September 1990 ISSN 0956-9979

VIRUS BULLETIN

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

Editor: **Edward Wilding**

Technical Editor: **Fridrik Skulason**, University of Iceland

Editorial Advisors: **Jim Bates**, Bates Associates, UK, **Phil Crewe**, Fingerprint, UK, **Dr. Jon David**, USA, **David Ferbrache**, Information Systems Integrity & Security Ltd., UK, **Dr. Bertil Fortrie**, Data Encryption Technologies, Holland, **Ray Glath**, RG Software Inc., USA, **Hans Gliss,** Datenschutz Berater, West Germany, **Ross M. Greenberg**, Software Concepts Design, USA, **Dr. Harold Joseph Highland,** Compulit Microcomputer Security Evaluation Laboratory, USA, **Dr. Jan Hruska**, Sophos, UK, **Dr. Keith Jackson**, Walsham Contracts, UK, **Owen Keane**, Barrister, UK, **Yisrael Radai**, Hebrew University, Israel, **John Laws,** RSRE, UK, **David T. Lindsay**, Digital Equipment Corporation, UK, **Martin Samociuk**, Network Security Management, UK, **John Sherwood**, Sherwood Associates, UK, **Roger Usher**, Coopers&Lybrand, UK, **Dr. Ken Wong**, BIS Applied Systems, UK.

CONTENTS

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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

Crying ''Wolf!''

A noticeable tendency in recent months has been for software companies and other virus 'interested' agencies to cry ''wolf!''. These commercial agencies thrive on computer viruses which have now become big industry particularly in Europe and the United States. Each and every new 'specimen' is greeted with 'shock-horror' exclamations and over-dramatic pronouncements of impending Armageddon. These organisations then offer solace through the purchase of their particular brand of medicine - usually an "all singing, all dancing" software remedy.

This commercial exploitation of the virus problem, driven by the need to produce ever more spectacular and frightening 'coups', is giving rise to serious questions about ethics and responsibility.

The recent *PC Today* incident in the United Kingdom (*VB, August, 90*) revealed a counter-productive tendency on behalf of some organisations and individuals to promote panic, unwittingly or otherwise.

There can be little doubt that the production of 'freebie' disks at *Database Publications Ltd* was seriously mismanaged prior to this incident. A letter from the Features Editor of *PC Today* which appears on page eight states that security has now been tightened. There are many disturbing issues arising from the incident, not the least of which concerns the reaction of *Database Publications*' management upon suspecting that they were responsible for distributing virus code nationwide. The first action taken, ill-advised as it now appears, was to contact the news media and other interested parties - a guaranteed way to promote hysteria.

The fact that the virus code and its trigger action on all the duplicated disks was inactive and that this information could be confirmed within half an hour of professional analysis should have rendered urgent alerts quite unnecessary. Instead, the *VB* office was innundated with calls from subscribers and others who had read apocryphal and lurid accounts of this incident on bulletin boards in both the UK and the States.

Widescale public alerts about these incidents are counter-productive. *Proclamations of impending disaster will be ignored if they consistently prove to be inaccurate.* This is the 'cry wolf' syndrome and may lead organisations to dismiss warnings even when they are warranted. *Public pronouncements that virus code is readily available at newsagents are also irresponsible and will inevitably result in inquisitive tinkering or more malicious activities*. In fact, this virus code, despite its inactivity, is probably in wider circulation and the subject of more extensive experimentation than would have been the case had no announcement been made. *Exaggerated clamour on this sensitive subject may well lead to public and corporate cynicism and a relapse from current vigilance.*

The 'virus industry', which is now a regrettable fact of life, must not stoop to the level of creating or exaggerating the dangers. The desire to produce the best software to combat the greatest number of viruses and the competitive urge to be the first with the 'news' is serving to reduce professional standards. *Statements are being made without due care and analysis*. This tendency towards grand announcements, noticeable on both sides of the Atlantic, is not in the interests of computer users and is to be frowned upon.

Regarding the *PC Today* incident, magazines which distribute 'freebie' software must take note of this incident and learn from it. At the very minimum, publishers intending to distribute such software must screen development machines, as well as disk duplication process. Many software companies are now using notchless disks (i.e. write-protected). However, *Database Publications*, in line with nearly all other UK PC magazine publishers, distributes its software on notched disks.

Any publisher or other distributor should recall virus-infected software. In this instance, hysteria was induced for no good reason because *Database Publications* announced, by means of a press release, that the disks contained virus code. *This information was widely and publicly disseminated before anyone had full possession of the facts*. A message on the CIX bulletin board on the evening of July 24th was among the first alerts placing the information in the public domain.

Learning from the various blunders that encapsulate the whole incident, future organisations finding themselves in this unfortunate situation would be well advised to get expert confirmation that suspect disks **are** virus infected and **active** before any further action is taken. *This analysis can be done quickly and efficiently without any need for public announcements.*

If the disks are in any way dangerous they should then be recalled. In the United Kingdom, two retailers, *John Menzies* and *W. H. Smith* are responsible for nearly eighty per cent of high-street newspaper and periodical trade; *notification to both outlets would be eminently sensible*. Subscribers who receive the contaminated software should be notified by first-class post. General alerts should only be posted once all dangerous materials have been taken out of circulation. Of course, such actions will only be necessary if slack or non-existent security continues in the publishing community.

Crying ''wolf!'' is both dangerous and pointless. We wish to assure subscribers that should they receive an alert from the *Virus Bulletin* at some point in the future, it will be because that alert is *warranted*.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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 NOTES

"Companion" Viruses

It may seem simple to give a precise definition of a computer virus. One of the most common definitions is:

A program that can infect other programs, by modifying them to include a (possibly altered) copy of itself.

However, this definition is not perfect as it does not cover one class of viruses, currently represented by the TPworm and AIDS II viruses reported in this month's update to the *VB Table of Known IBM Viruses* (pages 6-7).

These "companion" viruses do not alter programs They spread by using a special feature of the operating system - the fact that if two programs exist with the same name, but different extensions (COM and EXE), the MS-DOS operating system will execute the COM file. The viruses exploit this by locating an EXE file and creating a new program in the same directory, with the same name, but a COM extension. This program containing the virus code, is then hidden by changing the file attributes. When the user attempts to execute the EXE file, the COM file is activated instead. The virus does whatever it was designed to do and then it simply executes the EXE file. **Integrity checking programs which do not report the presence of unauthorised files or which only monitor EXE files will not detect 'companion' viruses.**

Joker - a Virus?

Earlier this year a diskette was sent from Poland to virus researchers in the West. It turned out to contain samples of the Oropax virus, two Polish variants of the Vienna virus, named W13-A and W13-B, as well as the Vcomm virus.

In addition the diskette contained a program named JOCKER.EXE (*sic*). Whether this program is a virus or not, is disputed. It has been widely reported as a computer virus but there is very little evidence to support these reports.

Virus researchers have been unable to make the program do anything of interest - when executed it will just display the message "Error in EXE file", as if DOS was reporting a damaged file. The program contains several interesting text strings, some of which contain spelling or syntax errors:

```
Water detect in Co-processor
I am hungry! Insert HAMBURGER into Drive A:
Hard Disk's head has been destroyed.
Can you borow me your one?
Missing light magenta ribbon in printer!
Insert tractor toilet paper into printer.
Disconnect your mouse, there are some cats!
```

These text strings would be typical of the TSR "joke" program - a Trojan which might display one of them whenever the user entered a command, but so far this behaviour has not been observed, nor is there evidence that JOCKER.EXE contains a virus, other than a report from one person who claims to have found traces of it in another program, after running JOCKER.EXE. This ability to replicate has not been observed under test conditions.

As the program is fairly long (12,806 bytes), does not appear to be a serious threat, and has appeared at a time when more pressing analysis of other programs has proved necessary, nobody has had time to disassemble it. The question of whether JOCKER.EXE is a virus or not still remains unanswered. *Someone, somewhere is laughing.*

Shareware Anti-Virus Programs

As *VB* reported last month, many companies have restricted the use of shareware and freeware programs, to reduce the risk of virus infections. The question arises if these restrictions should apply to shareware anti-virus programs as well.

The shareware programs are less expensive, but they have a major problem, which is the possibility of becoming infected with a virus, or corrupted in some other way, somewhere in transit between the author and the user. Other problems are the lack of support, documentation and sufficient beta testing.

On the other hand, the best shareware programs are updated faster than commercial programs - the delay between the first reporting of a new virus and an updated version of the program is often just a few days. From a technical point of view, shareware programs are not inferior - some of the Macintosh programs in particular are better than most, if not all, of the commercial ones. **Provided that the software is obtained from a reliable source, there is no special reason to avoid reputable shareware programs**. (*See 'Should We Trust Public Domain Anti-Virus Software?', VB January '90*).

''Multi-Partite'' Viruses

It used to be possible to divide viruses into two clearly defined groups - boot sector viruses (infecting the Partition Boot Sector or the Disk Boot Sector), and parasitic viruses, infecting COM and/or EXE files.

A new group of viruses has appeared recently which is able to spread by infecting both the boot sector and program files. This type of virus writes a short loader-type program to the boot sector, whose purpose it is to load the virus into memory. It will then also infect programs as they are executed. This ability to infect multiple system and program elements gives rise to the term ''multi-partite' infection. The viruses currently known to use this method are Anthrax, Flip (*see Virus Dissection, pp.18-20*) and V-1, but their numbers will, without doubt, increase in the future.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

OVERVIEW

"Stealth" Viruses

Good camouflage obviously improves the chances to penetrate enemy territory, without being noticed. The development of 'radar invisible' aircraft by the *United States Air Force* is testament to the role of deception in warfare. The expression "stealth" has now been purloined from the military to describe a series of computer viruses which attempt to hide from the enemy - *the combined forces of users and anti-virus software.*

Specifically, the term "stealth" is used to refer to the group of viruses which make the virus code *disappear* from the infected media while they are active in memory. *

The first "stealth" virus was the old and well-known Brain virus. While it was active in memory the virus would intercept any INT 13H operation, and if the boot sector was read, the virus would return the original non-infected boot sector instead. The virus could be detected in memory, but all infected diskettes looked "normal", while it was active. As this simple method reduces the probability of detection, it is expected to be a common feature of future boot sector viruses.

In the case of parasitic viruses, the implementation of camouflage methods is more complex. Two conditions must be met:

1. *Any increase in file size must not be detectable when the user issues a DIR command.*

2. *Any program reading from the file must not read the virus code, only the contents of the original program.*

The "companion" viruses described on page 3 of this edition fulfill both conditions, as they do not alter the "infected" file at all. It has not yet been decided whether to include this set of viruses in the "stealth" category. *The fact that "companion" viruses cause no file alterations and do not infect any programs at all appears to place them in a unique category*.

