May 1992 ISSN 0956-9979

VIRUS BULLETIN

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

Editor: **Edward Wilding**

Technical Editor: **Fridrik Skulason**

Editorial Advisors: **Jim Bates**, Bates Associates, UK, **David Ferbrache**, Defence Research Agency, UK, **Christoph Fischer**, University of Karlsruhe, Germany, **Ray Glath**, RG Software Inc., USA, **Hans Gliss,** Datenschutz Berater, West Germany, **Ross M. Greenberg**, Software Concepts Design, USA, **Dr. Harold Joseph Highland,** Compulit, USA, **Dr. Jan Hruska**, Sophos, UK, **Dr. Keith Jackson**, Walsham Contracts, UK, **Owen Keane**, Barrister, UK, **John Laws,** Defence Research Agency, UK, **David T. Lindsay**, Digital Equipment Corporation, UK, **Yisrael Radai,** Hebrew University of Jerusalem, Israel, **Martin Samociuk**, Network Security Management, UK, **John Sherwood**, Sherwood Associates, UK, **Prof. Eugene Spafford**, Purdue University, USA, **Dr. Peter Tippett**, Certus International Corporation, USA, **Dr. Ken Wong**, PA Consulting Group, UK, **Ken van Wyk**, CERT, USA.

CONTENTS

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

EDITORIAL

A Gaping 'Ole

Windows 3.1 has entered the nation's subconscious and introduced the unnerving concept of the *OLE*. PC users emerging from winter hibernation to Latino cries of 'Olé, Olé' resounding around the ether, are probably bewildered - but no more so than most of the so-called 'experts'. With a menagerie of self-appointed *OLE* specialists already creeping out of the woodwork, it was felt that the acronym merited at least a mention in this month's edition, however ill-prepared we felt to write about it. When asked what *OLE* actually *does*, one jaundiced observer said: 'It's *Microsoft's* secret weapon - it turns PCs into Macs'. In truth, little is currently known about *OLEs* outside the hallowed corridors of *Microsoft* itself.

Microsoft, not content with bombarding the beleaguered user with PIF files and Dynamic Link Libraries, has now decided to put the cat well and truly among the pigeons by introducing *Object Linking and Embedding* - a process which effectively reduces the fine distinction between executable code and data. This has given rise to a degree of trepidation among the virologists who, as a community, always felt comfortable with the fundamental diktat that data files do not spread viruses.

There is currently a gaping hole in our knowledge; *Microsoft* has kindly supplied *VB* with spec-sheets and technical data but this amounts to several hundred pages of information which have still to be inwardly digested. A short report on the *actual* implications of *OLE* will appear in next month's *VB*.

In the meantime, it is wise to ignore the wilder speculations of the overnight 'experts', and refer directly to the best available source of information - *Microsoft* itself. An introduction to *Object Linking and Embedding* can be found in the *Microsoft Systems Journal* (March-April 1992, Volume 7, Number 2), while the *OLE.1.0 specification* is available on *Compuserve* in the MSL and MSOPSYS forums and on *Microsoft Online*.

Management Tips

The following report received from Peter Wigfield (aged 8) displays his admirable resolve in dealing with quack doctors:

> *'One day in March I was playing in the study and my computer started to speak. It said I have got the Michelangelo virus. That's funny, it's not March 6th. So we took it to the computer doctor. He said it was programmed for the wrong date. So he mended it. It took at least 5 hours to mend. Then we went home with some computer medicine. So we set it up, gave it a dose and one of my games was wiped off. So we went back to complain about it. So we fired his company and mended it* [ourselves]. *So we played on it and it worked once again and we were glad we sacked them.'*

TECHNICAL NOTES

Bombed Out

In recent months, investigators and computer security consultants in the UK have reported an increase in incidents of malpractice by contract programmers.

Programmers are often commissioned to design and implement systems for small companies which have little or no in-house computer expertise. These systems are generally written in a high level database language such as *dBase*, *Paradox* or *Advanced Revelation* which can be tokenised or compiled to increase speed of operation. Unscrupulous programmers occasionally include a 'sting in the tail', usually in the form of a logic bomb which is designed to trigger some weeks or months after the contract is completed. In most cases, the victim company has entered into a maintenance contract with the programmer. When such logic bombs trigger, they may, for example, encrypt all the company's data or simply destroy a vital component of the software. Whatever the effects, the company is technically dependent on the contract programmer to rectify the problem. The programmer is thus recalled, reverses the effects of the logic bomb, resets the trigger and leaves, waiting for it to go off once more and another lucrative telephone call from the unsuspecting company.

There are several steps the contracting companies can take to minimise the risk of this malpractice:

1. Insist that the project is developed entirely on your premises using some form of auto-logging software such as *PVCS* (*Polytron Version Control System*). Changes made to the software will show up in the audit log.
2. Before assuming responsibility for the software and signing any maintenance agreement, ensure that you have copies of all the source code and request that the programmer acknowledges in writing that the software and its source code is his handiwork. Ask the contractor to supply all the passwords he has used during the development, make sure they all work and then remove them.
3. Ensure you have a source code file for each of the compiled or tokenised object files.
4. Use a peer-review system to analyse the source code. Obvious logic bombs can be located easily. Look particularly for batch files being created under specific conditions - for example, a particular date being checked or an incrementing counter reaching a particular value.
5. If you are in any doubt, solicit assistance from the company which supplied the programming language; in the case of *dBase* and *Paradox* for example, this will be *Borland*.

It should be pointed out that unscrupulous contractors who introduce these devices are very much in the minority.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

Thoughts on Disinfection

Developers of anti-virus software disagree about methods of disinfection. Some place high emphasis on disinfection capabilities while others ignore the subject completely.

It should be noted that it is not possible to disinfect all files - overwriting viruses corrupt the contents of the infected file while some appending viruses make small but irreversible changes to a file when infecting them. As an example, a virus may change the 'checksum' field in the EXE-file header, without storing the original value anywhere.

An extreme approach is to provide no disinfection capability at all thus forcing the user to restore from software master disks or backups. This approach guarantees that files are never improperly disinfected, but may prove impractical as backups may be out of date or non-existent, or the number of infected files may be too high to undertake manual disinfection.

With boot sector viruses, the situation is different. Several anti-virus programs provide the option to create a 'recovery' disk, which restores the DOS Boot Sector, the Master Boot Sector and even the original contents of the CMOS. Other approaches include specific disinfection - the original boot sector is recovered from the sector where the virus stored it.

There is a significant difference between disinfection of boot sector viruses and file viruses. When a program is disinfected it should be restored to its original form, but the restored boot sector does not have to be exactly identical to the original one - it is sufficient that it be *functionally* identical.

Generic boot sector disinfection provides two benefits. The user can remove 'new' viruses and software does not have to be updated. Also, viruses such as Azusa (*VB*, April 1991, p.3), which do not store the original Master Boot Sector at all, can be removed.

Two viruses have appeared recently which complicate boot sector disinfection: the Monkey viruses store the original boot sector in encrypted form (a simple XOR operation is used to scramble the code), so it is no longer sufficient simply to move one sector from one location to another. If more such viruses appear, many anti-virus products may abandon 'virus-specific' disinfection and opt for the generic approach of replacing the infected boot sector with standard 'clean' code.

The best protection against all forms of virus activity is regular verified backups.

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

Backups of these vital sectors can be taken using disk editing software such as *The Norton Utilities*

Virus Collection 'Quality Control'

Maintaining a virus collection is a 'non-trivial' task. Researchers have little or no control over the quality of incoming 'material' - sometimes several disks full of new (and suspected) viruses arrive in a single day. Of course, some virus 'suppliers' are more careful than others, but every sample must be analysed, regardless of whether it came from another researcher, a worried user, or an anonymous individual who downloaded the files from a virus exchange bulletin board.

It is a common practice that files included in a virus 'collection' are *assumed* to be viral, regardless of their actual operation, and automatically merged into a larger collection. This frequently leads to the inclusion of 'garbage' files, which may be just about anything, including:

- ➤ Text files with COM/EXE extensions.
- ➤ Viruses which have been partially overwritten with text.
- ➤ Viruses where the 8th bit has been stripped away.
- ➤ Researcher's 'dummy' or 'goat' files.
- ➤ Tools, such as *DEBUG* and *TELEDISK*

Perhaps the most curious example ever found in a virus 'collection' was a large file of 'null' bytes, followed by a single CD 20H. As the collection containing this file was used to 'certify' scanners (for a fee), one must question the qualifications of the person(s) maintaining that collection.

Below is a set of simple guidelines for anybody who maintains a collection of viruses - a list of things to do when one receives a file containing a 'new' virus sample.

1) Ensure that the file is not identical to one of the existing files in the collection by checking other files with the same size and comparing the program with them using DOS *COMP*. Duplicate files often occur in virus collections.

2) Make sure the file is really a virus - this can only be done by infecting other files. 'Second generation' files must replicate as well. Should they fail to do so, the 'virus' has no business being in the collection - it may be moved to a special 'garbage' category, re-classified as a 'wannabe-virus' or even a Trojan. With around two new 'real' viruses appearing every day, those which don't work merit little effort. After all, if they don't replicate, no user will ever get infected by them - not until the 'virus' is 'fixed', at least.

3) Check whether the virus is identical to an existing one. It is not sufficient to run a scanner on the file and see what it reports; only a handful of scanners perform exact identification. Also, the original sample may have been modified in some way - for example it might be compressed with *LZEXE* or inoculated against infection by a specific virus such as Jerusalem. The scanner should only be run against the 'second generation' copy, not the original sample.

Only if the 'new' virus sample passes all three tests should it be merged with the primary collection.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

It's Over!

It's official; the computer virus threat is over.

UK firm *Viru-Tek* (200 Laurel Road, Wrens Nest, Dudley, West Midlands, Tel 0384 236165) has at last solved the problem with its HX-008 *Virus Proofing Card*:

> *WE GUARANTEE TO DETECT ANY AND ALL VIRUSES, KNOWN OR UNKNOWN NO MATTER WHAT YOUR ANTI-VIRUS SOFTWARE CAN OR CANNOT SEE!!! Installed ONCE only. NO command procedure to activate (NO RELIANCE ON YOUR STAFF TO CHECK CONTINUALLY) Gives Fulltime, Fully Automatic protection, from bootup (NO MANUAL IMPLEMENTATION ACTION NEEDED).*
>
> *We GUARANTEE that no attempt to replicate will be allowed, and that even the 'stack space' used by 'Stealth Viruses' is checked, so don't rely on CRC checksum routines, for viruses can infiltrate those spaces becoming resident without them knowing! A/V software does get infected from time to time. THIS IS HARDWARE, thus totally immune!'*

In the light of this unexpected breakthrough, *VB's* editorial team would be pleased to receive suggestions as to possible new editorial directions or, indeed, suitable job offers.

Virus Prevalence Table - March 1992

Incidents reported to *VB* in the UK during March 1992.

Virus	Incidents	Total Reports
New Zealand 2	23	26.7%
Form	18	20.9%
Cascade	7	8.1%
Michelangelo	7	8.1%
Eddie 2	4	4.6%
1575	4	4.6%
NoInt	4	4.6%
Italian	3	3.5%
Tequila	3	3.5%
Spanish Telecom	3	3.5%
Flip	2	2.3%
Maltese Amoeba	1	1.2%
Jabberwocky	1	1.2%
Slow	1	1.2%
Vienna 2	1	1.2%
Yankee	1	1.2%
Joshi	1	1.2%
Captain Trips	1	1.2%
Brain	1	1.2%
Total	86	100%

DIRTY MACS

CODE 252

A new virus designated CODE 252 was discovered on Apple Macintosh systems on April 17. The virus is designed to trigger when an infected application is run or system booted between June 6 and December 31, inclusive. When triggered, the virus invokes a dialog box with the message:

```
You have a virus.
Ha Ha Ha Ha Ha Ha Ha
Now erasing all disks...
Ha Ha Ha Ha Ha Ha Ha
P.S. Have a nice day.
Ha Ha Ha Ha Ha Ha Ha
(Click to continue...)
```

No files or directories are deleted; however, a user may power-down the system upon seeing the message. The virus may damage certain applications. Under *System 7*, the System file can be damaged by the virus. Infected machines may refuse to boot or may crash.

