

The routine has been included to corrupt data input and (possibly) to irritate the user - paradoxically, its inclusion within the code actually increases the likelihood of the virus being discovered. Early discovery and removal of the virus will obviously pre-empt the destructive routines from triggering.

The INT 21H routine handles all activity of the virus, apart from INT 16H. The majority of functions are passed directly to the normal DOS routines.

The following functions are intercepted, 11H (Find First File), 12H (Find Next File), 3DH (Open File) and 4BH (Load & Execute).

Two new functions are also created. The first is a simple check to see if the virus is already memory-resident, the second handles the relocation of the original program, required when the virus has finished executing in infected .COM files and before control can be transfered to the original file.

The 'Find First File' and 'Find Next File' functions are redirected to the same routine. This calls the original DOS function and then examines the returned FCB (File Control Block). If it is not an extended FCB nothing further is done. However if it **is** an extended FCB, the seconds field of the time stamp of the file is checked for the value of 62. If this 62 seconds marker is present, the file is assumed to be infected and the length of the virus is subtracted from the length read.

Issuing the DIR command while the virus is active in memory will result in the original length of the file being returned rather than the extended length of the infected file. This is a typical primitive 'stealth' feature which is becoming more common in virus code.

The Second Destructive Routine

The 'open file' routine contains the second destructive part of this virus. Like the INT 16H routine, this monitors a counter installed by the virus when it was first loaded into memory. If the year is 1990 this counter is set to 16, otherwise it is set to 6. This counter is decremented every time a file is infected, once the counter reaches 0, every file that is subsequently opened using the DOS 'open file' command (INT 21H Fn 3DH) is deleted.

Infection Processes

The 'Load & Execute' routine handles the infection processes very carefully. When this function is called, the virus first checks that the available space on the disk is sufficient to store the infected file and aborts the infection process if it is not.

The last character of the filename extension is checked next: if it is 'M', the file is assumed to be a .COM file, otherwise it is assumed to be an .EXE file. The virus does not infect COMMAND.COM. The file is then opened and if the last five bytes are 'PcDos' the file is assumed to be infected and is ignored. The virus is then copied in memory.

In the case of .COM files, the file is read into memory immediately after this copy of the virus and the string 'PcDos' is added to the end of file.

A random key is then obtained from the timer, the virus is encrypted and the whole infected file is written to disk. The original date and time stamp of the file are subsequently restored, although the seconds field is set to 62.

Finally the counter for the open file routine is decemented and control is passed to the normal 'Load & Execute' routine. The process is identical for .EXE files except that the virus is written after the original file and the file header is altered so that the virus executes first.

Damage Maximisation

This virus was clearly written with the aim of maximising the damage on as many systems as possible.

It was written in 1990 and during that year the format function, which would immediately be noticed by users, was completely disabled. **In fact the first day on which this virus will trigger is 23rd September 1991**.

The routine to delete files is set so that during 1990 the initial counter of the number of files to be infected before deletion commences is set to 16 rather than 6 as it is in any other year. This 'delay' meant that the file deletion INT 16H routine was never activated during 1990. The developer clearly wanted to reduce the chance of the virus being detected during 1990. He may well have succeeded in this objective as this virus has only now come to our attention.

Detection and Disinfection

Despite the fact that the bulk of the virus is routinely encrypted, detecting it is relatively straightforward. Although this virus employs no sophisticated 'stealth' mechanisms, detection (as with all virus code) should be undertaken in a clean DOS environment.

The following search pattern will detect this virus:

```
33DB BE1C 00B9 4F07 2E8A 9708 002E 0010
```

This virus is in the wild, is very destructive and is set to trigger in September of this year; three facts which make its early detection highly desirable. Commercial software will doubtless be updated quickly to combat this virus. In the meantime, the addition of this pattern to the updatable library facility of a virus scanner such as IBM's *VIRSCAN* is recommended.

The simplest and safest disinfection method for all parasitic viruses is simply to overwrite and then delete infected files. The system can be restored from clean write-protected copies of the original master software.

To my knowledge no commercial scanner has yet been updated to locate this virus and no automated disinfection routines for this virus are yet available.

PRODUCT REVIEW

Dr. Keith Jackson

VISCAN

VISCAN is a virus scanner program for IBM-PCs. It has the distinction of being the first scanner program that I have come across which actually recommends that it is used in conjunction with other scanner programs to cross check any detected viruses. As I have laboured this point ad nauseum in many reviews of anti-virus products over the last few years, I'm pleased to at last find a scanner program actually recommending such a tactic. Given that the author of any virus scanning program will have his own methods (and contacts) for obtaining virus patterns, and his own methods of carrying out a scan, diversity should ultimately bring increased confidence in the level of virus detection.

This review is the first to use the newly extended *Virus Bulletin* set of virus test samples, containing more than twice as many viruses as the previous test-set. This point should be borne in mind when comparing *VISCAN* with any previously reviewed scanner programs. [*See footnote, p. 27*]

Documentation

The documentation that accompanies *VISCAN* is contained in a small booklet, and is also provided on disk as a README file. If you are looking for a voluminous explanation of the minutiae of how to operate *VISCAN* under all possible circumstances, then you will be disappointed, as only a small booklet is provided. Having said that, it contains clear and correct explanations of how to use *VISCAN* and how to approach the problems of computer viruses in general. What more can one really ask of documentation than to be accessible and correct? The documentation which is 13 pages long contains no index or table of contents.