It is easy to avoid an increase in file size by overwriting a program with the virus code, as the 405, 382 and Burger viruses do. This method destroys the original file, making the virus easily detectable. It is not a "stealth" technique.

The method used by the "Number of the Beast" virus is of considerable interest, as the virus code is hidden in unused, free space after the end of the program itself. This method has one serious drawback (from the virus writer's point of view) - the virus code is **not** included when the DOS COPY command is used to copy an infected program.

The most advanced method for hiding any increase in file length consists of intercepting the "Find first" and "Find next" functions of INT 21H. If the information returned indicates that the file is infected, the virus modifies it, returning the original length of the program. This method is used by Zero Bug and 4K (Frodo). If such a virus is active in memory, it is possible that programs will be irreparably damaged. The reason is a mismatch between the number of clusters in use according to the FAT and the number of clusters required according to the reported (but incorrect) length of the file. Running the DOS CHKDSK program while the virus is active in memory may result in a number of reported errors in the FAT. If the user attempts to correct this, by running CHKDSK/F, the virus-occupied clusters will be freed, making it impossible to recover the original program.

The second condition a virus must fulfill to belong in the ''stealth'' category is more difficult to implement. It is possible to hide virus code by the use of several methods, two of which are currently used. More sophisticated methods, not used in currently known viruses, are not described here.

The first method involves intercepting "open file" function calls and determining if the file being opened is an infected program. If so, the file is disinfected before control is passed on to DOS. Any anti-virus program opening a file for examination will not detect virus activity in the file.

A virus using this method, 4K for example, can be removed by a very simple disinfection method. Entering the command

```
COPY *.* NUL
```

in each directory will remove the virus from all infected programs. This will work unless, of course, the virus re-infects the file as soon as it is closed, resulting in files being "clean" whenever a program examined them, but infected otherwise.

The second method involves intercepting the "Read" function and exchanging the contents of the input buffer with the original code, whenever a part of the virus code is read. This method is used by the "Number of the Beast" virus, which overwrites the first 512 bytes of infected files with the virus code. When this part of the program is read, the virus locates and returns the original contents of the first 512 bytes.

Checksum Problems

Checksum programs are vulnerable to "stealth-type'' viruses, because their effectiveness is based on the assumption that the program they read from the disk is identical to the program which will be executed. **The importance of running this type of program only after the computer has been booted from a "clean" write-protected system diskette must therefore be emphasised once again.**

** In last month's VB, the 1260 virus was called 'Stealth'. The name has now officially reverted to 1260. Viruses which rely on a random encryption key to avoid detection are not categorised as "stealth" viruses.*

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

PROCEDURES

New Zealand Virus - A Non-Destructive Disinfection Routine

An increasing number of New Zealand (2) * virus infections are being reported from around the world. Eliminating this particular computer virus is a more involved task than removing an infection by a parasitic virus or any other boot sector virus currently in the wild. The procedures outlined here are designed to simplify disinfection of disks infected by this very common virus.

The reason for this added complexity in disinfecting an afflicted PC is that New Zealand is one of a few known viruses to date which infect the Disk Boot Sector (*see pp.15-16*). This is the first physical sector on a partitionable hard disk, addressable as sector 1, track 0, head 0. A dissection of the virus can be found in the May edition of *Virus Bulletin*.

The following is a short-cut procedure for eliminating the virus from infected disks using The Norton Utilities (tm), version 4.0 or above. Note that the virus **cannot** be eliminated from the hard disks by using the DOS FORMAT command - an alternative to this procedure is to use a low-level disk formatting program. (*See Detection and Brute Force Disinfection, VB, July 1990, pp 3-5*). Floppy disks can be disinfected by using the DOS FORMAT command.

Disks needed:

1. **Clean, write-protected system floppy disk** normally supplied by the PC manufacturer.
2. **Write-protected floppy disk with The Norton Utilities, V4.0 or above.** (*Referred to below as NU*).

Procedure

1. Switch the PC off.
2. Insert the clean write-protected system floppy disk.
3. Switch the PC on.
4. After the PC has bootstrapped, insert the write-protected floppy disk with NU.
5. Type
   ```
   NU C:
   ```
6. Select “Explore disk” option.
7. Select “Choose item” option.
8. Select “Absolute sector” option.
9. Select drive “C:”.
10. “Drive selected is C:” will be displayed. Select: “Side: 0”, “Cylinder 0”, “Sector 7”, “Number sectors: 1”
11. Select “Edit/display” option. Check that the sector displayed is the Disk Boot Sector - you will see some text in the right-hand column and the last two bytes of the sector will be “55 AA” hexadecimal. **Do not proceed if this is not the case**.
12. Select “Write item to disk” option.
13. Select “Absolute sector” mode.
14. Select new drive “C:”.
15. “Drive selected is C:” will be displayed. Select: “Side: 0”, “Cylinder 0”, “Sector 1”.
16. A warning will be displayed. Select “Yes” option.
17. NU will display “Finished writing”. The PC hard disk is now disinfected.
18. Press Esc repeatedly to exit from NU.

Caveat:

The above method works for the New Zealand (2) virus, but may not work in the case of a virus mutation. **Make sure that the sector displayed in step 11 above is the disk boot sector before copying it to its original position.**

Eliminating New Zealand from Floppy Disks

1. Bootstrap the PC from a **clean, write-protected system floppy disk** normally supplied by the PC manufacturer.
2. Backup any data from the infected disk (this can be done quite safely).
3. Use DOS FORMAT to eliminate the virus.

Further Notes

Data corruption: Please note that the New Zealand virus can corrupt the first File Allocation Table (FAT) on some hard disks. You can use NU to copy the second (uncorrupted) FAT into the first FAT. On 1.2 MByte 5 1/4 inch floppy disks corruption is also likely to occur, as the virus overwrites the third sector of the root directory, corrupting disks with more than 32 files.

Note: The above procedure will **not** disinfect hard disks infected by New Zealand (1).

* New Zealand (2) refers to the nomenclature used in the *VB Table of Known IBM PC Viruses* published in August 1990. This version of the virus is very common. We have received **no** reports of New Zealand (1) being found in the wild.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

KNOWN IBM PC VIRUSES (UPDATE)

This is a list of new computer viruses viruses affecting IBM PCs and compatibles, including XTs, ATs and PS/2s. The first part of the list gives aliases and brief descriptions of viruses which have been seen, while the second part lists viruses which have been reported. Each entry consists of the virus group name, its aliases and the virus type (See "Type codes" table). This is followed by a short description (if available) and a 10 to 16 byte hexadecimal pattern which can be used to detect the presence of the virus by the "search" routine of disk utility programs such as The Norton Utilities or your favourite disk scanning program. Offset (in hexadecimal) normally means the number of bytes from the virus entry point. For parasitic viruses, the infective length (the amount by which the length of an infected file has increased) is also given.

Amendments and updated information as of August 24th, 1990. The full table was published in *VB*, August 1990.

Type Codes:

C = Infects COM files	**D** = Infects partition boot sector (Logical sector 0 on disk)
E = Infects EXE files	**M** = Infects disk boot sector (Track 0, head 0, sector 1 on disk)
N = Not memory-resident after infection	**R** =Memory-resident after infection
P = Companion Virus [1]	

SEEN VIRUSES

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

```
1024    00B4 40CD 2172 043B C174 01F9 C39C 0EE8   ; Offset 170
```

AIDS II - PN: A "companion" virus [1], 8064 bytes long, which displays a message when it activates.To locate the virus, search for COM files corresponding to EXE files, but marked "Hidden" and located in the same subdirectory.

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

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

Burger - CN: Just like the 405 virus, this primitive 560 byte virus overwrites the infected files, which makes it easily detectable.

```
Burger  B447 0401 508A D08D 3646 02CD 2158 B40E ; Offset 01B
```

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

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

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

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

Joshi - MR: This virus from India displays the message "type Happy Birthday Joshi" on 5th January of every year. Unless the user enters the text verbatim, the computer will hang.

```
Joshi   50CB BB78 0036 C537 1E56 1653 BF2A 00B9 ; Offset 046
```

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

```
Slow    E800 005E 8BDE 9090 81C6                  ; Offset 0
```

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

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

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

TPworm - PN: A ‘companion’ virus [1] written by the author of the Vacsina and Yankee Doodle viruses. The virus has been distributed in the form of ‘C’ source code. Hence, the infective length and hexadecimal patterns depend on the ‘C’ compiler used.

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

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

V-1 - DCR: This virus is one of the first to infect both the boot sector and programs [2]. It is 1253 bytes long and destructive: when activated, it overwrites the disk with garbage.

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

REPORTED VIRUSES

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

1226 - CR: Reported in Bulgaria.

1381 - EN: Virus reported to contain the string “INTERNAL ERROR 02CH”.

2100 - CER: A Bulgarian 2100 byte virus, probably written by the author of the Dark Avenger.

AirCop - DR: Virus may display the message “Red State, Germ Offensive. AIRCOP” or crash the system. Originated in Taiwan.

Anthrax - MCER: One of the first viruses to infect the Disk Bootstrap Sector as well as programs [2]. Side-effects are not known, but the length is reported to be between 1040 and 1232 bytes. Originated in Eastern Europe.

AntiCad: Possibly an alias of V-1, but may also be a different 10005 byte long virus.

AntiPascal, 605 - CN: This Bulgarian virus corrupts PAS and BAK files. A 400 byte mutation has also been reported.

Casper - CN: This virus uses the same encryption method as the 1260 virus, but its length is only 1200 bytes.

Filler - DR: Reported in Hungary.

G-virus - CR: Reportedly a mutation of the Perfume virus, but with different text strings: ‘‘G-VIRUS V1.3” and “Bitte gebe den G-Virus Code ein”.

Leprosy - CRN: An overwriting virus with an infective length of 666 bytes. It originates from California.

Mardi Bros - DR: A French virus which changes the volume label to “Mardi Bros”.

Microbes - DR: A virus from India with unknown side-effects.

Ontario - CER: A 512 byte encrypted virus from Canada.

Phoenix, P1 - CR: A family of three 1701 or 1704 byte viruses from Bulgaria, not related to Cascade viruses. They are reported to use a self-modifying encryption method, similar to the that used in the 1260 virus.

Plastique - CER: A bug-ridden 3012 byte virus from Taiwan. A 4096 version has also been reported.