Between January 1 and June 5 (inclusive), the virus simply spreads from applications to system files and then on to other application files. The virus does not spread to other applications under *MultiFinder* on *System 6.x* machines, nor will it spread under *System 7*. However, it will run on those systems if an infected application is executed.

Defensive Software

Gatekeeper and *SAM Intercept* (in advanced and custom mode) are effective against the virus. Either program will generate an alert when the virus attempts to replicate. The *Virex* record/scan feature also detects the virus.

Authors of all Macintosh anti-virus tools are planning updates in order to detect and remove the virus. *SAM* version 3.0.8 (BBS 408-973-9598) was released on April 17 and *Virex INIT* version 3.8 (BBS 919 419 1602) will also detect CODE 252.

Disinfectant (courtesy of John Norstad and *Northwestern University*, Illinois) and *Gatekeeper* (courtesy of Chris Johnson) are available from the usual archives including *ftp.acns.nwu.edu*, *AppleLink*, *America Online*, *rascal.ics.utexas.edu*, *comp.binaries.mac*, *Compuserve*, *Genie*, *MacNet*, *Delphi*, *sumex-aim.stanford.edu* etc.

Virus Detective version 5.0.4 has been released. Registered users will receive a mailing with a search string to detect CODE 252. The search string is:

```
Resource Start & Size < 1200 & WData 2F2C#23F3C
#2A9A0*3F3C#24878#2A9AB; For finding CODE 252 in
System.
```

Acknowledgements to Professor Eugene Spafford and David Ferbrache.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

HEAD-ROLLS

McAfee Quits In Evaluation Row

The sudden and unexpected announcement that anti-virus software supremo John McAfee had parted company with the *NCSA* and its fledgling *Anti-Virus Product Developers Committee* has caused an irreparable rift in the world of anti-virus politics. According to a report by Joshua Quittner in *Newsday* (a major New York newspaper), Dr. David Stang, former research director of the *National Computer Security Association* fell out with McAfee over scanner evaluations. Stang says that McAfee influenced which software packages were evaluated and weighted the virus test-set in his favour.

Based in Washington DC, the *NCSA* prides itself on its independent 'certification' scheme in which scanners are tested against the *NCSA* virus library to assess detection rates. According to Stang 'It was his [McAfee's] idea that we certify products'. Many anti-virus developers were reticent to send products for evaluation; the fact that *NCSA* charged $500 per evaluation did little to encourage submission. In the wake of this industry indifference, Stang claims that McAfee ''sent me the products and the check and said 'go certify'''. McAfee is understood to have spent thousands of dollars certifying his competitors' products.

NCSA results in February 1992 produced a 99 percent detection rating for McAfee's *Scan* putting it in the lead by 11 percent over the nearest contender (*Certus NOVI* 1.01). Quittner's report states that the test was financed by McAfee and the products (*Central Point*, *Certus*, *Fifth Generation*, *Norton*, *Xtree* and *McAfee Associates*) were picked by him.

```
From: John McAfee
March 9, 1992

NCSA Members:

McAfee Associates is resigning from the NCSA
and from the AVPD [Anti-Virus Product Develop-
ers] commitee effective March 9, 1992.

McAfee Associates feels that the organization
as it currently stands is moving in directions
that are not in the best interests of the
computer using public, the PC software and
hardware industry, or the anti-virus product
developers.

We would like our name to be removed from any
publications, statements or opinions that may
imply McAfee Associates approval, co-operation
or consensus with NCSA goals and directions.

Sincerely,
John D. McAfee
```

Notice of resignation. McAfee's disappearance from the *AVPD* has caused rifts, not least within the *NCSA*.

McAfee is quoted as saying 'I've got 75 competitors. I pick the ones who are going to give me the most trouble that month'. Certain scanners which performed well in previous *NCSA* tests were absent from the test; *F-PROT*, *SWEEP* and *FindVirus* have outperformed *SCAN* in previous evaluations - these products were not included in the February evaluation. This caused speculation that McAfee had gained disproportionate influence over *NCSA* testing. Stang claims that he agreed to expand the virus test-suite using viruses supplied by McAfee which led to a dramatic increase in *SCAN's* ratings.

The report quotes McAfee as saying: 'If your product competes with mine, I'd like for those customers of mine to know that your product isn't as good as mine.' He defends the use of the virus test-suite as supplied by *McAfee Associates*, 'You can't jimmy these scores. You can't say that McAfee buys more certifications, therefore he'll get a better score, because other vendors would complain.' *NCSA* results were used to promote McAfee's products - a much criticised series of adverts from his UK dealer, *IDS Ltd*, has used *NCSA* data to vindicate *SCAN* as the premier product.

If accurate this story does little credit to the *NCSA*. Evaluation bodies which demand payment are vulnerable to abuse. Such evaluations favour companies with financial muscle; smaller companies may not be able to afford the evaluation fees - a monthly evaluation totals $6,000 per annum under the *NCSA* scheme. Those companies with clout may select or exclude competitive products at will. It appears that *NCSA* test results, if not entirely discredited, are of questionable validity - in this case a single manufacturer appears to have supplied the virus test-suite *and* dictated which products should be evaluated!

Stang Leaves NCSA

Dr. David Stang has left the *National Computer Security Association* to form the *International Computer Security Association* (*ICSA*). Executive Director of *NCSA*, Robert Bales, states that *NCSA* plans to issue scanner certifications. *ICSA* under Stang will run a 'Virus Research Center'.

Hoffman Axed

A reformed *NCSA AVPD* met in Portland, Oregon, on 24th April. *Central Point*, *Certus*, *XTree*, *Fifth Generation Systems* and *Symantec* were present. Alan Solomon of *S&S* 'attended' by telephone while John McAfee was present in a 'non-voting capacity'. Surprisingly, no representative of *NCSA* itself was available to chair the meeting! Franchised attendees voted to remove Patricia Hoffman as custodian of the *NCSA* test virus library. Other members of the *AVPD* were not consulted.

Hoffman's April 1992 *VSUM* evaluation (released on the day of the meeting) placed McAfee's *SCAN* first with a detection rating of 95.1%, with *F-PROT* (91.3%) and *SWEEP* (89.3%) taking second and third places. In this, the first and last test conducted with Hoffman as *NCSA* librarian, fourteen scanners were evaluated against a library of 1,080 viruses. Bob Bales of *NCSA* is to replace Hoffman as test-set librarian.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

ENEMY ACTION

Virus Warfare

The Peach virus, described on page 17 provides an example of a new and disturbing trend - the development of viruses that are aware of specific anti-virus programs and which are able to evade them or even attack them directly. Some examples of this class of virus are Antimon (*Flushot+* and *Dr. Panda Utilities*), Itti (*Flushot+* and *Virex*), Peach (*CPAV*), Thursday 12th (*Flushot+* and *McAfee's SCAN/CLEAN*) and Zherkov (Lozinsky's *AIDSTEST*). This subversion has taken some anti-virus software developers by surprise with many high-profile products remaining highly vulnerable to such attack.

Listed here are 'The Five Golden Rules for Virus Warfare' from the perspective of the virus writer as opposed to the anti-virus programmer. (A note to anyone who would condemn publication of these rules as an incitement or a 'how-to' instruction list for virus authors - rest assured, they have already thought of, and used, these approaches.) It is sections of the anti-virus community which need to learn these rules.

'Stealth' techniques are effective against a wide range of anti-virus programs and as such 'stealth' cannot be classified as 'targeted subversion'; stealth tactics are not discussed here.

Rule 1 - Know Your Enemy

To evade a specific anti-virus product, the virus author must know that product. This is easy when the program is widely available either commercially or as shareware, but hardly worth the effort if the anti-virus program is under-subscribed or a custom-built 'in-house' project. The most prominent manufacturers are often the victims of their own success - with *SCAN* being used by a reported six million registered end-users worldwide, it is not surprising that *McAfee's* software is regularly Trojanised or that the virus writers persist in their attempts to evade detection by it.

Generally speaking, non-specific products are easier to target, because they are updated only rarely. The virus author is at an advantage here, because he has access to the current crop of anti-virus programs before he releases his virus.

```
Virus NOTES: It's here 'Sicilian Mob Ia' Virus.
And UNDETECTABLE as of the Sept & Oct 1991. This
Virus wuz an development from the Vienna Virus.
Though is VERY different, this one had to be
changed a HOLE fucken lot to make it UnDectable
from McAfee & Ass (Scan 7.2V80). And I made this
one a Signature Virus. Passing the word that NUKE
rules...
```

'NUKE's Master-programmer' playing hide and seek.

```
Detection: Possibly a scanner which finds
violator, not SCAN or CPAV (yet)
Removal: Delete infected files
This virus is a hack of the RABID Violator strain
C, which has been altered to avoid detection by
SCAN v85. Instead of killing all the drives, as
violator C does, this simply replicates, (infects
1 COM file each run) and on the 21st of every
month, displays a message after a file is
infected. The following text can be found in all
infected files:
"ReplicatoR", "Dec", "FiRe is a LAMER... destroy
InFiniTy"

BTW - this virus contains no destructive code,
and infects files over 400 bytes, and under
64,000 bytes in size. Also will not infect if
FluShot or Virex PC are active in memory.

By the way, no virus-scanner that I know of can
identify C-Virus. Of course, it's only a matter
of time, so be sure to change the signature, the
screwvirex[] array, and the code frequently.
Nothing can remove C-Virus either. Oh well.
```

The AntiFire virus as described on a Virus Exchange BBS. The virus has been 'hacked' in order to evade *SCAN*. Note also the references to *CPAV*, *Flushot+* and *Virex PC*. No prizes for guessing the purpose of the the 'screwvirex[] array'!

Rule 2 - Avoid Enemy Surveillance

The most simple method to evade a generic monitoring program, such as *Flushot+* is to detect whether it is active, and simply refrain from performing any 'suspicious' actions if the monitoring program is in memory.

Many anti-virus TSRs define a specific interrupt function, such as INT 21H AX=FF0F, to determine whether they are installed. This is effectively an 'Are You There?' call, a procedure which is well known to any self-respecting virus writer! Such functions can usually be determined by disassembling the beginning of the TSR or by analysing the interrupt it intercepts while it is active. Actually, it does not matter what method the TSR uses to determine whether it is active - the virus can simply use the same approach. Several viruses operate in this way - PSQR (1720) and Rage (see *VB*, October 1991, p. 21) are good examples. Ironically, users of anti-virus software which has been targeted by a specific-virus which uses this method are often immune to infection as long as the anti-virus software is resident in memory!

Another way to avoid early discovery is to refrain from infecting those anti-virus programs which use some sort of self-testing. However, even widely used self-test routines such as *PKZIP* authentication and CRC could be subverted by the determined attacker (see *Some Authentication Methods Used in Anti-Viral Software*, *VB*, May 1991, PP 11-13).

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

Rule 3 - Kill the Guards Before Attacking

Detecting that an anti-virus TSR is active is a good tactic to self-preservation. The next step is to disable (or bypass) the TSR, so that it will not produce an alarm when the virus infects a file. This is extremely easy to do if the TSR provides a method to temporarily deactivate itself. The TSR might be combined with a disinfection tool which must write to executable files; naturally such a tool requires that the TSR does not 'sound the alarm' while a file is being repaired in a legitimate way. However, this provides a loophole which the virus can exploit by deactivating the TSR just before it infects a file and then reactivating it.