The first and last lines of the *VISCAN* documentation are identical: "The best protection against virus activity is REGULAR VERIFIED BACKUPS!!!". I could not agree more; such action removes the questionable need for inoculation, disinfection, or any other such dubious practices. If your computer becomes infected by a virus then the best course of action is to erase the infected files and replace them with non-infected copies of the original. If backups are not available then such actions are impossible.

Appropriate emphasis is placed by the *VISCAN* documentation on having a clean write-protected system floppy disk from which the computer can be booted. Instructions are provided to create such a disk. In summary, the advice contained within the *VISCAN* documentation is refreshingly simple and clear.

VISCAN stores all of its virus patterns in a single file and before each scan verifies that this file has not been tampered with. This precaution prevents other programs, including a virus targeted at a particular scanner program, from altering the patterns. There were 357 virus patterns and identities in the version of *VISCAN* tested for this review. A library utility is provided which permits the addition of new patterns.

VISCAN first searches for viruses in memory. If a virus is found in memory, then a reboot from a virus-free system floppy disk is enforced before scanning can proceed. If the inspected disk is a hard disk, the Master Boot Sector and the specified DOS Boot Sector and drive are scanned. On floppy disks, the boot sector is scanned. Options are provided to disable these boot sectors scans if so desired.

VISCAN can inspect either a single file, the contents of a directory (and/or its subdirectories), or all files on a named disk. On request it will create a log file on disk which contains a complete description of all files scanned, and details of any virus found.

Scanning Speed

On my Toshiba Portable (see Technical Details below) *VISCAN* reported that it verified 357 virus patterns and identities before scanning memory, the Master Boot Sector, the DOS Boot Sector and 1318 files. All this took 4 minutes 26 seconds. This time was reduced to 1 minutes 34 seconds (while scanning only 310 files), when using an option which constrained the scanning process to *executable* files only (defined by *VISCAN* as those files having an extension of COM, EXE, BIN, SYS, APP, PGM, DLL, OVR, OVL or PIF). This is very fast indeed and unlike some other scanner programs that have been reviewed by *Virus Bulletin*, has not been achieved at the expense of having to completely disassemble every virus before a suitable scanning process can be implemented. The *VISCAN* documentation goes on record as stating that the virus patterns contained in *Virus Bulletin* are routinely used. For comparison purposes, *SWEEP* from *Sophos* (version 2.21) takes 4 minutes 43 seconds to scan the same hard disk, and *SCAN* from *McAfee Associates* (version 4.5B66) completes its scan in 4 minutes 19 seconds. The speed at which *VISCAN* can inspect a disk is very impressive.

VISCAN does not check the complete contents of each file for a virus infection. It uses the author's knowledge of where viruses reside to provide a fast scanning rate. The results of testing how accurately *VISCAN* can detect viruses (see section on detection rate), shows that this approach has not been detrimental to security. However, it is possible that a virus could be contained in an unusual part of a file (especially in the case of multiple infection, and/or partial disinfection), and *VISCAN* provides an option which can be used to check the *complete* contents of a named file against all known virus patterns and identities. This option will prove useful when a multiple virus infection of a single file is suspected.

Detection Rate

There are now 114 unique viruses in the *Virus Bulletin* test-set, and variants of the same virus extend the number of samples used for testing to 183 examples. This new test-set includes only infected COM and EXE files and two boot sector viruses. The test-set will be further extended by the addition of more boot sector viruses in the near future.

The content of this test-set has been dictated by the rapid increase in the number of known PC viruses. Note that the number of viruses in this new test-set is smaller than the complete list of PC viruses known to *Virus Bulletin*. As a deliberate policy I have chosen to omit virus samples that have given extraneous results in the past, or have spread confusion by being known in various contradictory guises.

VISCAN correctly detected all of the viruses in the new test-set. What more can one ask? Given the links between Jim Bates (the author of *VISCAN*) and *Virus Bulletin* and the stated use of *VB* patterns for scanning purposes, this result is not surprising, but it is refreshing to review a package that actually achieves such complete success.

Comments

VISCAN has no knowledge of executable files that have been dynamically compressed by programs such as *LZEXE* or *PKLITE*. Such programs are stored on disk in compressed form and decompressed when loaded into memory at execution time. If such a file were virus infected and then compressed the virus would not be visible to a scanner program. Dynamic decompression is beginning to be used routinely and some commercial software is now distributed in this form. Not only does it provide a smaller executable file (thus saving on disk storage), but in many cases the overall time taken to execute a file actually decreases as the time taken to decompress the program is outweighed by the reduction in disk loading time for a smaller size of disk file.