Taiwan 3 - CER: Probably not related to the Taiwan virus, this is a 2900 byte virus from Taiwan with unknown side-effects.

TCC - CER: A 4909 byte virus from France. Side-effects are unknown.

Tiny family - CN: A family of the smallest viruses discovered so far.Their sizes range from 158 to 198 bytes and their country of origin is Bulgaria.

Wolfman - CER: A 2064 byte virus from Taiwan.

[1] The term ‘companion virus’ describes any computer virus which locates an EXE file and then creates a new program in the same directory with a COM extension. This bogus COM file contains the virus code and is invoked before the legitimate EXE file executes. For more information see *Technical Notes* on page 3.

[2] The appearance of computer viruses which infect both the boot sector and program files is a relatively new phenomenon. Two examples, Flip and V-1 have been analysed with a further specimen ‘Anthrax’ appearing in this month’s ‘reported only’ section.The type code of the Flip virus (published inaccurately in last month’s *VB*) has now been corrected to read MCER. (See *Technical Notes* on page 3 and *Virus Dissection, pp. 18-20*).

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

LETTERS

PC Today
Database Publications
Europa House
Macclesfield SK10 4NP

7th August, 1990

Dear Edward,

Further to our conversation today, I am most concerned about the report in your latest *Virus Bulletin* that a variant of the Stoned virus was found on our free disk entitled 10 Of The Best. This was on the cover of the March 1990 issue of *Personal Computing* (now *PC Today*) volume 3 number 11.

In view of the current publicity about the Disk Killer virus and our cover disk, I would like to make your readers aware of the circumstances since it would be possible for them to jump to the wrong conclusion - that the March disk was duplicated with a virus. This was not the case.

The incident concerned one specific disk. Tests on both the master copy and samples taken from newsagents' shelves failed to reveal an infection. The person who made the complaint suggested the most likely explanation to be that the newsagent's teenage son had taken the magazine from the shelf, run the disk on an infected machine, then replaced the magazine/disk in the shop.

To help avoid the repetition of such an incident, from the June issue we instituted a system whereby the cover disk has a self-checking CRC routine to verify that files have not been altered in any way. Not totally foolproof, but no other magazine is doing this.

With regard to the wider question of preventing viruses finding their way on to masters and thereby on to duplicated disks, from the September issue our master copies are being checked by Alan Solomon's company *S & S Enterprises*. Allied to more stringent checking at our office, this particular spectre should now be laid to rest as far as *PC Today* is concerned.

When you have verified this story, I'd be grateful if you could send me a copy of whatever you print on the subject.

Yours sincerely,

Ian C. Sharpe
Features Editor

Editor's comment. Mr. Sharpe points out that the 'Ten of the Best' disk distributed by *Database Publications* in March of this year had become virus infected **after** duplication and distribution. The infected disk was thus an isolated incident.

Our report (*VB, August 1990, p 24.*) was published to emphasise the fact that Mr. Sharpe was made singularly aware of the dangers presented by computer viruses six months prior to the *PC Today* incident. I also find it inexplicable that *PC Today's* July edition included an anti-virus program which either went unused or was totally ineffective.

If Mr. Sharpe's version of events in March is correct, and there is no reason to think it is not, the disk was infected by the newsagent's son. Software in high-street newsagents is vulnerable to such accidents as well as malicious tampering. A number of contamination incidents involving ground glass in baby foods, mercury injected into oranges and other such acts of terrorism aimed at consumers have forced manufacturers to adopt tamper-resistent packaging. Magazine publishers might care to take a leaf from the food industry and introduce tamper-resistant notchless floppy disks.

IBAS
Box 1250, N-2201
Kongsvinger, Norway

Sir,

Comment to Jim Bates article *'Datacrime II - Refined Hatred'* in *VB,* August 90.

We noticed from Mr Bates article that he believes that there is 'virtually no hope of recovery' [of the hard disk data] after having run the trigger routine of the Datacrime virus. The damage that this virus performs is according to Mr Bates that the first cylinder of the first physical hard disk in the PC is low-level formatted. This results in the Master Boot Record (*aka Disk Boot Sector, Ed.*) being destroyed, along with more or less of the FATs depending on the disk's size.

We would like Mr Bates (and *VB* readers) to know that the above descibed damage indeed has the best chances of recovery. In fact, all partitions but the first can be recovered totally without any complications. Also, non-fragmented files residing in subdirectories of the first partition can easily be recovered. Even more data can be recovered but at greater expense.

IBAS is a company which specialises in data recovery. We do recover damages like the one described above almost daily and we regard these as simple recovery cases. In our laboratories we also recover data after physical damage, like a headcrash. Where the magnetic coating has been damaged or removed, we recover data from the surrounding areas where the coating is still intact. It is also possible to recover data which physically has been overwritten.

Guro Bye BSc.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

Copy Protection Systems - a Necessary Evil.

Having read Dr Keith Jackson's damning conclusions (*VB, August 1990*), concerning copy-protected anti-virus products and your editorial decision to exclude such from all future reviews in *Virus Bulletin*, I wondered if your readership may like to read the other side of the argument.

I should point out that *Visionsoft's* anti-virus product, Immunizer, is indeed copy-protected but this was **not** the subject of any of Keith's criticisms as it was not submitted for review before your recent ban on protected software.

To provide a defence, we need to start by identifying the factors that make Keith so opposed to protection mechanisms. *Is it copy-protection that Keith objects to, or the nuisance element that seems to be a by-product of such mechanisms?* If we can reduce or remove these annoyances, perhaps copy-protection does have a place.

Keith quotes a catalogue of horrors associated with faulty copy-protected software. *These would still apply if the products in question had not been copy-protected.* For instance, software that can not be uninstalled easily, and delays with a supplier changing disks from one format to another. These are problems with software design and company efficiency, **not** copy-protection.

Keith's condemnation of the necessity for the master disk to always be present when running the software would normally have my full support. However, there are occasions when this is actually good practice - *virus protection is one such occasion*. The last thing anyone wants is for the anti-virus software to become a carrier. I believe that any EXE components that can reasonably be left off the hard disk during installation should be kept on a write-protected distribution disk.

Most, if not all, of Keith's objections can be removed by a well designed protection method and an efficient customer support plan. Using Immunizer as an example:

The program is provided on dual media as standard, so there are no problems about not being able to copy the software to your particular disk format. In the unlikely event of the user requiring a replacement disk, the company would send a replacement to a registered user the same day that the request was made.

The main component is a memory-resident device driver, that is not subject to any copy-protection mechanisms. The user interface software which is only needed for installation or configuration is a copy-protected EXE file. Thus you would only require the master disk to be present when performing any customisation via this interface. At all other times this EXE is stored on a copy-protected disk. At no time is any protected software installed on the user's hard disk.The user can remove the program from the machine at any time with a couple of key presses, via the above interface program.

While I sympathise with Keith's experiences, I feel that by grouping all copy-protection mechanisms under one umbrella, you are doing a slight disservice to products where copy-protection has been kept to a calculated minimum.

Why do companies choose copy protection in the first place? The expense of development costs of some anti-virus products, equivalent to the sums invested in product leaders in the spreadsheet or database arena, make such products a major investment. It is sound business practice to attempt to protect them in some way.

Obviously, as far as *Virus Bulletin* is concerned, the die is cast and copy-protected products are now taboo. But perhaps some of your readership may agree that copy-protection does not have to be a nuisance and is a necessary evil.

Kevin Powis
Technical Manager
Visionsoft Ltd.

Keith Jackson comments: Mr. Powis suggests that copy-protection can sometimes be a useful thing. I beg to differ on this point. *Copy-protection is a nonsense, always was and always will be.*

As for musing on whether I object to copy-protection per se, or to problems created by copy-protection schemes, I thought that I had made myself quite clear:

Copy-protection prevents users taking regular backups, interferes with the operation of software packages, and adds one more level of complexity to what may already be quite a complex process.

I usually sum this up with the quote "Life is too short to deal with the 'ritual dance' imposed by copy-protection schemes". Visionsoft have been aware of my views for some time (since 22nd May 1990 to be precise). They have chosen to copy-protect their anti-virus software and I think such a decision is wrong. Companies that purchase large amounts of software seem to agree with my expressed views, and studiously avoid copy-protected software. Visionsoft's sales will be the final arbiter of this matter.

A few points of fact:

1) The letter states that my 'catalogue of horrors' associated with copy-protection would still apply if the products were not copy-protected. This is provably untrue.

2) I disagree that my objections can be removed by a well designed copy-protection method, and an efficient customer support plan. They cannot.

3) The best 'calculated minimum level' of copy-protection is not to impose such methods in the first place.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

WORLDWIDE

Chinese Census Crisis

Concerned about possible disruptions to the fourth national census, the Chinese *state statistics bureau* has devised regulations aimed at curbing damage to national statistical computing systems from computer viruses. Reporting from Beijing (August 27th), the *Xinhua* press service says that the regulation was enacted “to combat a dozen kinds of viruses which are threatening various computer systems in China”. The *state statistics bureau* is responsible for national social, economic, scientific and technological statistical work and related data processing. The agency owns some 10,000 micro and minicomputers.

United States Proposes Tough Legislation

A congressional hearing has welcomed the proposals of a Senate bill to modernise the *Computer Fraud & Abuse Act 1986*. The bill extends the definition of “access” to include the intentional transmission or distribution of unauthorised and pernicious software. Felony penalties could run to a maximum five year prison sentence and a $250,000 fine.

However, the bill goes further, saying that anyone who unknowingly but “recklessly” transmits destructive software could face a misdemeanor penalty of up to one year’s imprisonment and a fine of $5,000. The bill also recommends extending these provisions beyond ‘federal interest’ computers to include computers used in interstate commerce and communications. It would also allow civil actions for compensation against losses caused by computer misuse.

The *US Department of Justice* stated that the misdemeanour provision was designed to add flexibility. *Deputy Assistant Attorney General* Mark Richard said “It could be employed against computer hackers and persons responsible for computer viruses where the intent to damage a system or defraud can’t be shown conclusively”. The misdemeanour provision is causing disquiet in legal circles due to the likelihood of lawsuits against companies responsible for accidental computer virus transmission.

Marc Rotenberg of *CPSR* (*see below*) said ‘‘The reason for this recklessness provision is to make clear to computer users that they should not engage in experiments that place other users at risk’’.

US Group Addresses Civil Rights

The *Computer Professionals for Social Responsibility* (CPSR), a public advocacy group based in Palo Alto, California, is to host a series of discussions in Washington DC about computer users’ civil rights. Eric Roberts, *CPSR* president said a gap between technicians and policy makers should be bridged. According to Roberts there is too much misinformation about computers and networks which “could produce terribly misguided policy”.