If the TSR cannot easily be deactivated, it is probably easiest to patch it in memory. Remember that under DOS, with its wide-open architecture, any program can modify any other memory-resident software in any desired way. Patching anti-virus TSRs so that they do nothing, or at least nothing which the virus author considers discourteous, is almost invariably straightforward. Note that although the anti-virus program may perform extensive checking of its image on disk, it is unlikely to detect changes made to it in memory.

```
Strain A of the Itti-Bitty virus is 161 bytes in
length. It detects the presence of Virex-PC and
Flu-Shot+ by calling interrupt 21h with AX set to
FF0Fh. If either of these TSR virus-protectors
are loaded, then Itti-Bitty aborts.

If all files are infected, then as I stated
above, their C: disk is trashed and the computer
is locked up. No message, no fanfare, no nothing;
their just plain fucked. That's it. Only 161
bytes, too.
```

The Itti-Bitty virus evading detection by *Virex-PC* and *Flushot+*. In this instance, the virus simply aborts if either program is active and no harm is done.

Rule 4 - Seek and Destroy!

Any important data files that are a part of the anti-virus package can be tampered with if they are accessible - for example if they are stored on the hard disk. If a virus or Trojan can locate and modify a signature database for example, it could remove its own identification string - thus preventing that particular scanner from detecting it. It could also corrupt every signature - making the scanner unable to detect any viruses whatsoever. If the signatures are encrypted, or the file is protected with a checksum, it might still be possible to delete the database.

Signature files are not the only 'legitimate' targets. The Peach virus attacks a checksum file for example, while configuration files are obvious targets - in some cases they can be modified so the anti-virus program will be of no use at all.

```
Look (and look out) for these fine warez
by Nowhere Man:

** C-Virus – My first virus, the
program that proves that C CAN be used
to write good virii. Available now.

** Nowhere Utilities – A group of fine
utilities to assist you in the
development and distribution of trojans
and logic bombs. Also great for just
having around when you need them. Check
it out. Available January 1992.

** Super-Flu – The virus to redefine
virii. An appending virus written in
assembler, Super-Flu features Virex-
Protection(C), which disables any and
all TSR virus-detectors during
operation. Now no one is safe! Available
January 1992.
```

'Nowhere Man' sets up his stall...a tongue-in-cheek tribute to Peter Norton and a determination to subvert memory-resident anti-virus programs.

Rule 5 - Spread Doubt and Confusion

Even the best anti-virus program in the world is worthless if it is not used. A simple way to target software is to undermine user-confidence. A program's credibility can be impaired if users believe it is badly written, incompatible with other software or simply a nuisance to use. There are numerous ways to create a negative impression. If the virus goes resident before the anti-virus program, it can intercept the execution of it and generally do whatever it desires. Examples include:

- Crash the computer thus making it appear that the anti-virus program is to blame.
- Generate an error message, designed to reduce the user's confidence in the product, such as:

  ```
  Error in BASIC run-time library
  Stack overflow at 3428:7762
  ```

- Simulate the execution or installation of the anti-virus product, by producing the appropriate messages, without actually doing anything.
- Modify at random some bytes in the memory image of the anti-virus program and hope it will crash.
- Make programs crash, if the anti-virus program is installed.

Conclusion

All anti-virus programmers would be well advised to remember these rules during the conceptual stage of software design and with its implementation. It is generally agreed that any program can be subverted. However, it appears likely that only the widely circulated shareware products or high-profile commercial products will be subjected to the virus writers' scrutiny in the coming months.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 26th April 1992. Hexadecimal patterns may be used to detect the presence of the virus with a disk utility or preferably a dedicated scanner.

Type Codes

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

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

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

Seen Viruses

Cascade-1621 - CR: This short variant of the Cascade virus has been modified and the encryption routine has been changed, so most existing scanners fail to catch it.

```
Cascade-1621    FAE8 0000 5B81 EB07 0183 BF01 0100 740E 8DB7 2101 B934 0631
```

Chinese Fish - MR?: This boot sector virus has not been fully analysed, as only a part of the virus code (the boot sector) is available for disassembly.

```
Chinese Fish    C537 1E56 1653 BF2B 7CB9 0B00 FCAC 2680 3D00 7400 268A 058A
```

Creeper - CR: There seems to be some confusion regarding the 'Creeper' name, as various 'Creeper' viruses have been reported and their descriptions do not match at all. This one is 475 bytes long, and is found at the beginning of COM files.

```
Creeper         0E0E 071F C3CD 2050 2D00 4B74 2658 3DFF 4375 15A1 8A01 8BF0
```

Freew-692 - CN: When this virus activates (in 1993), it overwrites programs with a Trojan that simply displays the message 'Program terminated normally.' when run. The virus is 692 bytes long.

```
Freew-692       81F9 C907 7206 80FE 0175 0145 B41A BA03 01CD 21B2 00BE 0301
```

Harakiri - CEN: This 5488 byte High Level Language virus is not expected to become a real threat as it is much too obvious - it simply overwrites files when infecting.

```
Harakiri        5DC2 0400 052A 2E65 7865 015C 052A 2E63 6F6D 3659 6F75 7220
```

Horse Boot - MR: This virus infects the Master Boot Sector and stores the original on Track 0, Head 0, Sector 7, but on floppy disks it is kept on Track 39, Head 1, Sector 9.

```
Horse Boot      FC29 C08E D8BD 007C FA8E D08B E5FB 5055 A113 04FF 364E 00FF
```

Lovechild-B3 - MR: This virus is probably written by the author of the Lovechild virus, but its code is dissimilar. The virus is similar in many respects to the New Zealand virus.

```
Lovechild-B3    33C0 8EC0 B801 028B DC2E 803E 047D 0074 462E C606 047D 0090
```

Monkey - MR: Two new New Zealand derivatives, which store the original Master Boot Sector encrypted thus making disinfection more difficult than usual.

```
Monkey-1        48A3 1304 B106 D3E0 0420 8EC0 C356 8BFB BE20 0003 FEFC B9DC
Monkey-2        48BF 1404 4F89 05B1 06D3 E004 208E C0C3 8A34 B801 03E8 E1FF
```

Possessed-2443 - CER: This variant is very similar to the other two known variants, which are 2438 and 2446 bytes long, and detected with the pattern published in September 1991.

Rock Steady - CER: This 666 byte variant of the Diamond virus is extensively modified. 'Garbage' instructions have been added, probably to bypass scanners. The trigger-effect has changed; the virus now formats the hard disk on the 13th of any month.

```
Rock Steady     BF00 0150 5857 5058 AB50 58A4 95C3 EB1C 908C DA90 83C2 1090
```

Terminator - CR: A family of two viruses, 918 and 1501 bytes long. Both are very badly written, in particular the 918 byte variant which does not infect files properly but simply overwrites them.

```
Terminator-918  595B 58FB CF80 FC4B 740C 80FC 1174 0780 FC4E 7402 EB2B FA50
Terminator-1501 061E 9C8C CB8C D93B D974 03E9 6601 B400 B280 CD13 9D1F 075F
```

Troi - CR: A very simple, 322 byte virus, which does nothing but replicate.

```
Troi            0157 A5A4 C32A C0CF 9C80 FCFC 7504 B0A5 9DCF 80FC 4B74 03E9
```

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

TECHNICAL BRIEFING

Heuristics - The Way Ahead?

Sooner or later the ever-increasing numbers of viruses combined with the appearance of ever more convoluted encryption schemes will make it impractical to continue using traditional virus detection patterns in virus scanners due to run-time and disk space limitations. The developers of such programs are already hard-pressed to keep pace with developments and with the number of virus variants doubling every nine months or so, a sea-change now appears inevitable.

Many anti-virus producers have been looking at other detection methods, typically checksumming and other types of integrity checking program. However, these programs have one obvious drawback - they cannot prevent a virus from spreading or doing damage; they can only tell the user that his programs or data have been modified or corrupted. Anti-virus hardware may present a partial solution but the hardware approach is currently judged by many to be unreliable, impractical or too expensive to implement.

The Theory

Researchers can usually determine whether a suspicious program is a virus or not, quickly and with ease. Many have speculated whether that knowledge could be built into a program. Such a program would analyse a file and using a set of rules (heuristics), determine whether the file in question contains a virus or not. The theory is that such a program (or 'Expert System') would never, or rarely, need to be updated.

That theory is now being put into practice. Clumsy early efforts at heuristic analysis (such as the primitive public domain Trojan detector *CHK4BOMB*) have given way to more sophisticated systems. *FRISK Software* in Iceland has been systematically developing and refining a heuristic scanner; it is much used in the virus research community but presently causes an unacceptable percentage of false-positives. When scanning *VB's* 'in house' false-positive battery, *F-PROT's* heuristic scanner logged some 7 percent of innocent files as suspicious. *FRISK Software* goes to great lengths to point out that the software is experimental and even lists a number of programs known to cause false-positives including *Windows 3 Kernel*, *Microsoft Word*, and *IBMBIO*!

The Criteria

The ideal heuristic scanner (which must remain a pipe-dream) would fulfil two criteria:

1. It must never produce a false-positive - claiming a program is infected, when in fact it is not.
2. It must never produce a false-negative - claiming a program is not infected, when in fact it contains a virus.

It is easy to produce programs which satisfy one of these two criteria; pseudo-code for two such programs is given below:

Program 1 is guaranteed not to produce false-positives.

```
for every program X
    display 'X is not infected with a virus'
```

Program 2 is guaranteed not to produce false-negatives.

```
for every program X
    display 'X is infected with a virus'
```

Unfortunately, as Dr. Fred Cohen has often pointed out meeting both criteria at the same time is theoretically impossible. In fact, heuristic programs *assume* that they are attempting the impossible, and accept that they will occasionally produce a false-positive or a false-negative - the producers of such programs presumably hope the users will tolerate the false-positives and the false-negatives which may occur.

Today's Technology

The heuristic analysis programs in use today have an 80-95% detection rate - truly remarkable, as this exceeds the detection rate of some pattern-matching scanners. However, the reason they have not yet gained widespread usage is that the number of false-positives is still unacceptably high, ranging from 1 to 10%. Today these programs are useful tools in the hands of experienced users, but they should not be used at all by novices, where a false-positive might prompt the user to take a totally inappropriate action, such as formatting his hard disk.

If this technology is widely accepted, the level of false-positives may well place an unacceptable burden on each producer's technical support or indeed on the support departments of their principal competitors. A heuristic declaration that a program 'might contain a virus' or 'contains suspicious instructions' will necessitate examination and consequently waste skilled man-hours in fruitless analysis. Should this burden fall solely on the developer(s) of the heuristic scanner(s) it is probable that the software will be revised or even discontinued. However, sound anti-virus advice dictates that people use two or more scanners from different sources - the false-positive burden introduced by one manufacturer's heuristic scanner could thus tie up the technical support lines of dozens of other manufacturers as spurious reports are cross-checked. From any perspective, the factors in the heuristics 'equation' combine against the use of these programs in their current form - one critical factor is the penalty incurred due to the non-usage or destruction of legitimate programs which are flagged as suspicious.

At the moment heuristic analysis tools are not a replacement for traditional pattern-matching scanners. This may change as a result of intensifying R&D over the next few years. However, in the rush to replace traditional and increasingly obsolescent technology, manufacturers would be well advised to think very carefully before releasing an alternative with such a propensity to 'cry wolf'.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

BUSINESS CONTROLS

John Silltow
Systems Security
Woolwich Building Society

Virus Control Within the Woolwich

Over three years ago, we became aware of the virus threat. Not because we had been hit, but because there was a growing appreciation of the vulnerability of our microcomputer systems and their increasing importance to the *Woolwich Building Society*.

The view taken then is true today - a virus is a disaster and like any disaster, contingency plans can be developed to combat or control it. We therefore developed and implemented a Virus Contingency Plan.