Even though it contained only four files, the *VISCAN* distribution disk was notable for having no empty space available for other files. This has been achieved by filling up the remaining disk space with one large hidden file, with the stated aim of ensuring that the files on the distribution disk (especially the virus pattern file) are not extended and/or altered. Given that you can copy the files from this disk, alter them at will, and replace them along with a different hidden file to fill up the available space, I would query the usefulness of such a tactic. Permanent write-protection of the distribution disk would achieve the stated aim in a better way.

At first sight the option to "Display general advice on infection detection" did not appear to work. I expected the advice to be displayed immediately, while in reality a scan was initiated. I was confused by this. However, it slowly dawned on me that advice is proffered only when a virus infection is found and not on a routine basis. I'm not complaining about the execution of this option but about the way in which it is described on the help screen. Maybe I'm just getting old.

As for the cost of *VISCAN*, I'll quote from the relevant part of the documentation: "Remember that *VISCAN* may be freely copied as long as you don't distribute it as part of a commercial transaction". What more can I say.

In conclusion, I found *VISCAN* to be a thoroughly reliable tool (if a touch expensive!). The documentation is not comprehensive, but the technical content of the software is quite simply excellent. Highly recommended.

Technical Details

Product: VISCAN

Developer: Bates Associates, 64 Welford Road, Wigston Magna, Leicester LE8 1SL, U.K., Tel 0533 883490

Availability: IBM PC/XT/AT, PS/2, or compatible running MS-DOS v2.00 or higher.

Version Evaluated: 3.03, dated February 1991.

Serial Number: None visible

Price: £20.00 (Updates £10.00). May be copied freely (see text).

Hardware Used: A Toshiba 3100SX laptop portable with a 16MHz 80386SX processor, one 3.5 inch (1.44M) floppy disk drive, and a 40Mbyte hard disk, running under MS-DOS v4.01. Also an Amstrad PPC640 with a V20 processor, and two 3.5 inch (720K) floppy disk drives, running under MD-DOS v3.30.

Virus Test-Set: This suite of 114 unique viruses and one Trojan (according to the naming convention employed by VB), spread across 183 individual virus samples, is the standard *VB* test-set. It comprises two boot 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. Where more than one variant of a virus is available, the number of examples is shown in brackets.

1049, 1260, Twelve Tricks, 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, Brain, 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, Italian, 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.

(*Editor's note*: The test-set for this month's comparative review (p. 8) differs from the standard *VB* test-set which appears above and was assembled by Dr. Keith Jackson. The comparative test-set was constructed 'in-house' from computer virus samples made available to *VB* in February 1991.)

END-NOTES & NEWS

The *Virus Bulletin* Conference

The *Virus Bulletin* Conference on Combating Computer Viruses, September 12-13th 1991, *Hotel de France*, St. Helier, Jersey. The final programme is now available from *VB*. Speakers include Fridrik Skulason (*University of Iceland*), Jim Bates (*Virus Information Service*, UK), Vesselin Bontchev (*Bulgarian Academy of Sciences*), David Ferbrache (*ISIS*, UK), Ross Greenberg (*Software Concepts Design*, USA), Dr. Jan Hruska (*Sophos*, UK), Jon Norstad (*North Western University*, USA), Yisrael Radai (*Hebrew University of Jerusalem*, Israel), Ken Van Wyk (*CERT*, USA), Prof. Gene Spafford (*Purdue University*, USA), Martin Samociuk (*Network Security Management*, UK), Dr. Simon Oxley (*Reuters*, UK), Mike Perryman (*Manufacturers Hanover Trust*, UK), Steve White (*IBM High Integrity Computing Laboratory*, USA) and Kent Anderson (*European Security Programme, Digital UK*). Presentations on DOS, disassembly, forensics, anti-virus tools, recovery, Macs, Unix, DECNet/VMS, mainframes and networks, probable developments, malicious programming, corrupt work practices, blackmail and extortion. Information and copies of the programme are availaible from Petra Duffield, *Virus Bulletin Conference*, UK. Tel 0235 531889.

VB Education, Training & Awareness Presentations

Education training and awareness are essential as part of an integrated campaign to minimise the threat of computer viruses and malicious software.

Virus Bulletin has prepared a presentation designed to inform users and/or line management about this threat and the measures necessary to minimise it. The standard presentation consists of a lecture of one hour supported by 35mm slides, followed by a question and answer session. Throughout the presentation, technical jargon will be kept to a minimum and key concepts will be explained in accurate but easily understood language. However, a familiarity with basic MS-DOS functions is assumed. The presentation can be tailored to comply with individual company requirements and ranges from a basic introduction to the subject (suitable for relatively inexperienced users) to a more detailed examination of technical developments and available countemeasures (suitable for MIS departments).

The presentations are offered **free of charge** except for reimbursement of travel and any accommodation expenses incurred. Information is available from the editor, *Virus Bulletin*, UK. Tel 0235 555139.

Editor's note: The traditional 'End-Notes & News' section will reappear in May. Its absence is due to time limitations imposed by attendance at the *4th Computer Virus & Security Conference* which took place in New York last month. (A report on this event will appear in the next edition.)

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including delivery:

USA (first class airmail) US$350, Rest of the World (first class airmail) £195

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