Electronic information systems are giving rise to a number of legal debates. Fundamental issues to be tackled include the concept of ‘electronic trespass’ and whether or not digital information is protected by the *First Amendment* of the *United States Constitution*.

Roberts hopes to bring together key policy makers, computer users, law enforcement agencies, computer industry representatives and security experts. *CPSR* will also act as a watchdog over computer crime investigations. The project is underwritten with a $275,000 grant from the *Electronic Frontier Foundation* founded by the computer pioneer Mitch Kapor.

Virus Plagued Russia Invokes the ‘Hooliganism’ Act

One of the earliest cases of a computer virus in the Soviet Union occurred in 1988 when an unidentified programmer at the *Gorky Automobile Works* on the Volga river was charged with deliberately using a virus to shut down an assembly line in a dispute over work conditions. The man was convicted under Article 206, the ‘hooliganism’ law. Article 206 states that “*violating public order in a coarse manner and expressing a clear disrespect toward society*” is an offence punishable by a maximum prison sentence of six years’ imprisonment.

Pirated software has been reported as very common in the Soviet Union and is believed to have contributed to the escalating virus problem. Computer viruses at large in the USSR include Dark Avenger, Yankee, Vacsina, Sunday, Pixel, Disk Killer, Cascade, New Zealand, Italian, Jerusalem, Brain, W-13 and a number of unspecified members of the Bulgarian family. Indigenous anti-virus programs include Aidstest by *Lozynsky*, Anti-Kot and Anti-Kor by *Kotik*.

Japanese Computer Viruses “Commissioned”

The *Japan Times* reported in July that 40 computer hackers across Japan were paid to write computer viruses specifically targeted at *Sharp Corporation* products.

The *Japanese Computer Clubs Federation* claimed that a high school student working at the club’s facilities had participated in developing and distributing virus programs.

A version of ‘Farside Moon’, a computer game from *Art Dink Co.* (Narashino, Chiba prefecture) was reported as being virus infected. 1,500 copies of the infected program, which is compatible with the Sharp X-68000 PC, are believed to have been distributed.

Sharp Corporation are understood to have shipped 100,000 X-68000s. Copies of ‘Farside Moon’ are being recalled.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

SPECIAL FEATURE

David Ferbrache

Trojan Horse Techniques to Compromise Sensitive or Classified Data

In this article, David Ferbrache examines the threat to high security systems posed by Trojan horse and viral code. In particular the issue of compromise of classified data through the use of such code is considered.

Secure systems tend to apply two main forms of access and data flow controls, namely:

- **Discretionary access controls** (DAC)

- **Mandatory access controls** (MAC)

In terms of the *U.S. Department of Defense Trusted Computer System Evaluation Criteria* (TCSEC, also called the 'Orange Book') [1] these fall into class C and B systems respectively. Discretionary access systems rely on the user (or the system by default) to specify restrictions on access to user data. Such restrictions are in the form of an *access control list* (ACL) or matrix, or in the form of a capability based system.

The former consists of a list of authorised users of each classified object, the latter allows the creator of the classified object to pass out a right to access the object. This right (or capability) can be propagated by the receiver.

A utility invoked by a privileged user could access data at that user's access level, and without restriction can normally write or propagate data to the unrestricted access files or networks. **Trojan horses can thus compromise data in such environments**. A Trojan horse in the form of a useful utility could easily be run by a wide variety of users, with a wide variety of system permissions.

Mandatory Access Controls

The military security principles of **formal clearance**, and **need to know** before access is allowed to classified data, have been encapsulated in the *Bell-LaPadula* security model. This model (which is the basis of the TCSEC 'Orange Book' mandatory access model) specifies two constraints upon data flow, namely:

- **No Read Up** - a user's clearance must be greater than or equal to the classification of the object or data he is trying to read.

- **No Write Down** - a user may only write data to objects of classification greater than or equal to his clearance.

These two principles implement the "formal clearance" requirement, preventing the user from reading highly classified data, and from declassifying data by writing it to a lower security file. Data is thus constrained to remain at the classification, or to be raised in classification level.

This simple model of hierarchical security classification (e.g. UNCLASSIFIED, RESTRICTED, CONFIDENTIAL, SECRET, TOP SECRET) is extended by including the concept of compartments of information. The compartment system prevents data from flowing out of a compartment. Each user is then authorised to access certain compartments of data (at a given classification or lower), thus implementing the "need to know" by further access restrictions.

The MAC scheme inhibits the writing of data to a lower classification file (or a file which is not in the compartment), where it could be potentially compromised by the person introducing the Trojan. In the absence of DAC there is **no** restriction on the Trojan horse copying its code into higher and higher classification files (with stricter and stricter compartments). MAC only attempts to restrict the flow of information to lower classification environments, not to higher classifications.

The 'Orange Book'

Devised by the *United States Department of Defense* in 1975, the 'Orange Book' (officially, *DOD Trusted Computer Security Evaluation Criteria*) provides formal criteria for security in a multi-user computer system. The Orange Book is concerned with the **confidentiality** of data which limits its relevance to commercial computer security with its increasing emphasis upon systems and data *availability* and *integrity*.

The Orange Book criteria are divided into **divisions** and **classes**. Confidence or trust in a computer system increases with each successively higher division awarded. The criteria are divided into four divisions: D, C, B and A ordered in a hierarchical manner with Division D reserved for systems failing to meet evaluation standards and Division A being reserved for systems providing the most comprehensive security. There are then a number of sub-divisions called classes (C1, C2, B1, B2, B3, A1 and A2). Covert channel analysis, whereby covert channels must be searched for and the maximum bandwidth of each channel measured, commences at Division B Class B2.

Trusted Computer System Evaluation Criteria, Department of Defense, Computer Security Center, Fort George Meade, Maryland 20755, USA.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

In summary:

1. DAC - cannot prevent data from being written to lower classification files.

2. MAC - prevents 1. but cannot prevent the Trojan from writing its code to higher classification files.

Integrity Levels and Compartments

An extension to the *Bell-LaPadula* model to incorporate the concept of program and data integrity has been proposed, using the corresponding idea of integrity levels and compartments [2]. Thus each program in the environment has an associated "integrity" level which represents the certainty that the program is a correct implementation of an authorised algorithm. This element of certainty may be based on the adoption of formal methods of verification, restrictions on design methodology and the incorporation of configuration change controls.

In this regard a program of a given integrity cannot:

- Modify an object of a given integrity unless the program's integrity (and that of the data used by the program) is greater than or equal to that of the object.

- Execute an object unless the object's integrity is greater than or equal to that of the program executing it.

The integrity extension will inhibit a "low-integrity" Trojan horse (from an untrusted source such as USENET) from:

- Modifying an object of higher integrity (ie propagating its viral or destructive code)

- Executing a lower integrity object (i.e. invoking a privileged system operation)

This model, if correctly implemented, would resolve the Trojan horse propagation problem in the *Bell-LaPadula* model. The difficulty with such a system is that a formal authentication procedure must exist for **each** program introduced into a high integrity level. Objects tend to accumulate in high security and low integrity levels as a result of the application of the above constraints.

There exists, however, one final route by which a Trojan can propagate data in apparent violation of MAC security constraints - that of the **covert channel**.

Covert Channel Overview

A covert channel is a "hidden" or unspecified route over which information can be transmitted without the intervention of the security system. Numerous covert channels exist (with a variety of bandwidths) in the complex environment of modern operating systems.

Covert channels are divided into three categories, namely:

1. **Storage channels** (using common areas of memory or storage between processes)

2. **Timing channels** (using careful measurement of system performance)

3. **Denial of service channels** (by detecting forced denial of resources and services)

All covert channels result from deviations in the virtual machine model (in which each process active on the system conceptually has exclusive use of all system resources). These deviations may take the form of a shared data area linking two processes (possibly through the use of a variable maintained by the kernel which is accessible to both processes), changes in system throughput and process delays, or denial of access to system resources.

Examples of each type of channel might be:

1. **Storage** - an accessible flag in the kernel; a counter of processes created (or other system gathered statistics); file creation, destruction or manipulation in shared secondary storage; use of users' terminals to relay data between processes; data retained accidentally by server system processes.

2. **Timing** - contention for system processors; flushing of secondary storage caches; changes in paging and swapping behaviour of virtual memory; changes in disk access speeds... all detectable by carefully measuring the time taken by the system to carry out services on behalf of the processes.

3. **Denial of service** - through the exhaustion of fixed kernel data structures; disk capacity; resource contention and deadlock; and, in the extreme, crashing of the entire machine.

A simple timing covert channel can be described as:

Process **A** has access to classified information (SECRET) and contains a Trojan horse.

Process **B** has no privileged access and so can write data to public networks, and thus to the person introducing the Trojan.

A cannot write data to **B** directly because this would violate the MAC model.

So **A** decides to signal data slowly by running in a tight loop to signal a "1", or suspending for 1 second to signal a "0".

B is on the same processor, so it runs a timed code segment which takes 1 second with exclusive use of the processor. If it takes one second to execute then it knows that **A** is suspended, and has sent a "0". If it takes 2 seconds then it is contending with **A** for the processor, thus **A** has sent a "1".

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

Thus a slow, but possible, covert timing channel has been implemented.

Numerous routes exist for a Trojan horse utility to affect the system in a manner detectable by a non-privileged monitor process.

The data transfer rates of such covert channels are low in comparison with those in overt channels (i.e. disk or communications channels), varying from about 100 bits/second to fractions of a bit/second. The TCSEC requires that all covert channels of bandwidth > 0.1 bit/second be audited in B3 or greater systems. This rate represents a transfer speed of:

Per second	0.1 bit
Per hour	360 bits (45 bytes)
Per day	8.6 Kbits (~1Kbyte)

This small data rate may represent a significant compromise of data if the target file is classified or commercially sensitive (such as the corporate medium term plans).

The United Kingdom confidence levels [4] address covert channels briefly at UKL 4 evaluation (where it becomes a requirement to identify the classes of such channel which may exist), and at UKL 6 (where a requirement exists for a comprehensive search for all such channels).

Bell-LaPadula Model

Devised by D. E. Bell and L. J. LaPadula in a study for the *Mitre Corporation* in the United States in 1976, the *Bell-LaPadula* model proposes a formal set of access control rules.

The security model was originally devised for military systems and introduced the concept of **subjects** and **objects** whereby a system is described as secure only if the security clearance of the subject (system processes such as Read, Append, Write) is comparable with the classification of the object (files containing information). In order to determine whether or not access is permitted, the clearance of the subject is compared to the classification of the object and a determination is made as to whether the subject is authorised for specific access.