Corporate Security Policy

The development of this plan, should be put into context within the *Woolwich's* general views on security.

Because the *Woolwich* is a financial institution, it has traditionally been more aware of the need for security than many other organisations. A Corporate Security Policy is already in place, therefore, and the Virus Contingency Plan fits neatly under it, alongside other related policies like the Third Party Policy which controls the access that non-*Woolwich* staff have to our systems.

Disciplinary procedures were also in place and understood, so there was little need to develop more than the essential detail of the plan.

Staff have been made aware of their role and responsibility under the plan. This is deliberately kept simple; users must inform the centralised Help Desk if they suspect their computer is not performing properly. Help Desk refers the problem to us (Systems Security Department) if they suspect virus contamination. Alternatively, users may notify us directly if they wish.

The Help Desk is given technical advice and support on the basics of virus detection, whereas the users are given only rudimentary information. This information is updated occasionally but we have to avoid the accusation of 'scaremongering.'

Disk Scanning

The backbone of the *Virus Contingency Plan* is a centralised virus-scanning process through which all disks moving around the *Woolwich* must pass before use.

The type of disks we would expect to scan contain:

- ➤ unsolicited software
- ➤ demonstration software
- ➤ educational and leisure software
- ➤ 'in house' development software
- ➤ software from unapproved suppliers

In addition, branches are advised against using disks that are not *Woolwich* property, unless specifically approved. Under this arrangement, we would expect all such disks to be submitted for scanning anyway.

Essentially we opted for a centralised scanning service based on the following premises:

- ➤ we have over 3,200 microcomputers and did not feel we could justify installing a scanning program on each one.
- ➤ even if we could afford the installation costs, we could not cope with the support and update overheads of a UK-wide network of around 600 sites.
- ➤ we do not trust the use of one scanning package only.

This last point is significant in two ways. Firstly, scanning packages always lag behind the virus writers and therefore are out-of-date even before release. The different products, as they are updated at different times, are continually leap-frogging each other in their ability to locate the latest viruses.

The second point is the belief by the user, that once a package is installed, it is good for evermore. We still come across people with version 1 of IBM's *Virscan* (released in October 1989) loaded on their PCs, who are convinced that it is still protecting their machine! This misplaced 'warm' feeling is one of the greatest problems we face.

"We still come across people with version 1 of IBM Virscan (released in October 1989) loaded on their PCs, who are convinced that it is still protecting their machine!"

For all these reasons, we have set up the central scanning machine with a minimum of three scanning packages in operation at any time. We also test and evaluate other packages, and at the moment we have seven scanning packages available and these are:

- ■ *VIS Utilities*
- ■ *Sophos Sweep*
- ■ *S&S Toolkit*
- ■ *Norton Anti-Virus* 1.5 and 2.0
- ■ *McAfee's SCAN*
- ■ *Central Point Anti-Virus*

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

Autoloader

Using multiple scans is time-consuming and we have no staff solely dedicated to this process. At present we are only achieving a maximum scanning rate of 70 disks a day.

We are therefore looking at installing an autoloader to speed the process up. This device has a hopper which is filled with disks and it then checks all the disks one after the other, with the scanning program. It drops the passes and failures into separate collection boxes. Depending on the contents of the disk and the length of the programs involved, scanning several hundred disks an hour is achievable.

Quarantine PC

Our 'quarantine' machine, which is located within the *Systems Security Department*, is a '386 with 3.5 and 5.25 inch drives. It has DOS 3.31 loaded at present although we have OS/2 on the shelf waiting for the day when viruses start to appear in that operating system!

The scans are run within a batch file and this monitors the returned error codes from each of the programs. This enables us to run the scans unattended, yet have a screen message summarising the results which the operator can then read. This screen message along with the date and time of the scan is recorded in a log file for future reference should a virus be found.

As a word of warning, trapping the error codes is not always easy. Some manuals do not provide the information and others give it incorrectly. It is therefore helpful to have some viruses on hand to test your procedures!

Once scanned, the disks are marked with a sticker which bears the date and a unique number. This information is stored in a database for later analysis and statistical reporting.

It should be appreciated that we are not saying that once scanned, the disk is guaranteed 100% virus-free, nor that it will remain so. We are saying that we *believe* the disk to be virus-free and we will therefore take the responsibility for that disk being used in a *Woolwich* machine.

The *Woolwich Building Society* has 3,200 computers at 600 sites across the UK. Purchasing anti-virus software for each and every PC was discounted as unviable due to support and maintenance costs. Instead, John Silltow and his *Systems Security* team opted for centralised scanning which has proved remarkably successful.

Software Approval Mechanism

A software approval mechanism has been introduced. This requires that all software proposed for purchase should either already be approved or go through the approval process.

One requirement is that all software is scanned and found to be virus-free. Enshrining this principle in a formal document is an excellent boost to the credibility and awareness of the need for vigilance about viruses.

Machine Scanning

Part of the successful approach to security within the corporate environment is to work *with* the business and to compromise where necessary. This means recognising that occasionally there will be a requirement for a business area to use untested software.

In these circumstances we allow such software to be used, but preferably on a standalone machine. We visit the site to undertake the scan using *VIS*, *Sweep* or *S&S Toolkit* depending on which has just been updated.

As part of our obligations to maintain software licensing standards (we were *FAST's* first corporate member), we undertake regular software audits on machines using our own software. Previously we scanned these disks on their return, but we have now purchased a number of copies of *VIS* to incorporate into our software. This will extend the audit time per machine, but will give us a regular opportunity to undertake a scan of every machine.

Other machine scans are undertaken on an *ad hoc* basis or where an infection is suspected.

We also believe it prudent to scan the PC and disks of anyone who has left 'under a cloud' for any reason.

Network Scanning

Most of our networks use *Novell NetWare* and, as yet, there has been no requirement to regularly scan them. We do, however, scan the networks when the need arises; all the scanning packages regularly used are network compatible.

Network scans take a long time, but usually the biggest hurdle is getting the supervisor password. It's at such times that security staff despair of security!

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

As more systems are linked into the networks, the need for regular scans will increase. We plan to run batch jobs in the silent hours to scan the servers remotely. Such activity will need to integrate, and not interfere, with other network operations such as centralised backups and system updates.

Communication Scanning

Apart from the insertion of a disk into a machine to pass an infection, it is possible to transmit virus code from one location to another using some form of host carrier.

Systems now being installed include direct communications between the *Woolwich* and its external trading partners. Such partners may be working with us in the traditional sense whereas others may be servicing the systems being used.

Outside organisations interfacing with our computer systems must adhere to our Third Party Policy. This dictates how they can use our systems and the penalties for improper use or abuse. In accordance with our security approach of having more than one line of defence, we are planning to monitor line transmissions for viruses. We are testing a number of anti-virus products for this role.

"We have a large number of other specialists whom we can use for additional support where required, including the staff of the Woolwich Centre for Computer Crime Research at Exeter University."

Home to Office

With the growth of home computing and the use of portables, we recognise that, albeit unwittingly, our staff transfer virus code between home and office.

We thus offer a counselling and support service which extends beyond the *Woolwich's* immediate boundaries. We believe that the virus problem is a common one which we all have to work together to overcome.

Subsidiaries

The *Woolwich* now has a number of subsidiaries including two located in Europe. The centralised scanning option is open to them all, but clearly in some cases it is impracticable. We are therefore working closely with them to ensure that the virus scanning systems they introduce are equivalent to ours. By this process, we can continue to exchange disks with them in relative safety.

New acquisitions or mergers require a major effort in education to instil in the new staff the same values which the *Woolwich* expects of its current staff. It is at such times of change that disaffected staff can become particularly active and deliberate attacks are to be expected.

Recovery

Internally, we have a two-man recovery team (of which I am one) and we respond to a call wherever it is. We have a large number of other specialists whom we can use for additional support where required, including the staff of the *Woolwich Centre for Computer Crime Research* at *Exeter University*.

The recovery team may, in certain cases, need to undertake actions which require a very high level of authority. Closing a business-critical network or searching personal belongings for disks are actions which fall into this category. Management should be aware of this before such teams are let loose!

Incidents

The level of incidents within the *Woolwich* has been far lower than I would have dared predict - so far we have had only four. Three have been picked up on the centralised scan and one (non-functioning) version of Michelangelo was found on a machine late last summer.

We have, however, found that while the number of incidents has been well below that expected, the amount of counselling and advice provided has been greater than envisaged. Users have seen alarmist media reports and some have believed that they or the organisation could be specifically targeted by a virus. Others report problems using phrases such as 'all the letters have fallen off the screen' when they mean that the screen has gone blank! In view of this, we no longer tackle the problem purely from the machine viewpoint. We take the threat seriously, but try to offer a more light-hearted approach when counselling users to help them accept and understand the situation better.

When a contaminated disk is received from an outside organisation, it is my responsibility to fax (or write) to them to explain the problem. I follow this up with a telephone call to discuss the situation to find out what happened, what they have done about the problem, and what other risks there may be. I also extend our counselling to them if this is the first indication they have of a problem.

Summary

This is the essence of the *Woolwich* approach to the virus problem. We have a number of processes in place to create layers of defence and reduce the likelihood of a virus getting to a machine, but should it do so, we believe we have the recovery techniques in place to minimise its impact.

Complacency is *not* an option and we will continue to monitor and refine our processes and procedures.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

VIRUS ANALYSIS 1

Jim Bates

The Vacsina Development Path

Reports continue to come in of PCs infected with one of the TP viruses (a.k.a. Vacsina/Yankee Doodle). At least seventeen of these arrived en masse from Bulgaria in early 1990 and they represent a a development path ranging from clumsy and primitive programs up to quite intricate code. Interest in the TP series stemmed from an accompanying report which purported to tell the story of their development.

The Storyteller

The Storyteller, a Bulgarian himself, claimed to know the author, whose initials he gave as TP. The story goes that TP became fascinated with viruses and wrote one (presumably TP01). He then became equally fascinated with ways in which viruses might be stopped and so he wrote an anti-virus program. He then refined his virus so that it could circumvent the anti-virus program and then improved the anti-virus program so that it could stop it ... and so on and so on. The anti-virus program took the form of a device driver with the name VACSINA (the Bulgarian for 'vaccine') and the developing virus code contained this name as its target.

An 'Upwardly Compatible' Series

While engaged in this circular method of product development, TP took great care (the Storyteller tells us) to ensure that his viruses were not destructive. They were developed to be 'upwardly compatible' such that a later version, upon encountering an earlier model, would first remove it and then re-infect with the new code, thereby avoiding any potentially destructive clashes. TP's later viruses (following this series) were described as 'much more clever' since they were able to 'hide' in all sorts of exotic places within the machine's memory - PC DOS I/O buffers, unused space of EXE headers and so on. They also became able to 'circumvent any memory-resident program that monitors INT 13H' and eventually became so clever and dangerous that even his close friend (the Storyteller) was forbidden access to the code.

Implausible Distribution

Thus far we have an everyday tale of virus folk happily going about their business. However, the Storyteller then proceeds to explain that the reason the viruses escaped to the outside world was not because TP spread them himself, but because they were left 'lurking' on his PC which was a shared machine. TP warned the other users of the presence of viruses on the machine but this was 'taken as a joke' and that is how the virus spread! At this point the story's credibility passes the breaking point - but true or not, this tale of foolishness did have a number of specimens of virus code accompanying it.

TP04 and TP46 Spread in Europe

The viruses are identified by TP numbers ranging from 4 to 46. Not all members of the series are included and only numbers 4 and 46 have been reported at large in Europe. The samples range in date from February 1989 to November 1990 and, somewhat strangely, they don't occur in order of development. The oldest is TP33 while the most recent is TP39.

With TP25, the author developed a musical penchant and included a routine to play the George M. Cohan tune 'Yankee Doodle' when the user pressed Ctrl-Alt-Del.

Description and Operation

The earliest version, TP04, is a resident parasitic virus which intercepts the DOS INT 21H service routine, infects during LOAD and EXECUTE requests and infects executable files regardless of their file extension.

A somewhat archaic process occurs when the virus attempts to infect a file which begins with the characteristic 'MZ' header of EXE files. In this case the virus only appends a small stub designed to relocate the host file in memory. In appending this code, the 'MZ' in the header is destroyed but the program usually continues to function correctly. At this point the host program does not contain the main virus code and is not infectious. The next time the virus tests this file, it will appear to be a COM file and that is when infection occurs.

A notable feature of this virus is that it 'beeps' as it infects each file.

The appended code is 1212 bytes long and only files between 1213 and 62854 bytes (inclusive) will become infected. TP05, TP06 and TP16 were much the same but did not have the beep sound effect.

Various minor changes were made to TP23 and TP24, including the removal of the 'VACSINA' string and an 'are you there?' call which was being used in detection software. With TP25, the author developed a musical penchant and included a routine to play the George M. Cohan tune 'Yankee Doodle' whenever the user pressed Ctrl-Alt-Del.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

Anti-Debugging and Hamming Code

The Storyteller has an attack of conscience about TP33 - 'At that time I pointed out to the author that even a non-destructive virus can be dangerous.' You or I might react to this by ceasing development. Not TP; he decides instead to protect his virus code against modification by using self-correcting (Hamming) code. We are told that this was the first virus to introduce armoured code to deactivate debuggers. TP34 continued the development theme in unambitious style until TP38 arrived where TP at last cracked the problem of EXE file infection in a single pass (without the relocation stub).

TP41 is yet another minor variant but a new direction appeared with TP42. This version reportedly checks for the existence of the Italian virus (Ping-Pong) and attempts to disable it. TP44 and TP45 are still awaiting full analysis and TP46 definitely targets the Italian boot sector virus. This is a discrepancy in the Storyteller's narrative, since he insists that TP46 'is able to fight the 1701/Cascade virus'. Analysis of TP46 shows that this is definitely not the case.

TP46 Targets The Italian Virus

TP46 is the most developed of the series and it contains quite sophisticated self-correcting code as well as code armouring (anti-debugging code).

The musical trigger is still present but the virus has no other trigger routine other than code to deactivate the Italian virus. This is achieved by generating a checksum from 97 words of code found at offset 12EH of high memory. If this checksum matches that known to TP46 as indicating the presence of the Italian virus, a routine is invoked which modifies certain data areas, inserts a specific jump instruction and then appends an extra routine which effectively makes the Italian virus start incrementing 'generation' numbers.

This modification only takes effect in memory and is not transferred to an already infected hard disk. However, subsequent copies of the Italian virus will carry a counter which deactivates the virus when it reaches 256. The starting point for the counter will be the generation number set by TP46 (i.e. first infection = 1, second infection = 2 and so on). Each infection will increment the resident counter (and thereby the starting counter for that new infection) until it rolls over to zero (256 increments from zero). Then the new code (inserted by TP46) disconnects the Italian virus from the system and frees the memory it was using. When the machine is rebooted, the process begins again. Note that once the Italian virus has been altered by TP46, it fails future checksum tests and is therefore not re-modified.

This sequence of events is at variance with the Storyteller's description of the anti-Italian virus which he names as TP42. He insists that this changes it 'in such a way that after 255 reboots it will kill itself and the only thing which will rest on the disk will be the 'dead body' of the virus.' The mistaken identification of the anti-virus code within TP46 is inexplicable; it contains specific addresses which are instantly recognisable as belonging to boot sector activity.

Straightfoward Detection and Removal

The range of TP viruses has not grown since the original group of fourteen were sent to the West.

Only TP04 and TP46 have been reported in the wild, but while there are internal differences, all the versions can be removed by deleting infected files and replacing them with known clean copies.

Vacsina

Virus Name - TP04

Aliases - Vacsina, TP04VIR

Type - Appending Parasitic on COM and EXE files (including COMMAND.COM)

Infective length - 1212 (infects files between 1213 and 62854 bytes)

Intercepts - INT 21H for infection route via 4B00H only

- INT 24h for internal error masking

Trigger - none

File recognition :

```
BABD 00B8 2425 CD21 0E1F BA14 00B4 0FCD 21B8
0043 8E5E 0E8B
```

Virus Name - TP46

Aliases - Vacsina, Yankee Doodle 46

Type - Appending Parasitic on COM and EXE files

Infective length - 2996 on COM files

- 2992 on EXE files

Intercepts - INT 01H for internal tracing/anti-debugging

- INT 03H for internal tracing/ anti-debugging
- INT 1CH for trigger timing
- INT 21H for infection route via 4B00H only

Trigger - Plays the tune 'Yankee Doodle' an indeterminate time after virus goes resident. Contains code to recognise and deactivate the Italian (Ping-Pong) virus. Non-destructive.

'Are You There?' Call - Place 4BFFH in AX register and issue an INT 21H request. The call returns with 55AAH in the DI register if virus is resident.

File recognition :

```
B8FF 4B33 FF06 CD21 0732 C081 FFAA 5575 02FE
C02E 8887 5F00
```

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

VIRUS ANALYSIS 2

Roger Riordan

MUMMY 1.2 - A Dangerous New Virus In Australia

We recently received a file which was reported to be infected with a non-standard variant of the common Keypress virus. When examined, it was found to be a 'sandwich', which had been infected with Keypress (a.k.a. St. Leonards), then with an apparently new virus, and then with Keypress again.

The virus was reported in the April 1992 issue of *VB* under the name Jerusalem-Mummy, but had not then been analysed. It has a quite destructive warhead, which will at best cause considerable inconvenience, and usually serious loss of data. The arming mechanism can go off at any time, and the virus gives no warning and causes no obvious symptoms.

Each copy of the virus contains a 'D-Day' counter. This is decremented whenever an already infected file is found. It is set to zero if an INT 21H, function 0FFH, with AL not equal to 0FFH is detected. This function call is used by Maltese Amoeba, Sunday, PSQR/1720, probably by other viruses and possibly by some legitimate programs.

All files opened with DOS function 3DH, or executed with DOS function 4BH, are checked. If the file has the initial bytes 'MZ' (signifying an EXE file) the file is assumed to be an EXE file. Anything else is assumed to be a COM file.

When the virus finds an EXE file, it looks for a companion file with the same name but with the extension SHW. If it finds this, the original file is treated as a COM file. The reason for this test is not obvious. Otherwise the virus checks the header to see whether the file is infected. If so the D-Day counter is decremented. If the file is not infected, and the counter is not zero, the file is infected and then executes normally.

The D-Day counter is checked whenever any file is opened, or loaded for execution. If it is zero, the warhead is detonated and rubbish is written over the first 63 logical sectors of the disk from which the file was loaded, using INT 26H. This will usually destroy the first copy of the File Allocation Table while on small disks it may also destroy the second copy of the FAT and possibly the root directory as well. Unlike Michelangelo, this virus does not overwrite much data. In most cases it would be possible to recover most of the data but this is likely to be difficult and expensive.

The intitial value of the D-Day counter is set by the first infected file loaded. The initial value is fixed for any given file, but each file infected will inherit the (lower) current value. It is possible for the counter to be reset (if an INT 21H, function FF is detected, and the next file opened is an infected EXE), but this is unlikely. Usually the hard disk will be overwritten as soon as the counter reaches zero.

In the sample analysed the counter was initially set to 155. Successive generations have rapidly decreasing values. As every EXE file opened is infected, and the counter is decremented each time an infected EXE file is opened for any reason (including scanning for viruses!), it could easily reach zero in a single day's computing. Thus the virus should be regarded as extremely dangerous.

The virus is a Jerusalem derivative, but has unusual tricks:

1. It starts with the sequence:

```
0003  MOV AX,CS:[000D] ;contains the code JMP AX
0007  PUSH AX
0008  MOV AX, 69
000B  JMP SP           ;I didn't know this was legal
000D  JMP AX
```

The net result is that control transfers to location 0069

2. Like the Jerusalem virus, it writes a three-byte transfer procedure to the Interrupt Table, and uses this to move the virus code to the start of the memory block. However, it writes this procedure to INT 3H, apparently in an attempt to make analysis more difficult.

3. The virus contains the encrypted message:

```
Mummy Version 1.2
Kaohsiung Senior School
Tzeng Jau Ming presents
Series Number = [XXXXX]
```

This is decrypted in memory but is not displayed (Kaohsiung is a major Taiwanese city).

4. The virus traps INT 21H, functions 3D, 4B, 52 and FF.

If function 52 is detected, the word at offset 1 in the virus' memory control block (the PSP segment address of the owner) is set to 0008 (presumably to hide the virus) before the call is passed to the previous DOS handler. If function FF is detected and AL is also FF, the current value of the D-Day counter is returned in AL. Otherwise it is set to zero and the call is passed on.

This is a different version from the one reported in *VB*. It increases the length of infected files by 1399 bytes, instead of 1489 bytes and is not found by the previously published pattern, as this contains an address which has been changed. However, it is found by the *VB* string for Jerusalem-Barcelona (March 1992). A supplementary string is published below:

```
2638 05E0 F98B D783 C203 B800 4B06 1F0E 07BB
2500 9C2E FF1E
```

Mummy 1.2 is detected by *VET* 6.85 and infected files can be repaired safely. The virus is detected if it is active in memory and is optionally disabled so that files can be repaired even if no clean system diskette is available.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

VIRUS ANALYSIS 3

Multiface - Striking Screen-Effects

The Multiface virus probably originated in Portugal and there are reports of it circulating in the United States. It is a parasitic virus which infects COM files by appending its code and modifying the initial program instructions to ensure that the virus code executes first.

Operation

Execution commences by testing a value stored at the beginning of the host program to see if is equal to 60. If it is, a flag is set before processing continues with an 'are you there?' call to the operating system. In this case the 'are you there?' call consists of placing 0FFCCH into the AX register and issuing an INT 13H function request. If the virus is resident, the request returns with 0FAFBH in the DI register. In this case, the code repairs the first five bytes of the host program image in memory and transfers control to it.

If the virus does not reply to the call, it modifies the size of the allocated memory block, searches the existing Memory Control Blocks before adding a new MCB (owned by DOS) to the chain. The virus is copied to this area (which is now a bona fide part of the operating system) and processing transfers to the resident code.

Initialisation consists of collecting and storing the interrupt handling addresses for INT 08H (User Timer Tick) INT 13H (BIOS Disk Handling) and INT 21H (DOS Function Requests). The method of collection in each case is via a normal function 35H (Get Vector) request to DOS. Once these addresses are collected and stored, the virus hooks-in its own handling routines to each vector. The INT 08H routine deals solely with the trigger effect, the INT 13H routine simply answers the original 'are you there?' call and the INT 21H routine intercepts only a 4B00H request (Load and Execute). The insertion of the new addresses is by direct memory access to avoid alerting simple monitoring software. Once initialisation is complete, the code repairs the host program image and transfers execution to it.

The INT 13H interception routine simply checks for the value 0FFCCH in the AX register and, if it finds it, places 0FAFBH into the DI register before issuing an IRET to terminate the request. Any other value in AX results in the request being passed to the normal handling routine.

The INT 08H routine, after saving the caller's register contents, checks a flag value to see whether any action is required. This flag is only set during the initial counter check, and the counter is only updated during the infection check. Once the flag is set, the routine completes a 'countdown' before enabling a 'dancing faces' effect for 14 seconds on and 14 seconds off. Efforts are made to avoid screen corruption, but there are mistakes in the code and the effect is poor. The INT 08H interrupt is serviced 18.1 times per second and since the toggling counter runs from 0 to 255, the real-time interval is 256/18.1 or 14.14 seconds. Once started, the effect is continuous except for being inhibited during file infection.