The *US Department of Defense* used the *Bell-LaPadula* model in devising its ‘Orange book’ criteria.

Secure Computer Systems: Unified Exposition and Multics Interpretation, MTR-2997 Rev 1., MITRE Corp., Bedford, Mass., USA. March 1976.

Estimation of Covert Channel Bandwidth

Various estimation formula have been proposed regarding the bandwidth of covert channels. In general a noiseless channel has a transmission rate of:

$b / (T_r + T_s + 2T_{cs})$ bits/sec

where:

b = bits of data transmitted by each state change implementing the channel

T_r= time for the receiver to read the channel

T_s= time for the sender to write the channel

T_{cs}=time for the operating system to switch between execution of the receiver and sender processes

Thus if a shared flag exists which takes 0.1ms to be changed by the sender, 0.1ms to be read by the receiver, and the operating system takes 0.5ms to context switch, then the data rate of the channel is 833 bits/second.

The throughput of a covert channel degrades rapidly in the presence of noise (generally caused by inadvertent interference by other processes, or by deliberate screening using random noise by the operating system). A recent paper [3] analyses a number of storage channels in Secure Xenix [TM], including:

1. Exhaustion of available space in a kernel/system table known as the *inode table*.

2. Creation of a new process and associated incrementing of a global variable holding the process identifier of the last created process.

3. A policy conflict channel in which the system will prevent a process from removing a directory until all files in the directory have been removed.

Simulation of these channels was carried out, and the bandwidths estimated at:

Inode exhaustion	16 bits/second
Process id channel	12 bits/second
Directory removal	2 bits/second

In the presence of other conflicting processes these rates would be reduced. These represent typical channels, although all three would be subject to audit (but not prohibition) by Orange Book C2 audit mechanisms.

Prevention of Covert Channels

The implementation of a virtual machine for each process effectively segregates processes, and thus prevents a covert channel from being exploited through the use of resource contention or exhaustion.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

This does however imply:

***** No user accessable global system statistics (i.e. process counts, disk activity, active users etc.).

***** No kernel structures to be available in user space (i.e. file pointers must be references to a mapping table which points to kernel structures).

***** No resource exhaustion routes (including strict per process quotas, and the spooling of exclusively acquired resources).

* Segregated filespaces for each compartment/classification.

***** No forms of handshake information to be returned to a process writing data to a “higher” classification object or stricter compartment, including no indication that a write may have failed, or that a file has been successfully opened.

***** Scheduling for each compartment and classification, in which when all processes suspend at a given classification a dummy “soak” process runs so that the variation in system load is not detected.

* Removal of known bugs.

If a covert channel cannot be removed it must be:

*** AUDITED** - with suitable warnings to the system security administrator if the channel is being utilised.

*** SUPPRESSED** - by introduction of random fake events and noise into the channel.

In Summary

Trojan horses can infiltrate high security environments in the absence of integrity extensions to the *Bell-LaPadula* model.

Once present they can effectively compromise data by utilising covert channel signalling methods.

Such methods cannot be eliminated easily but can be audited and suppressed to some degree.

References

[1] DOD Trusted Computer Security Evaluation Criteria, Dec 1985, DOD 5200.28-STD

[2] Shirley and Schell, Mechanism Sufficiency, Validation by Assignment, *IEEE Symposium on Security and Privacy*, 1981

[3] Tsai and Gligor, A Bandwidth Computation Model for Covert Storage Channels and its Application.

[4] CESG Computer Security Memorandum No.2, *Government Communications Headquarters.*

FOR MANAGEMENT

Specifically...

In last month’s *VB* we highlighted just some of the difficulties facing the developers of virus-specific anti-virus software.

One of the original arguments as to why such software could provide the best defence was that new samples of computer viruses could be distributed quickly to the research community and efforts coordinated in developing remedial software. In truth, there is little coordination, *if any*, of individual virus research efforts. Analysis is being duplicated and findings are rarely shared. There are also divisions and rivalries between research factions. In contrast, the commercial sector, with its traditional rivalries, is awakening to the fact that cooperation is increasingly necessary. This acknowledgement is dictated by the enormity of the problem in terms of the numbers of new viruses and their increasing complexity.

Lack of coordination and cooperation is an important factor inhibiting virus-specific software. However, as developments unfurl over the coming months, the most critical factors will prove to be the accelerating number of computer virus samples appearing and their increasing complexity. The Flip virus which we report this month (*Virus Dissection, pp.*18-20) is demonstrative of the problem. The virus requires disassembly and analysis in order to develop a suitable search ‘identity’. Second-generation viruses are adopting ever more sophisticated camouflage techniques necessitating ever more extensive and time-consuming analysis. There is already a significant back-log of work for researchers, and no signs that the bombardment is easing - quite the opposite.

If these factors are impeding simple scanning software, they may well curtail the development of even more specific ‘speed search’ and ‘disinfection’ software. Such programs require intimate knowledge of each virus - it is unlikely given the current climate of limited manpower, fragmented efforts and non-cooperation that any such program will ever be up-to-date or reliable. Moreover, a new philosophy which accepts deletion of infected files and restoring them from master copies or backups is taking hold. Only in exceptional circumstances is there a need to disinfect a program - the infection of a unique ‘in house’ development program (without a backup being available) is such a scenario. Then again, backups are now part of every PC user’s life..

To repeat a familiar message: **virus-specific software is for diagnosis only and provides no generic defence**

Such methods are always steps, if not strides, behind the virus writers. Forward thinking organisations have already adopted *integrity checking* methods for generic defence.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

FOR PROGRAMMERS

Fridrik Skulason

The Structure of Virus Infection - Part III: Boot Sector Viruses

As the name implies, boot sector viruses can infect the boot sectors of both diskettes and hard disks.

The boot sector contains code which is read into memory and executed whenever the computer is "booted" after being switched on. Normally the boot sector contains code to load the MS-DOS operating system, but this code may be replaced with other code, including a virulent program.

Floppy Disks

On a floppy disk the boot sector can be found at track 0, head 0, sector 1. This "physical" location is used by INT 13H calls, but it may also be referred to as *logical sector 0*.

Hard Disks

Hard disks contain two different types of boot sectors. This is necessary as one hard disk may be divided into several partitions.

Each partition contains one boot sector like the one described above, which is referred to here as the **Partition Boot Sector**. It is located at logical sector 0 in each partition, but the physical address (head, track and sector) depends on the size of the preceding partitions.

The other type of boot sector contained on hard disks is the **Disk Boot Sector**, which is always located at track 0, head 0, sector 1 of each hard disk in the system.

The Disk Boot Sector contains information about the partitioning of the disk, as well as code to locate and load the boot sector of the active partition. This applies even in the case of small hard disks, 20 Mbytes or smaller, which usually contain only one partition. Some boot sector viruses may not be able to infect hard disks at all, but those that do may infect **either** the Disk Boot Sector or the Partition Boot Sector.

Infected Sectors

It is often possible to identify infected boot sectors by visually examining a dump of the data contained there, as the customary messages may be missing and replaced with virus code. This is not always the case, however, especially not in the case of some of the latest boot sector viruses.

In some cases the virus is small enough to be stored entirely in the boot sector, but in most cases the code there is only a small "loader-type'' program, whose purpose it is to load the rest of the virus into memory and execute it.

Memory Manipulation

Most boot sector viruses hide at the top of available memory, reducing the amount of memory available to DOS by a fixed amount, usually 1 to 6 Kbytes. This memory is obtained by reducing the value stored at the memory location 0040:0013, which may for example result in a computer with 640 Kbytes of RAM appearing to contain only 636 Kbytes. Some legitimate programs may also reduce this value, so a suspiciously low value does not necessarily indicate a virus infection.

One virus, E.D.V., may hide above the 640K mark - it starts searching for free RAM at segment address E800H and moves downward, until it finds a free space to hide.

When a boot sector virus has installed itself in memory, it must intercept some interrupt function, in order to be activated later. Most viruses use INT 13H for this purpose, becoming active when a program attempts to do any disk I/O operation.

After the virus has installed itself into memory, it usually finishes by loading the original boot sector into memory and executing it.

Relocation and Allocation of Code

Most boot sector viruses are larger than the usual sector size of 512 bytes, and require more than one sector. The virus must also store the original boot sector somewhere, but where?

The following methods are used by the viruses known today:

***** Track 0
***** Last directory sector
***** A bad cluster
***** Last track
***** Extra track

Track 0

Most viruses infecting the Disk Boot Sector make use of the fact that track 0, head 0 is often unused, apart from sector 1. Hiding the rest of the virus code in these unused locations works well, except on the few computers where the disk is structured in an unusual way and this area is already in use.

The New Zealand (Stoned) virus is a typical case, but it infects floppy disks in a different way.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

Last Directory Sector

If the virus is small, perhaps needing only one sector in addition to the boot sector, the last sector of the root directory may be used. This will not cause problems, unless the root directory is almost full, but on a standard 360K diskette for example, the last sector is only used if the root directory contains more than 96 entries. This method is used by the Korea, New Zealand and PrintScreen viruses.

A problem may arise in the case of 3.5 inch disks or 1.2 Mbyte disks, as the last sector of the root directory is stored in a different sector, and a part of the FAT area may be overwritten instead.

Bad Cluster

A virus may also hide in any free sector on the disk, marking the corresponding cluster as "bad" in the FAT to prevent it from being overwritten. This is the method used by Disk Killer and the Italian virus. The drawback of this method, from the virus-writer's viewpoint, is a suspicious increase in the number of "bad" clusters, which is easily detected by running the DOS CHKDSK function.

Last track

As the last sectors of the last track are not used unless the floppy disk is almost full, this area is a suitable hiding place for viruses. The virus code may be overwritten, but it will not be noticed by CHKDSK. The Alameda (Yale) virus was the first to use this method, but now it is also used by the E.D.V., Form and Swap viruses.

Extra Track

Floppy disks normally contain 40 or 80 tracks, numbered 0-39 and 0-79. It is possible to format an extra track at the end - numbering 40 or 80. This track is not copied by the DOS DISKCOPY function, but makes an ideal hiding-place. The Ohio and Den Zuk viruses were the first to use this method, although they could only handle 360K disks. The method has gained popularity recently, and some of the most recent boot sector viruses, including V-1 (*described below*) use it as well.

Disinfection

While it is possible - **although highly inadvisable** - to disinfect programs with parasitic viruses active in memory, this is usually not the case with boot sector viruses. Many of them monitor *any* attempt to write to floppy or hard disks, and even if an anti-virus program manages to disinfect the boot sector, the virus may reinfect it *immediately*.

In line with all computer virus infections, booting the computer from a clean (virus free) write-protected system floppy disk is essential for successful recovery.

It is usually quite simple to remove boot sector viruses. **The disinfection program must locate the original boot sector and write it back to its original location** This can also be done with utility programs such as PC Tools or The Norton Utilities (see page 5). The virus code may be left on the disk, as it is now *inactive*, but preferably it should be erased by positively overwriting. One group of viruses requires a little extra work: If the virus hides by marking the clusters it occupies as "bad" in the FAT, the FAT should be corrected, releasing the space.

In the case of the Swap virus, automatic recovery is not possible, as the virus partially overwrites the original boot sector, but does not store it in unmodified form To recover programs and data from a Swap-infected diskette, just use COPY or XCOPY to copy everything to a clean diskette. The infected diskette should then be reformatted. Do **not** use DISKCOPY to copy its contents, as the virus would then be copied as well.

V-1, Flip and ''Multi-Partite'' Viruses

Finally, a new virus called V-1must be mentioned. It is remarkable because it is able to spread by boot sector infections, as well as infected COM files The boot sector of infected diskettes contains code to load the rest of the virus into memory. The rest of the virus code, located on track 40, contains code to infect COM files with the virus when they are executed. The virus also infects the Disk Boot Sector of hard disks, storing the rest of the virus code on track 0.

Hidden within the virus, although not on a sector boundary, is the original boot sector, so disinfecting a diskette infected with the V-1 virus involves reading the virus code into memory, locating the 512-byte block containing the boot sector and writing it back.

Infected COM files, once identified, can be deleted using the DOS DEL command and restored from write-protected master software. A disinfection routine is also possible, as the virus just overwrites the first 7 bytes of the COM file, stores them, and then appends the 1253-byte virus code to the end.

The other virus which combines boot sector and program infection techniques is Flip which is the subject of this month's virus dissection, on page 18. A further virus called Anthrax is also reported to employ these tactics.

This ability to infect multiple system and program elements has given rise to the term '*multi-partite'* infection.

Disinfection of the common New Zealand virus which infects the Disk Boot Sector of the hard disk is described on page 5. Standard 'brute force disinfection' of both parasitic and boot sector viruses was described in VB, July 1990, pp 3-5.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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 DISSECTION

Richard Jacobs

Short Sharp Shock - The Baffling 'Tiny' Virus

The 'Tiny' virus (aka '163') diverges from the current trend towards sophisticated 'second generation' viruses. As the name suggests this virus is just 163 bytes long and is one of the shortest viruses seen so far. Since its discovery two months ago, a family of 'tiny' viruses, ranging in length from 158 to 198 bytes, has been reported in Bulgaria. It seems that a challenge to write the shortest operating computer virus is under way.

Tiny's code contains just 59 instructions. It is *extremely* simple in its coding and contains no side effects, no encryption or clever hiding mechanism - not even the ability to change drive or directory. The virus' sole purpose is self-replication.

Tiny is a non-resident parastic virus infecting COM files only. Every time an infected program is executed, one new COM file in the current directory is infected. Re-infection is avoided by checking a 2 byte flag stored within the virus. The virus does not make itself memory-resident or affect normal operation of the computer in any way, other than infection.

The viral code is simplicity itself. When an infected program is executed, control is transferred immediately to the virus. The virus then searches for the first COM file in the current directory; if there are none, control is returned to the parent program, otherwise the file is opened for both READ and WRITE instructions. The first 3 bytes of the file are then read in to the end of the virus. The first of these bytes is compared to the JMP instruction (E9H) and if it does not match, the virus searches for the next COM file. The second and third bytes read in are stored to provide the address to return to the main program after the virus has finished. These 2 bytes are also used to give the address of the re-infection flag. A further 2 bytes are then read in and are compared to 0807H, if they match this value, then the virus searches for the next COM file in the current directory.

The length of the COM file is checked next, to ensure that there is enough space for the virus, as a COM file must be less than 64 Kbytes long. Again, if there is not enough space the virus searches for the next COM file. The virus then copies itself to the end of the COM file, alters the offset of the JMP instruction at the start of the file to point to the start of the virus, closes the file and jumps to the start of the original file.

The virus is a tidy block of code that checks both for previous infections and whether or not a file can be successfully infected, without corruption. However, the handling of files is careless and does not close files that have been opened, but not infected, i.e. files that have already been infected and files that are too large, or do not start with a JMP instruction. This may eventually lead to errors as no checking is performed to find out whether or not a file has been opened succesfully; DOS only allows a single process to have a limited number of files open at any one time.

Disinfection may be performed simply and safely by deleting all infected files and replacing them with backup copies.

A Possible Explanation

It is unlikely that this virus was ever meant to be released into the wild. It neither displays or contains any messages and does not intefere with the normal operation of the computer. There seem to be three possible explanations for this virus:

1) *It was written to demonstrate or prove self-replicating code.*

2) *It is a prototype (or 'beta test') for a more sophisticated virus, which would have unspecified features added to its code.*

3) *It is a bid to be the most compact computer virus.*

The third explanation, intriguing as it is, appears unlikely as certain features could be removed safely (thus decreasing program length) without disabling the program's virulence.

The second explanation appears to be the most likely, as another virus, Kennedy, appeared at around the same time as Tiny and is based on the same code. Kennedy is a considerably more sophisticated virus. It does not alter the date/time stamp or the attributes of an infected file and it tries to infect all COM files in the current directory. It does not infect COMMAND.COM. The Kennedy virus is believed to have originated in Denmark. The programmer of the Kennedy virus makes reference to the San Fransisco based anarchic rock group *The Dead Kennedys*.* The Kennedy virus activates on June 6th, November 18th and November 22nd of any year. November 22nd is the anniversary of the assassination of John F. Kennedy, the other dates refer to the deaths of Robert and Joseph Kennedy.

Despite differences in the operation of the two viruses, sufficient parts of the Tiny virus can be seen throughout the Kennedy virus, to suggest that the Kennedy virus was written as a development of the Tiny virus (*or vice versa, Tech Ed.*).

A final note, pertaining to such compact and simplistic code, is that this particular virus is *extremely* easy both to disassemble and modify thus increasing the probability that 'hacked' versions (possibly destructive in nature) will appear.

** The connection between rock music (particularly heavy metal) and adolescent computer misuse has been observed and noted - the attraction of these phenomena and there inter-relation would be a suitable subject for a psychological examination. Ed.*

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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 DISSECTION 2

Jim Bates

FLIP - A Professional's Handiwork?

It has long been expected that a composite boot sector/parasitic virus would appear. The FLIP virus is one of three current viruses which operates in this way, the other two being Anthrax and V-1.

The code seems to have been written by an experienced programmer and obviously took time to develop. The trigger routine affects only systems with EGA or VGA monitors and produces a mirror image effect in screen modes 2 and 3, flipping the display horizontally and showing a modified font which reverses each character.

Disturbing new trends are evident within the code, particularly a number of routines which attempt to circumvent memory-resident monitoring software or pervert the actions of a particular virus detection program

The code appended to files has a simple encryption algorithm but that written to the Disk Boot Sector and elsewhere on the disk is **not** encrypted and is easy to recognise. **Apart from the instance mentioned above, FLIP makes no attempt to protect itself from detection, although the randomisation of the position of the decryption routine does prevent the extraction of a recognition pattern for the parasitic code** File checking programs however, will be able to recognise the file changes introduced by infection.

Infection Routine

There are three paths by which this code can be executed, infected COM type files, infected EXE type files and the Boot code. The virus does **not** check file names, so renamed COM or EXE files are still processed correctly according to their type. Subsequent references here to COM or EXE files should therefore be considered as COM (or EXE) type files.

Parasitic infection is invoked via an interrupt handling routine which intercepts the LOAD and EXECUTE function request. This means that files with other extension names (BIN, OVL etc.) could become infected although overlays are specifically excluded. COM type files are only infected if they are less than 62856 bytes in length, and once infected they take no further part in the virus operation or replication processes.

When an infected COM file is run, the virus code is decrypted and the first few instructions are executed to repair the file header. Processing then returns directly to the host program regardless of whether the virus has been installed or not.

This infection of COM files with totally ineffective code appears to be deliberate and may indicate an intention to add "improvements" at a later date

When an infected EXE file is executed, a special "are you there?" call is issued to determine whether the virus code is resident and active in memory. If the virus code responds, program execution is transferred to the host program and no further virus code is executed. This special call consists of placing 0FE01H into the AX register and issuing an INT 21H request. If the virus is resident, the AX register will contain 01EFH when the interrupt returns.

If the virus is not-resident, a check is made to ensure that enough memory is available for the virus to be installed and then the BIOS top of memory pointer is modified before the virus code is installed into high memory. This method therefore avoids using any of the system TSR services.

Once the code is relocated, the Disk Boot Sector of the first physical fixed disk is checked to see if it is infected. This check consists of examining the word at offset 28H in the Disk Boot Sector (Track 0, Head 0, Sector 1) for a value of 01FEH.

If the Disk Boot Sector is not infected, the partition table is searched for the first partition of type 1, 4 or 6 - these are standard partition types and on most machines the first (possibly only) partition will be type 4 (DOS - 16 bit FAT). Once found, the settings of this partition are checked to ensure that infection is possible and then modified to allow hidden storage of the virus code.

Most boot sector viruses hide their additional code in available sectors on the hard disk marked as "bad" so that DOS will not use them. This virus adopts a different technique.

Within the partition table there are pointers to the physical limits of each partition. These indicate the absolute track, head and sector addresses of the start and end of each partition. This virus subtracts 6 from the value of the sector address which points to the end of the partition. This effectively reduces the size of that logical drive by 6 sectors (around 3 Kilobytes) and leaves 6 sectors "in limbo" beyond the end of drive. The first of these sectors is used to contain an uninfected copy of the Disk Boot Sector (but still with the modified partition table) and the remaining five contain the virus code.

The original Disk Boot Sector is then infected and written back to the disk. The final stage of this section is to hook the virus's own INT 21H handler into the system and then processing returns to the host program.

A system infected by execution of an EXE file will **not** display the trigger effect even if the appropriate video adapter is available and the date and time are right. This is because an inhibit flag is set which can only be cleared when the machine is booted on the correct date.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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 amount of time wasted in writing this virus is phenomenal and the programmer displays considerable experience in certain areas."