This virus infects solely during a Load and Execute request. The target file is checked for a COM extension and the disk is checked for available space, before the file is opened for Read/Write access. File attributes remain unchanged as do file date/time stamps, so files with a Read Only setting will not be infected. On suitable files, the first five bytes are read into the virus data area and replaced with a jump instruction to the virus code and the trigger counter (initialised at zero). The remainder of the virus, together with the data area, is appended to the file before the original function call continues.

The trigger counter is increased by one only when an already infected program is executed. This incrementation occurs until a value of 60 (3CH) is reached. The trigger thus executes when an infected program is run for the 60th time (at least). The delay before the distinctive screen-effect is seen is 4 hours and 36 minutes.

Comments

The virus is trivial, easily recognised and easy to remove. The only real claim to fame is its visual effect: TV people will love it. Television reports about computer viruses have invariably used library pictures of the Cascade virus and its hailstorm of falling letters - the Multiface screen effect will at least provide a welcome change!

Multiface Virus

Virus Name - Multiface

Aliases - none known

Type - Appending Parasitic on COM files only (including COMMAND.COM) Code becomes resident in newly created MCB.

Infective length - 1441 bytes

Intercepts - INT 08H for Trigger

INT 13H for 'Are you there?' call (AX=0FFCCH -> DI=>FAFBH)

INT 21H for infection route via 4B00H only

Trigger - Non-destructive visual effect of 'dancing faces'. Occurs 4 hours 36 minutes after 60th invocation of infected program file. Effect repeats at 28-second intervals. May corrupt screen contents.

File recognition :

```
1E02 01BE 0401 03F3 8BDE 2E80 3E04 013C 7506
2EC6 878A 0101
```

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

VIRUS ANALYSIS 4

Peach Virus Targets Central Point

As a rule, computer viruses are badly written and often convey the impression that they are the author's first attempts to write assembly language programs. Indeed, the programming of some viruses is so execrable that most analysts have occasionally felt the irrational desire to tear the diskettes containing some of the worst examples to pieces.

The Peach virus is a refreshing exception - the code is clear and easy to understand. It is a shame that somebody who obviously has a fair understanding of assembly language should choose to waste his time writing viruses.

Structure and Operation

Structurally and functionally, this virus is unremarkable - 887 bytes in length, it is a memory-resident COM (including COMMAND.COM) and EXE file infector, which infects programs on execution. The only unusual feature of the virus is the method it uses to subvert the *Central Point Anti-Virus* product (*CPAV*).

When run, the Peach virus first checks whether it is already resident. Unlike most viruses it does not define a special interrupt function for this purpose, but instead it stores the text string 'Roy' at memory location 0040H:00FCH.

This text is also located at the beginning of a longer message inside the virus - which may be the author's genuine address, or that of someone he wishes to harass or inconvenience (it may also, of course, be completely bogus).

```
Roy Cuatro
No 2 Peach Garden
Meyer Rd. Spore 1543
```

File Infection

If the virus is already resident, it determines whether the current program is structurally a COM or EXE file by looking at the second byte of the internal buffer which stores the beginning of the original program. If that byte is 'Z', the file is assumed to be an EXE file (virtually all EXE files start with 'MZ'). The virus then restores SS:SP and jumps to the original entry point.

If the file is structurally a COM file, the virus restores the first 19 bytes to their original values and jumps to address 100H in the current code segment - transferring control to the original program.

If the virus is not already memory-resident it allocates memory by manipulating the Memory Control Blocks and reducing the size of the last block. The virus then copies itself to the newly created hole and intercepts INT 21H, before returning control to the original program in the way previously described.

The INT 21H Routine

The INT 21H interception routine is straightforward - the virus only intercepts the Load/Execute function (4BH), which is called whenever the user runs a program. Peach then hooks into interrupts 23H and 24H. The reason for intercepting INT 24H (critical error handler) is obvious - otherwise the familiar 'Abort, Retry, Ignore ?' message would appear whenever the virus tried to infect a program on a write-protected diskette. The interception of INT 23H is quite unusual and is probably intended to prevent the user from aborting the infection routine by pressing Ctrl-C.

The virus increments an internal counter and checks whether it is equal to 1. If so, the string 'Roy' is stored at location 0040H:00FCH, to indicate that the virus is resident. Why this is not done at the same time as the virus goes memory-resident is not clear. If the counter has reached 27 the virus jumps into ROM at address FFFFH:0000H, which reboots the computer. This means that with the virus active, every 27th program run will cause a reboot, regardless of whether any other programs actually become infected. The virus also subverts *CPAV* at this time, as described later.

In order to determine whether to infect a program, the virus reads the first 24 bytes of it. If it appears to be an EXE file (again determined by checking whether the second byte is 'Z'), the virus checks the initial IP value. If it is equal to the one used by the virus, the program is assumed to be infected and is allowed to run normally. The virus next checks the initial SP value and does not infect the program if the value is equal to 7200H or 7600H. The virus presumably does this to avoid infecting an as yet unidentified program which uses self-checking code (the most obvious candidate would be McAfee's *SCAN* but this program has different intitial SP values, as do the EXE files in *Central Point Anti-Virus*).

At this point the virus checks whether its INT 21H routine has been modified starting with a PUSH AX instruction. The reason for doing this is not clear. The most probable explanation is that this checks for the presence of an INT 21H-monitoring program. If the PUSH AX instruction is not found, the virus proceeds with the infection by modifying the header to reflect the necessary changes and appending its code.

COM files are processed in a slightly different way. Files shorter than 512 bytes or longer than 64511 bytes are never infected. Otherwise the virus compares the beginning of the program to a 19-byte stub of code located at the beginning of infected files. The purpose of this code is to transfer control to the virus located at the end of the file. This is a more complex procedure than the more usual modification of a JMP instruction. If the blocks do not match, the program will be infected.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

Attacking CPAV

As mentioned before, the unusual feature of the Peach virus is its ability to attack the *Central Point Anti-Virus* package. This is done by exploiting a fundamental design flaw in this anti-virus program. When *CPAV* is run for the first time it creates a file (called CHKLIST.CPS) containing a checksum and other information for every executable file. On subsequent executions the scanner calculates the checksum for the file, and only if it does not match the stored checksum will the file actually be scanned for known viruses.

The *CPAV* authors took some precautions against one line of attack; by not publishing the details of the checksumming algorithm used [*if such exists. See page 27. Ed*.] they denied the attacker proprietary information which might enable him to infect a file, and then update the relevant checksum. Such subversion appears possible: all copies of *CPAV* seem to use the same, apparently trivial method to calculate the checksum. However, the Peach virus exploits a far more fundamental weakness within CPAV which its designers overlooked.

The Peach virus simply deletes file CHKLIST.CPS (the checksum file). It seems incredible, but this method actually *works* - the CPAV program recreates this file the next time it is run, calculating a new checksum for infected file(s) and fails to indicate a change to the integrity of the files. *Central Point Software* will presumably close this loophole in the next version; in the meantime, users of the package should be grateful that this virus has not yet been reported 'in the wild'.

PEACH VIRUS

Virus Name - Peach

Aliases - none known

Type - Appending Parasitic on COM files (including COMMAND.COM) between 512 bytes and 64511 bytes in length and EXE files. Code becomes resident in newly created MCB.

Infective length - 887 bytes

Intercepts - INT 23H for
INT 24H (Critical Error Handler)
INT 21H for infection route via 4BH only

Trigger - Every 27th program executed causes a reboot. The virus subverts checksumming method employed in Central Point Anti-Virus versions 1.00 through 1.20

System recognition via 'Are you there?' call which places text 'Roy' at memory location 0040:00FCH

File recognition :

```
33C9 33D2 E851 FFB4 40B9 1800 8BD7 807D 015A
7406 B913 00BA
```

PRODUCT REVIEW 1

Mark Hamilton

D-FENCE

'The manufacturer assures me it's been tested on a wide range of machines', the editor told me on the telephone before Easter, 'and it should uninstall cleanly. However, to be on the safe side I would back up everything on your hard disk!' 'Oh, great', I thought. 'It's one of *those* products'. It always amuses me when commissioning editors ask me to look at a security product with which they have little or no experience by offering such morsels of advice and reassurance.

Disk Authorisation

The product under discussion is *Sophos' D-FENCE* which is a disk authorisation package. *D-FENCE* is not strictly an anti-virus system, but it is designed to augment a corporate-wide anti-virus strategy because it actively prevents users copying from, or to, unauthorised floppy disks.

In common with *PC Armour* (*VB*, March 1992, pp. 19-21) *D-FENCE* is not, and does not claim to be, a 'high security' product: a technically competent and determined attacker could disable *D-FENCE.*

The *D-FENCE* manual points out the product's security limitations.

Envisaged Use

The proposed scenario is this: a corporate user installs *D-FENCE* on all its machines, except one or two which are known as 'gateway PCs' situated in the corporate security department or in Technical Support. All floppy disks entering the building have to pass through one of the 'gateway PCs' to be scanned for viruses and then converted in to the *D-FENCE* format. This must be done, because once a PC has been *D-FENCEd*, it cannot read or write non-*D-FENCE* diskettes. *D-FENCE* scrambles the Partition Table on *D-FENCE* disks in such a way that they are only intelligible to a PC which has been booted-up from a *D-FENCE* hard disk or diskette.

Presentation

The documentation and software is supplied in an A5, three ring, cloth covered binder. The 91 pages of *D-FENCE* documentation are accompanied by a perfect-bound volume entitled the *Data Security Reference Guide* - this publication provides a sound introduction to computer security principles; it must also be the only product catalogue in the world with an ISBN number! The software itself is provided on both 5.25 inch 360K disk and 3.5 inch 720 K disk.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

A Dry Run Through The Manual

I rarely read software documentation. Manuals tend to go straight on the shelf only to be dusted-off, opened and consulted if something goes awry. This is inadvisable with *D-FENCE* - its manual really **must** be read before you attempt any installation. The manual provides a straightforward, discussion of the product, its use and potential problems which might arise with particular attention to known conflicts with other memory-resident utilities. A full index and glossary are provided. The manual's prose is unremittingly 'dry'.

I decided against installing it on my Apricot Qi486 since this has a hardware access-control mechanism which is enabled and testing access-control software could be compromised. Besides, it also has 300 Mb of data which would take several hours and piles of floppy disks to back-up. I therefore opted to install *D-FENCE* on a 386SX-clone whose 50 Mb hard drive has little other than DOS and *Windows* installed upon it.

The manual says you should create a system boot disk for the PC before installing *D-FENCE* on it. This is vitally important as without a system disk you will not be able to uninstall the software. Having made a system disk, the user inserts the *D-FENCE* disk, logs onto its drive and types 'DFENCE'.

Options

It has a simple menu system similar to that used in other *Sophos* products such as *Sweep* and the *Sophos Utilities*. The menu offers six choices:

1. Install *D-FENCE* on PC. This is the option normally chosen to install the product, unless disk drives require a device driver to be loaded via CONFIG.SYS. For example, should you have drives attached to a *Future Domain* SCSI Controller card, you will also have to select option 5 to configure the relevant device driver.

2. Deinstall *D-FENCE* from PC. This is the converse of 1 and it replaces the *D-FENCE* Boot Sector with the original it has stored safely. Again, if you have drives accessed by device drivers, you will also need to uninstall *D-FENCE* from the relevant device drivers.