Boot Sequence

The boot sequence on a machine with an infected hard disk proceeds as follows:

After the normal initialisation of the Stack Segment and Stack Pointer registers, the BIOS top of memory pointer is modified to allow 3 Kilobytes of space above available memory. The virus code is then read from the disk by reference to the partition end-address and using the BIOS INT 13H service.

Processing then transfers into the Hi-Mem copy of the code and continues by loading the "clean" copy of the Disk Boot Sector into the boot area at 7C00H.

The attached video facilities are then checked using BIOS INT 10H and if EGA/VGA capabilities are found, the system date is checked to find the day number. If the date is the second of any month, then the inhibit flag is cleared and a further 4 Kilobytes of high memory are allocated. This area is then filled with a bit-reversion copy of the EGA character set and the appropriate access pointers are prepared. If the video adapter or the date fail the checks then these routines are not executed and the inhibit flag is set before processing jumps to the final installation stage.

This involves hooking interrupt handling routines for INT 1CH, INT 21H and INT 9FH.

The INT 21H vector is uninitialised at cold boot time but the intention is to insert the virus's handler address and collect the existing contents for comparison within the INT 1CH (Clock Tick) handler. The INT 9FH vector is a user defined interrupt and the handler is not used by the virus code, this will be discussed later.

Boot processing is then transferred to the "clean" Disk Boot Sector at 7C00H. All of the boot sector viruses that I have examined gain processing time by hooking their own handler into the BIOS services (usually INT 13H - disk access). This virus hooks into DOS services even though they do not exist at boot time. It uses the INT 1CH service to gain initial processing time and thereafter swaps its attentions once the system INT 21H service is detected as having been installed.

Subversion of TSR Anti-Virus Monitors

One of the most worrying aspects of this virus is its use of an interrupt "strip" mechanism which can examine the relevant chain of vectors and strip out all those installed after the system has loaded.

This stripping process makes use of the DOS single step interrupt facility whereby execution of the routine pointed to by the INT 01 vector is forced by the hardware after **every** program instruction if a particular flag (the TRAP flag) is set. Thus throughout the chain of probably several thousand instructions within an interrupt service, the single step handling routine can examine the state of the processor registers (particularly the code segment register via the return address on the stack) on a continuous basis.

This facility enables the virus code to use unmonitored services thereby gaining access to the system "underneath" any memory-resident anti-virus software.

The extreme reduction in speed that would result from the execution of all these extra system calls is avoided by having the single step routine turn itself off once the original system service vector has been located.

It is not difficult to produce resident anti-virus software which is immune to this sort of subversion but I suspect that there are few, if any, packages which currently do this.

Targeting of a Specific Anti-Virus Program

Another disturbing feature of this particular virus is its apparent targeting of a specific anti-virus scanning program. Unfortunately, there is no way within the code of knowing what the name of this program is.

During the boot infection process a flag is set and then cleared to indicate the correct completion of the virus-write routine. If this flag is not cleared, the virus continues to function but includes an extra routine which checks the contents of the target file for a distinct pattern.

If this pattern is found, at the point in the file where it occurs, two bytes are inserted which will become an INT 9FH instruction when the file code is executed. INT 9FH is one of the interrupt vectors provided by DOS for user definition and in this case the virus will already have installed and activated such a routine. The insertion of these two bytes is in addition to the subsequent execution of the normal parasitic infection routine. The INT 9FH routine, executed when the affected file runs, accesses the program's code and data areas by looking back along the stack.

Various changes are made including the testing of a data item for the value 1FH - which happens to be the file time infection marker used by this virus. Other data is modified by having the length of the virus subtracted from it.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

Without access to the original program it is impossible to be certain of the exact effects of these changes but after analysing them it is reasonable to assume that the program contains some sort of checking routine which is looking for virus signatures and the changes will prevent modifications introduced by **this** virus from being discovered.

This targeting of an anti-virus program is extremely specific but the target program may well be in widespread use. Both the original search algorithm as well as its representation within the program are published here in the hope that someone will recognise it and come forward with the details. In this way, the vendors of the protection software can be contacted and informed of exactly how their product is being modified. (*See Figure 1*).

```
SEARCH:
     CMP           WORD PTR [DI],168BH
     JNE           NOTFOUND
     CMP           WORD PTR [DI + 4],1689H
     JNE           NOTFOUND
     CMP           WORD PTR [DI + 8],168BH
     JNE           NOTFOUND
     CMP           WORD PTR [DI + 0CH],1689H
     JNE           NOTFOUND
     ........
     ........
     ........
     ........
NOTFOUND:
     INC           DI
     CMP           DI,SI
     JB            SEARCH
```

Figure 1. Search algorithm used in the Flip virus to targeting an anti-virus program, but which one?

This search routine is started after the program has been loaded into a buffer and with SI containing the full length of the program and DI containing zero.

This pattern is consistent with a program listing as follows:

```
     MOV           DX,[Dat1]
     MOV           BX,[BX + DI]
     DB            ?,?
     MOV           DX,[Dat2]
     MOV           BX,[BX + DI]
```

where the memory locations Dat1 and Dat2 are unknown and the two bytes noted with question marks are also unknown.

Technical Summary and Detection

Summing up the various properties of this virus:

***** COM type files are infected but do not actually run the virus code. If an infected COM file is run on a clean machine it does **not** infect anything. [1]

***** Running an infected EXE file will install the virus into memory and attempt to write the Disk Boot Sector (boot) infection to the first hard disk.

***** No trigger will occur even if the date and time are correct, since the system date is only checked during the infected boot process. Thus only machines with a battery clock (as well as EGA/VGA facilities) will be able to display the trigger effect. The installed virus however **is** capable of infecting other files.

***** Booting a machine with an infected fixed disk will install the virus and will display the trigger effect between 16:00 and 16:59 on the 2nd day of every month.

***** The virus code installed via the boot routine is exactly the same as that introduced by EXE parasitic action, and the boot infection only infects the first hard drive. This means that although this virus has composite features, it can only spread from machine to machine via infected EXE files.

* The infected Disk Boot Sector contains the first 66 bytes of the virus code in unencrypted form and the following sequence of bytes at offset 2EH into the Disk Boot Sector will identify an infected drive:

```
33DB 33FF 8EC3 2629 0613 04CD 12B1 06D3
```

***** The infective length of this virus is 2,343 bytes for both EXE and COM files and infected files are marked by having the time field of their directory entry set at 1FH (= 62 seconds).

Conclusions

This virus is representative of a new series of viruses which employ radical infection methods and which target anti-virus software. The ability to infect multiple system and program elements gives rise to the term *'multi-partite virus'*. Fortunately it is no more difficult to detect than most other viruses and modifications to memory-resident virus monitoring programs will make them immune to the "interrupt stripping" technique which the Flip virus employs.

The amount of time wasted in writing this virus is phenomenal and the programmer displays considerable experience in certain areas. It would appear that some large organisation has underworked programmers, one of which is using company time to write viruses. *How many large organisations actually check just what their programmers are up to?*

[1] Running an infected COM file in tests has been reported to cause infection in some cases. The cause for this discrepancy is not known but an explanation will be published in the October edition of *VB*.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

MAC THREATS

'Dirty Mac Brigade' - A New MDEF Strain and the Advent of CDEF

A new virus and a strain of an existing virus are added this month to the list of Macintosh threats.

A new strain of the MDEF virus has appeared. *VB* reported the original virus in July of this year after its discovery at *Cornell University*, New York on May 16th. The new strain contains no malicious code but adds MDEF resources to system files and applications. In advanced or custom mode, *Symantec's* SAM 2.0 anti-virus for the Macintosh will alert the user to attempts to change or add MDEF resources and thus stop the virus from spreading. Please note that *Symantec* call the virus Garfield.

Microcom Software Division, the distributor of the commercial Virex Macintosh anti-virus program, refer to this virus as MDEF 2.

The latest Virex release, version 2.8, combats both the new strain of MDEF and the CDEF virus reported below. Virex 2.8 also identifies and combats a very rare virus called Frankie which attacks Amiga computers while emulating Macs.

The following virus definitions can be added to Virus Clinic to identify and specify infection by this virus.

To detect this specific strain of MDEF, *Symantec* advise the following definition:

Virus Name:	Garfield
Resource Type:	MDEF
Resource ID:	0
Resource Size:	532
Search String:	2F3C4D4445464267487A(hex)
Search Offset:	304

A second definition will detect both existing strains of MDEF. (and delete any earlier MDEF definitions entered). Scanning times will be marginally increased. It is probable that this definition will be capable of identifying future MDEF strains:

Virus Name:	Garfield
Resource Type:	MDEF
Resource ID:	0
Resource Size:	Any
Search String:	A9A92F0CA9AA2F0CA9B0(hex)
Search Offset:	Any

CDEF

CDEF is similar to the widespread WDEF virus (*VB*, *January 90*).This new virus contains no destructive or malicious routines but causes frequent system crashes. It can also infect hard drives immediately upon the insertion of an infected diskette. It adds CDEF resources to desktop files only. CDEF will **not** spread if SAM 2.0 is running even in the Basic level. The 'Desktop Guardian' feature prevents code in desktop files from executing while the Finder is running.

If SAM is configured to standard level or higher, it will alert the user of a CDEF infection when the desktop file is opened. A "Code in desktop file (CDEF)" alert will be issued. Preempting the opening of the infected desktop file causes the Finder to rebuild the desktop and thus eliminates the virus. The following virus definition should be added to Virus Clinic:

Virus Name:	CDEF
Resource Type:	CDEF
Resource ID:	1
Resource Size:	510
Search String:	454633C0001487A0046A9AB (hex)
Search Offset:	420

Macintosh Anti-Virus Utilities

SAM, *Symantec UK*, NKA House, 36 King Street, Maidenhead SL6 1ES, UK. Tel 0628 776343. Commercial. Updated regularly.

Symantec Corp. 10201 Torre Avenue, Cupertino, CA 95014, USA. Tel 408 253 2167.

Virex, *Microcom UK*, 2D Dukes Court, Duke Street, Woking, Surrey GU21 5EH. Tel 0483 740763. Commercial. Updated regularly.

Microcom Virex Group, PO Box 51816, Durham, North Carolina 27717, USA. Tel 919 490 1277.

Disinfectant, *John Norstad*, Northwestern University, 2129 Sheridan Road, Evanston, Illinois 60208, USA. Shareware, free. Available from many user groups and major bulletin boards including Genie, CompuServe and Internet archives. Updated regularly. A comprehensive manual to this program is available.

VirusRX, *MacUser Userware*, PO Box 320, London N21 2NB, UK. Shareware, free. Distributed by Apple Computer and available from Apple retailers and bulletin boards.