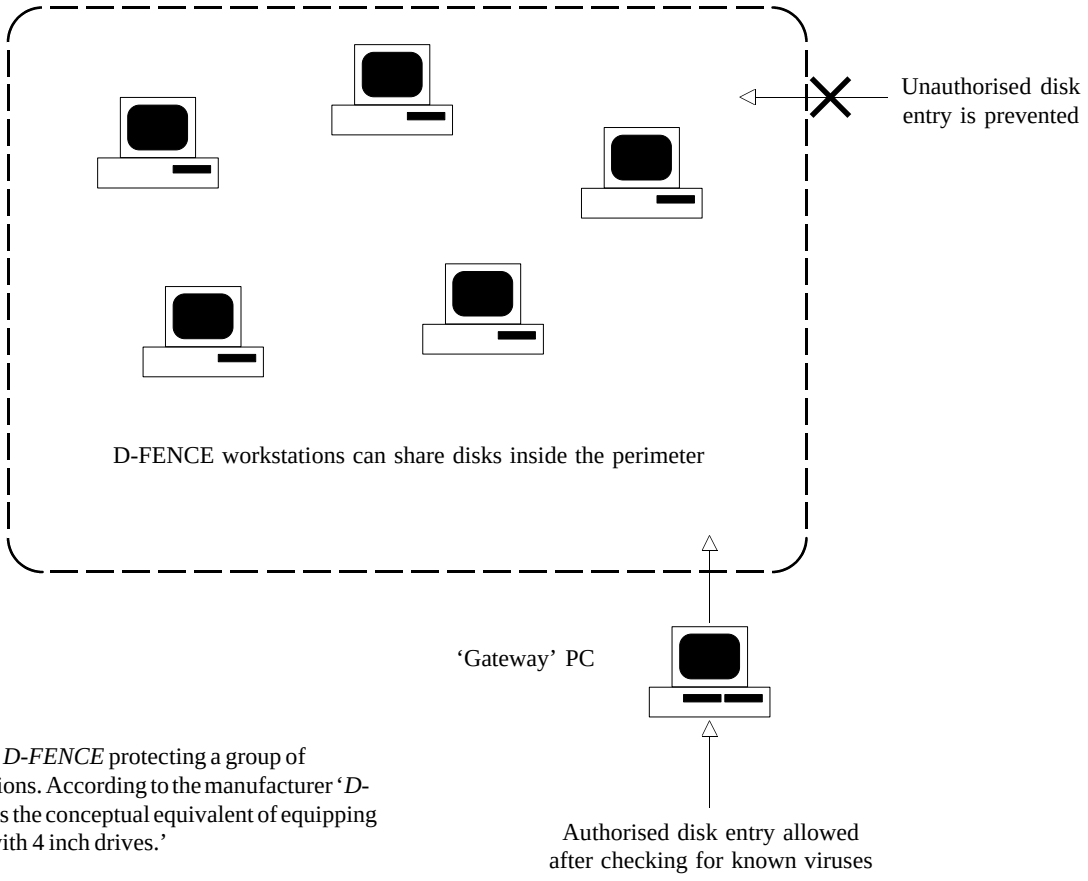


Figure 1. *D-FENCE* protecting a group of workstations. According to the manufacturer '*D-FENCE* is the conceptual equivalent of equipping all PCs with 4 inch drives.'

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

3. Install *D-FENCE* on floppy disk. This option converts a normal diskette to one that a *D-FENCE* PC can read, and is undertaken on a 'gateway' PC once the disk has been scanned for viruses. Once disks have been converted, they must be unconverted (using the next menu option) before they can be read on non-*D-FENCE* PCs.

4. Deinstall *D-FENCE* from floppy disk. If your PC at work is *D-FENCEd* and you also have a second, non-*D-FENCEd* PC at home, any disks you create for use on your work PC will have to be un-*D-FENCEd* before they can be used on your other PC. This menu option is provided for that purpose.

5. Device driver configuration. This option is used to install and uninstall *D-FENCE* from device drivers used to control non-standard devices such as external disk drives, Bernoulli drives and tape streamers. The manual clearly warns that this process overwrites the original device driver, so you may wish to ensure that you have a 'clean' version before proceeding with *D-FENCEing* it. As CD-ROMs are starting to become popular (which are accessed via device drivers); the *D-FENCE* manual does not explicitly state whether you should *D-FENCE* a CD-ROM device driver - or indeed if a CD-ROM drive is accessible on a *D-FENCE* PC. This is a serious omission which should be addressed.

6. Return to operating system.

Because my clone has no hard disks or external devices requiring device driver entries in CONFIG.SYS, I installed *D-FENCE* on it using option 1. I also took out an 'insurance policy' by making copies of the Master Boot Sector and DOS Boot sectors and backed-up 2 Megabytes of data files.

The manual states that the original boot sector is relocated and replaced with *D-FENCE* code. Ironically, this product works much like a boot sector virus. When you boot up, the *D-FENCE* boot sector is loaded into memory, it loads the remainder of its code which goes memory-resident before decrypting the original boot sector and loading the operating system. When you format a floppy, its root directory and FAT are scrambled using the same 'key' (the manufacturer uses the phrase 'ID') as is found on the hard disk.

'ID' Management

As delivered, *D-FENCE* has a standard 'ID' or code that it uses to perform its encryption and decryption. This means that disks can be read from and written to on any similarly protected PC - ones where the ID has not been changed. Should you wish to change the ID, *Sophos* provides a utility called SETID. This program records the new ID in the main *D-FENCE* program (DFENCE.EXE).

This arrangement has the basis of a major PC control capability making it possible to split an organisation into a number of work groups, each with its own *D-FENCE* ID. Anyone within a given work group would be able to read all floppies converted for use within that work group only, but no others.

Unfortunately, *Sophos* has decided to record the IDs within the *D-FENCE* program, rather than within an external configuration file. This means that a separate copy of *D-FENCE* has to exist on the gateway PCs for each and every work group and that the correct version is used to convert disks to and from the *D-FENCE* format. A more flexible approach would be to password-protect *D-FENCE* and let it convert disks to and from any ID it knows about.

Performance 'Under Fire'

D-FENCE certainly works. I could not read any floppies on the *D-FENCE* PC until they had been converted. Similarly, I couldn't read *D-FENCE* disks on the non-*D-FENCE* Apricot. In both cases, error messages suggested that the floppies hadn't been formatted. I also tried booting the *D-FENCE* machine using the system boot disk I'd created before installing *D-FENCE* - the hard disk simply 'disappeared'. Despite early reservations, *D-FENCE* uninstalled cleanly and efficiently - no need to cash in my 'insurance policy'!

There is, however, a downside to all this which proves that 'there's no gain without pain'. The pain in this case will be felt most accutely when installing new software. Manufacturers are increasingly producing software on permanently write-protected disks. Such disks cannot be read on a *D-FENCE* PC and the disks themselves cannot be converted. Thus by necessity, backups of the software must be made before they can be authorised. Problems will also arise with copy-protection mechanisms such as *CopyLock* and *SofGuard*. To install copy-protected software, the *D-FENCE* boot sector must be removed and the PC booted normally (*D-FENCE* is re-applied after the software is installed).

In summary, *D-FENCE*, while not actively preventing viruses, is a worthwhile front-line defence to reduce the risk of viruses and unauthorised software entering an organisation. One of its strengths is its transparency - the user remains oblivious to its presence until he tries to use an unauthorised disk. It will also reduce software and data theft. Bob Hay would approve...

Technical Details

Product: *D-FENCE* Version 2.01

Developer: *Sophos Limited*, 21 The Quadrant, Abingdon Science Park, Abingdon, Oxon OX14 3YS, UK. Tel 0235 559933, Fax 0235 559935.

Availability: IBM PC/XT/AT/PS2, compatibles and networks

Price: £19.50 per PC (minimum order of ten PCs). £15.60 per PC for site licences of 1,000+ PCs.

Modus Operandi: Encryption of the Partition Table on fixed disks. Encryption of FAT and root directory on all authorised floppy disks. Unauthorised disks cannot be read.

Key Specification: Built in at time of software manufacture. User configurable.

Memory Footprint: 2 Kbytes

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

PRODUCT REVIEW 2

Dr. Keith Jackson

Central Point Anti-Virus (V1.2)

Central Point Anti-Virus has been reviewed before by *VB* (v1.0, June 1991). This review was written by another contributor, so as far as I am concerned *Central Point Anti-Virus* is new. This review looks at the features which were criticised in the previous review, to see whether they have been rectified/modified, but it also looks at *Central Point Anti-Virus* in absolute terms, as one the more prominent anti-virus software packages on the market.

Setup and Documentation

The *Central Point Anti-Virus* documentation was described in the original review as being a 'professionally produced work'. Although professionally produced, much of it is somewhat uninspiring, comprising simple descriptions of the obvious points about each of the myriad menu options offered.

Detailed explanatory information is limited to the 'Trouble-shooting' Chapter, which is excellent and discusses potential problems in commendable detail. Information about specific viruses is prolific, nearly a third of the manual (45 pages out of 149 pages) is dedicated to a 'virus dictionary' which provides details about the viruses that *Central Point Anti-Virus* can detect: currently 1009, including variants.

Defence In Depth

Central Point Anti-Virus explains its philosophy succinctly. Users are advised to create an 'emergency' disk for use when a virus has slipped through *Central Point Anti-Virus*'s defences, use the utility provided to check the boot sector and partition table, immunise executable files, use one of the memory resident anti-virus programs to monitor for virus activity, scan each hard disk, each new floppy disk, and each file downloaded from a bulletin board. With the single exception of immunising files (see below), this advice is unimpeachable and reflects *Central Point's* well-known (and correct) strategy of providing muli-layered anti-virus defence.

Central Point Anti-Virus is very easy to install, with or without resorting to the manual. However, the Uninstall option did present minor difficulties. The README file states quite clearly that all files should be deleted when uninstall is run - this didn't happen when I installed to drive D:. Upon trying to re-install *Central Point Anti-Virus*, the installation refused to proceed, saying that not enough disk space was available. This was true, but the reason was that the *Central Point Anti-Virus* files were still there from the previous installation! The only way out of this impasse was to delete all of the *Central Point Anti-Virus* files manually.

Näive users may find this somewhat fraught. They may also be rather confused by the fact that *Central Point Anti-Virus* alters the date and time of the AUTOEXEC.BAT and CONFIG.SYS files even if the options are set so that installation does not alter the contents of these two files. Curious.

Integrated Shell

Central Point Anti-Virus provides one main program which provides virus detection, immunisation and removal. Other programs are included which provide memory-resident anti-virus protection, and guard the boot sector and partition table against virus infection.

All anti-virus features are controlled by the main program - the integrated shell, which operates in what is a fairly standard Windows-like manner. It is known as *CPAV*.

CPAV can operate either under MS-DOS or under *Windows*. It is not however a *Windows*-specific program. The testing for this review was carried out under DOS, although I did scan disks under *Windows* to verify the claim that *CPAV* operates correctly under *Windows* - it does.

I found the main program easy to use, navigating around was no problem whatsoever, given the standard program style of drop-down windows, buttons, and multiple windows. However, users would probably be rather discouraged by the long list of configuration options that are provided. The default choices are eminently sensible, but if they were not, näive users might well find the bland descriptions in the documentation (see above) less than useful.

Scanning

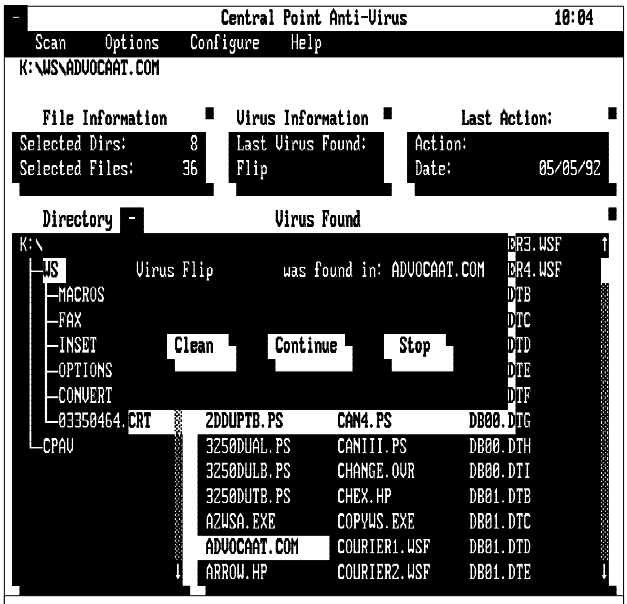
I tested the scanning capabilities provided by *CPAV*, by seeing how well it could detect viruses from the usual test sample (see *Technical Details* section), and how fast it could scan an entire hard disk. [*Readers should note that this test-set is now very dated. It is to be expanded next month. Ed*.]

From the 183 virus samples that were used (114 unique viruses), *CPAV* failed to detect three - Kamikaze, Lehigh and TUQ. Failure to detect the Lehigh virus was most surprising, as this is one of the oldest. This corresponds to a detection rate of between 97.4% and 98.4%, depending on whether the total number of virus samples, or the number of unique viruses was used to calculate this figure.

I used *CPAV* to scan my entire hard disk, both as a DOS program, and under *Windows*. The time taken to perform this task, along with comparative figures measured for the latest versions of *FINDVIRUS* from *S&S Enterprises*, and *SWEEP* from *Sophos* are shown in Figure 1.

The hard disk was scanned twice for each program, once when all files on the disk were searched, and once using whatever 'Turbo' mode was offered by each program.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.



The Flip virus is detected. This virus has in the past caused problems for *CPAV* - the wildcard search string used to detect it was stored unencrypted under version 1.1 giving rise to a spate of false-positives from other scanners. The problem appears to have been corrected under version 1.2

Note that scanning under *Windows* slightly impedes *CPAV* introducing roughly a 25% overhead. The results (Figure 1.) show that *CPAV* is not the fastest scanning program available but neither is it a slouch in these matters.

	Complete Scan	Turbo Scan
CPAV (DOS)	3 mins 22 secs	2 mins 25 secs
CPAV (Windows)	4 mins 10 secs	3 mins 00 secs
FINDVIRUS	1 min 24 secs	35 secs
SWEEP	4 mins 54 secs	59 secs

Figure 1. Some comparative scanning speeds

Questions of Integrity

The original *VB* review complained that although *Central Point Anti-Virus* claimed to monitor file integrity by a checksum process, this checksum did not seem to be calculated across the entire file. Indeed the algorithm used to calculate these checksums was thought to be 'minimal' as enabling or disabling verification of these checksums did not seem to make any difference to the time taken to scan a disk for viruses. At least the manual is now quite open on this point. It states that in each subdirectory there is 'a database of records, called checksums, about each executable file in the directory including information about each file's size, attributes, date, and time.' Note that there is no mention of a checksumming algorithm.

I tested this by calculating checksums for EXE files of 32K and 640K, and COM files of 2K and 49K, in an attempt to see how long checksum calculation would take. The actual measurements are not worth listing (a couple of seconds) as I could detect no difference between any of them. Although the manual does not state explicitly that no calculation is carried out across the content of the file, these simple tests show that this is very probably the case. It is just not possible to access 640K of data on a floppy disk in a few seconds, and even if that were possible, calculating a checksum over that amount of data takes a finite amount of time. Much longer than a few seconds.

Central Point Anti-Virus checksums are held in a database file (CHKLIST.CPS) which is created within every subdirectory that contains executable files. If the option to recalculate this database is enabled, *Central Point Anti-Virus* recalculates the 'checksum' file whenever it cannot be located. This leaves a specific target route by which viruses can attack.

In order to infect files and stand a good chance of remaining undetected, the attacker could simply erase the checksum database, infect files in that subdirectory, then either recalculate the correct checksums, or wait for the next *Central Point Anti-Virus* search of the hard disk to recreate the database file(s). This is not mere theory; the Peach virus (see page 17) exploits the weaknesses in the *Central Point Anti-Virus* checksumming process in exactly this manner.

"The checksum process as it stands is undeniably ill-conceived: it should either be made secure or scrapped."

The complaint made by the previous reviewer that checksums were always held on hard disk and were therefore vulnerable to attack/subversion by a virus, has been acted upon. *Central Point Anti-Virus* now contains an option to store checksums on a floppy disk and recalculation of new checksum files can be disabled. However, this only glosses over what is really a very poor attempt to implement checksumming. Other anti-virus programs cope with checksumming much better.

Immunisation

Central Point Anti-Virus provides facilities to immunise files against alteration by a virus. This process is not fully

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

explained; the documentation merely says that 'Once immunised a file has its own anti-virus capabilities', and an 'immunised file can 'heal' itself, returning to its original state'. Neither statement wins a prize for technical lucidity.

A list is provided of the types of file that won't work properly with immunisation: 'EXE files containing overlays or other information at the end of the file (usually debugging information), EXE files with a corrupted header, COM files larger than 63K, files containing an independent self-checking system, *Windows* files and OS/2 files'. This list is quoted exactly from the manual, and rather curiously omits the obvious example of files that modify themselves. Given the cursory nature of the 'checksum' process described above, it is easily possible to see how a file could be extensively modified, and still pass the integrity tests carried out by *Central Point Anti-Virus*.

I would caution against users applying immunisation to their own files. It doesn't always work, and even when successful, it alters the original manufacturer's executable file. The previous review of *Central Point Anti-Virus* explained in detail that there were too few cases where immunisation would work for it to be of much use. I concur with this advice. Original manufacturers of software may well introduce immunisation features within their own executable files, and provide well thought out and tested results. Users doing this for themselves run the risk of introducing subtle faults.

Memory-Resident Programs

The memory-resident programs work with either full monitoring for virus activity at all times (requiring 22K of memory), or monitoring only program execution and access to disk drives (requiring just 8K of memory). Each of these choices can be installed either as a terminate-and-stay-resident COM file, or as a device driver, and can be loaded high if required to save using up standard memory.

Obviously memory-resident programs which monitor computer activity for virus infection consume some of the computer's processing power. This is less relevant in these days of powerful hardware, but is more relevant when operating systems such as *Windows* are already quite profligate with computer resources. The actual overhead induced by these programs will depend upon the particular hardware and software configuration in use.

Conclusions

The original review was damning in its conclusion about some aspects of *Central Point Anti-Virus* being fundamentally flawed. Most of these problems have now been fixed, but one gaping hole remains. The checksum process as it stands is undeniably ill-conceived: it should either be made secure or scrapped. This conclusion is reinforced by the fact that at least one virus (there may be more) targets it for special treatment. Equally, the opportunities for immunisation are so few that this approach is not worth pursuing.

Having said that, money spent on *Central Point Anti-Virus* is not wasted. Its scanner is reasonable both in terms of speed and detection, and has knowledge of a creditably large number of viruses. Moreover, *Central Point Anti-Virus* is extremely easy to use and highly configurable to the varying requirements of the corporate end-user.

Technical Details

Product: *Central Point Anti-Virus*

Developer: *Central Point Software Inc.*, 15220 NW Greenbrier Pkwy, Suite 2000, Beaverton, OR 97006, USA, Tel: +1 (503) 690 8080, Fax: +1 (503) 690 7133, Bulletin Board: +1 (503) 690 6650.

European Vendor: *Central Point Software Europe Ltd.*, 3 Furzeground Way, Stockley Park, Uxbridge, Middlesex UB11 1DA, UK, Fax: +44 (81) 569 1017.

Availability: IBM PC/XT/AT or compatible, PS/2 (all models) PC- DOS/MS-DOS v3.00 or higher, at least v3.20 is recommended. 512K of RAM is required. A hard disk drive is recommended, but is not mandatory.

Version Evaluated: 1.2

Serial Number: 100182.001

Price: £115.00. Quarterly updates cost £19.50 each.

Hardware Used: A Toshiba 3100SX laptop with a 40 Mbyte hard disk, 5 Mbytes of RAM, a 16 MHz 80386 processor, and a single 3.5 inch floppy disk drive.

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

1049, 1260, 12 TRICKS, 1600, 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 800, 8 TUNES, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), Anti- Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Cascade (2), Casper, Dark Avenger, Datacrime, Datacrime II (2), December 24th, Destructor, Diamond (2), Dir, Diskjeb, Dot Killer, Durban, Eddie 2, Fellowship, Fish 6 (2), Flash, Flip (2), Fu Manchu (2), Hymn (2), Icelandic (3), Internal, Itavir, Jerusalem (2), Jocker, Jo-Jo, July 13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (2), LoveChild, Lozinsky, MIX1 (2), MLTI, Monxla, Murphy (2), Nina, Number of the Beast (5), Oropax, Parity, Perfume, Piter, Polish 217, Pretoria, Prudents, Rat, Shake, Slow, Subliminal, Sunday (2), Suomi, Suriv 1.01, Suriv 2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Taiwan (2), Terror, Tiny (12), Traceback (2), TUQ, Turbo 488, Typo, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Whale, Yankee (7), Zero Bug.

VIRUS BULLETIN **©1992 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /92/$0.00+2.50** This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

END-NOTES & NEWS

2nd International Virus Bulletin Conference, Edinburgh, September 2nd-3rd. Final programme now available. Tel 0235 531889.

Intelligent Quotient Ltd has been granted a full **US Patent (5086502) for the 'Real Time Continuous Backup'** technique which involves simultaneously copying changes to a disk drive onto another non-volatile storage device such as a tape drive. The copied information is stored as a continuous stream of data, rather than being overwritten, so that the original disk's contents can be restored at any time. The company is currently shipping *KRONOS Professional*, a real-time continuous backup utility for DOS PCs. Information from *IQ Ltd*, UK. Tel 0822 614477.

Sophos has released a generic **VAX/VMS version of the *SWEEP* virus scanner** which is available for *PATHWORKS*, the VAX-based PC networking system. *SWEEP for PATHWORKS* runs as a VAX/VMS process, checking DOS files held on the VAX server. It can be run as a permanent background process, constantly scanning DOS executables for viruses. *Sophos* claims the technology is 'completely immune' to subversion by stealth viruses. *SWEEP for PATHWORKS* is updated as new viruses appear and is sent out to subscribers each month on TK50 cartridges. Information from *Sophos Ltd*, UK. Tel 0235 559933.

IBM UK is holding a **PC Security workshop** (18th May), a **Virus Management seminar** (19th May) and a **Virus Hands-On tutorial** (20th May) at its *IBM Education Centre* in Sudbury. Tel 081 864 5373.

Protecting Your Computer Against Computer Viruses is a one day conference on 11th May 1992 at the Hyde Park Hotel, London. Information from *Westminster Management Consultants*. Tel 0483 740730.

S&S Consulting Group is holding seminars on **Data Recovery** (13th-14th May), **Viruses: The Management Response** (20th May) and **Advanced Data Recovery** (3rd-4th June) at Great Missenden, UK. Tel 0442 877877.

A two-day seminar on **Avoiding Corporate Fraud** presented by renowned investigator Mike Comer takes place on 3rd-4th June 1992 in London. Information from *IBC Technical Services* Ltd, UK. Tel 071 637 4383.

The ***NCSA 1st International Virus Prevention Conference & Exhibition*** (18th-19th June), Washington DC. Tel 717 258 1816.

Symantec has released ***Norton Backup* version 2.00** with tape support, automatic virus scanning and a scheduler for automated backup at predetermined times. Information from *Symantec (UK) Ltd*. Tel 0628 776343.

Fifth Generation Systems has released ***Fastback Plus* 3.02** with *Windows* interface, mouse support, scheduler, DES file encryption and *Novell* network support. Information from *Fifth Generation Systems*, USA. Tel 504 291 7221.

VIRUS BULLETIN

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

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

This publication has been registered with the Copyright Clearance Centre Ltd. Consent is given for copying of articles for personal or internal use, or for personal use of specific clients. The consent is given on the condition that the copier pays through the Centre the per-copy fee stated in the code on each page.