Interferon, *MacUser Userware*, PO Box 320, London N21 2NB, UK. Shareware, Charity Donation.

Virus Detective, *Jeffrey Schulman*, PO Box 50, Ridgefield, CT 06877, USA. Shareware, $25.00. Available from bulletin boards. Updated regularly.

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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 EVALUATION

Dr. Keith Jackson

VIRUSCAN and Associated Utilities

This review was initiated by a different process than is normally used for the products reviewed in *VB*. The manufacturer of an anti-virus product is usually keen to provide a copy for review purposes, but as I often use VIRUSCAN (and mention it in comparative reviews), one manufacturer complained that I had not put it through the same testing process that his own product had been subject to. This seemed a fair point and the following review aims to rectify this omission.

Worldwide Distribution

McAfee Associate's VIRUSCAN program inspects disks, or entire systems, and identifies virus infections by searching for virus specific patterns. VIRUSCAN only works on standalone PCs, other products from the same developer (not reviewed here) are available for networks. VIRUSCAN is updated very frequently (every couple of weeks), and the new files are distributed around the world, in compressed form, via various electronic conferencing systems and bulletin boards. A validate program is included so that all files can be checked to see if they have been altered in transit. The version of VIRUSCAN used for this review was the latest available (*see Technical Details section below*). For obvious reasons, VIRUSCAN is usually referred to as SCAN.

Documentation and Search Routine

The documentation that accompanies VIRUSCAN is contained in a README file. This is the weak point of the whole package. The documentation contains no table of contents, no index, indeed very little structure at all. It does, however, have a fair description of how to use VIRUSCAN. An associated file provides technical details of all the viruses currently known to VIRUSCAN.

Known viruses infect the hard disk partition table, the DOS boot sector of hard disks or floppy disks, and executable files. VIRUSCAN checks all of these areas as well as inspecting memory for viruses already lying in wait. The executable files may be operating system programs, system device drivers, COM files, EXE files, overlay files or any other file which is loaded into memory and executed. VIRUSCAN can check the entire system, an individual disk, a sub-directory or an individual file. If a virus is found, the name of the infected file or system area is displayed, along with the name of the identified virus.

LZEXE Scanning

VIRUSCAN will perform both an internal and an external scan on programs that are dynamically compressed with LZEXE (*see VB June 90*). The compressed file will first be scanned in its raw form, then decompressed and scanned again for an internal infection.

Speed of Execution

The documentation states that VIRUSCAN requires "approximately 3 minutes of run time for each 1,000 files". This figure can only be an order of magnitude estimate. On my test computer VIRUSCAN tested 570 files in 11 minutes 36 seconds, much slower than the quoted search rate. On a faster 386 computer (*see Technical Details section below*), VIRUSCAN inspected 976 files in 3 minutes 18 seconds. Therefore the stated rate of 1,000 files inspected in 3 minutes is indeed attainable, but only on fast computers.

In last months review I quoted VIRUSCAN as searching my hard disk in 4 minutes 55 seconds. The hard disk content has hardly changed in recent weeks, so this new version of VIRUSCAN must be somehow different. In some ways VIRUSCAN is a victim of its own success; it now searches for far more viruses (129 unique viruses, 213 variants in total), and also scans all overlay files. The previous version only searched for 86 viruses, and an unstated number of variants. VIRUSCAN is therefore encountering the inherent limitations of anti-virus products that work by searching for patterns; the total number of viruses is expanding rapidly, and the amount of time spent checking a disk is (and always will be) in direct proportion to the number of viruses.

Detection Rate

The complete list of viruses known to VIRUSCAN is contained in a file which accompanies the explanatory documentation. This file describes each virus, and states the number of known variants of the virus. There are currently 129 unique viruses identified in this file. The number of known viruses is now very large, and expanding apace.

The standard *VB* set of test viruses contains 49 unique viruses, with variants on these viruses increasing the number of test samples to 101. This comprises 99 parasitic viruses, and two boot viruses. The specific viruses used for testing are explained in the *Technical Details* section below. VIRUSCAN correctly detected 97 out of the 99 parasitic viruses, and correctly detected both boot viruses. Given that the version of VIRUSCAN mentioned in last month's review only detected viruses in 86 of the 99 test files, this is a dramatic improvement.

The Lehigh and Virdem viruses were the only test samples that VIRUSCAN did not detect as being virus infected. There were, however, some minor problems of misidentification (possibly

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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.

problems of nomenclature). Samples of the test South African virus were identified as being a variation of the Icelandic virus. This confusion has been caused the *VB* test sample being incorrectly named. Samples of Virus-B were identified as Friday the 13th. Most interestingly the file containing a sample of the 1260 virus was thought by VIRUSCAN to be infected by both the 1260 virus and the V2P2 virus (their notation). Given that both of these viruses are encrypted with a random key, and have no unique identification string, some confusion is not surprising.

The excellent detection rate shown by VIRUSCAN, and its frequent rate of update, are the main reasons why I use this program. It says a lot for VIRUSCAN that it can detect 99 out of 101 viruses from a third party's test samples. Any scanning program will (should) detect 100 percent of its own test samples, the hard part is to achieve a high rate of success when virus samples are from other sources. I believe that VIRUSCAN achieves this.

Related Programs

Other 'sister' programs are available from the developer of VIRUSCAN, and they are usually all updated simultaneously. They include:

CLEAN-UP, a program that removes viruses, and attempts to repair or delete infected files. CLEAN-UP identifies the virus which is to be removed by means of a virus ID (a short name) provided by VIRUSCAN. Therefore VIRUSCAN must be executed first to find out which virus is present (if any).

VCOPY, a replacement for the MS-DOS COPY command that checks for viruses as it copies files. VCOPY implements all of the functions available in the MS-DOS version 3.3 COPY command. VCOPY will also search executable files that have been compressed with LZEXE (*see VB June 1990*).

VSHIELD, a memory-resident program that prevents viruses from entering a computer system by monitoring and scanning programs as they are loaded. VSHIELD claims that it does not cause false alarms. This claim is impossible to test thoroughly. I could not find any occasion when VSHIELD erroneously stated that a file was infected, but this does not prove that such events will never occur.

Given my comments in last month's review about the lack of technical information provided about how a given memory-resident program operated, I feel obliged to point out that the problems that may be encountered when loading VSHIELD into memory, the difficulties that can be encountered on a network, the overheads added to the file loading process, and the possible loss of data when cache programs are operating are all thoroughly discussed in the documentation. VSHIELD requires between 3 and 25 Kilobytes of system memory, depending on how it is configured, and is incompatible with WINDOWS/386.

Conclusion

In conclusion, I find VIRUSCAN a useful and reliable tool. It is shareware, and many a barbed comment has been aimed at shareware in the past, but this method of distribution greatly facilitates the frequent update rate that is necessary for all scanning programs. The documentation is poor, but the technical content of the software is excellent. VIRUSCAN looks for viruses swiftly, and detects more viruses than most competitor programs.

Technical Details

Product: VIRUSCAN

Developer (and Vendor): McAfee Associates, 4423 Cheeney Street, Santa Clara, CA 95054, USA. Tel: (voice) (+1) 408 988 3832, Fax: (+1) 408 970 9727, BBS: (+1) 408 988 4004.

Availability: IBM PC/XT/AT, PS/2, or compatible running MS-DOS. The documentation says nothing about operating system versions.

Version Evaluated: 4.5V66-B, dated 13th August 1990. Version 65 of VIRUSCAN did not appear, as a Trojan version 65 was found on a few bulletin boards in March 1990. This version of VIRUSCAN logically follows version 64.

Serial Number: None (Shareware)

Price: The registration fees (in US dollars) are: VIRUSCAN=$25, CLEAN-UP=$35, VCOPY=$15, VSHIELD=$25. Disks are not normally sent when the registration fee is received, add $9 if a disk is required. Site licences are available.

Hardware Used: ITT XTRA (a PC compatible) with a 4.77MHz 8088 processor, one 3.5 inch (720K) floppy disk drive, two 5.25 inch (360K) floppy disk drives, and a 30 Mbyte Western Digital hardcard, running under MS-DOS v3.30. Also a Toshiba 3100SX laptop portable with a 16MHz 80386SX processor, one 3.5 inch (1.44M) floppy disk drive, and a 40Mbyte hard disk, running under MS-DOS v4.01.

Viruses Test Suite: This suite of 49 unique viruses (according to the virus naming convention employed by *VB*), spread across 101 individual virus samples, is the standard *VB* test set. It comprises two boot viruses (Brain and Italian), and 99 parasitic viruses. There is more than one example of many of the viruses, ranging up to 10 different variants in the case of the Cascade and Vienna viruses. 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 *Virus Bulletin*:

1260, 405(2), 4K(2), AIDS, Alabama, Amstrad(2), Anarkia, Brain, Cascade(10), Dark Avenger(2), Datacrime(3), dBASE, December 24th, Devils Dance, Eddie(2), Fu Manchu(3), GhostBalls, Hallochen, Icelandic(2), Italian, Jerusalem(6), Kennedy, Lehigh, Macho-Soft, MIX1(2), Number of the Beast, Oropax, Perfume, Prudents, PSQR, South African(2), Suriv(8), Sylvia, Syslock(2), Taiwan, Traceback(4), Typo, Vacsina, Valert, Vcomm, Vienna(10), Virdem, Virus-90, Virus-B(2), VP, W13(2), XA-1, Yankee(5), Zero Bug,

VIRUS BULLETIN **©1990 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139. /90/$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

The Japanese virus problem is evidently worsening. *NEC Japan* has issued an official warning to users of the PC-9800 series of computers about a ‘‘Merry Christmas’’ virus. A number of machines were delivered from the factory infected by the virus. If a file is dated 25 December, the screen displays ‘‘Merry Christmas to You!’’. Infected programs greater than 30,720 bytes are destroyed. In addition to the *Sharp Corporation* incident reported on page 10, *Fujitsu* has revealed that in May a virus was found on twenty on-site microcomputers. Official Japanese concern about these and earlier incidents has led to the drafting of guidelines by the *Ministry of International Trade (MITI)* and the establishment of a reporting office in the *Information Promotion Agency (VB, July 1990)* . In July, Japanese officials visited researchers and anti-virus software developers in Europe and the United States to familiarise themselves with the technical aspects of computer viruses and the procedures and software to combat them.

Symantec are to release **Norton Antivirus** for the IBM PC and compatibles on 13 September. The company is making bold claims for their memory-resident scanning and checksum program which is described as “the fastest and most complete virus protection utility on the market”. According to UK chairman Michael Skok, Norton Antivirus will scan a 40 Mbyte hard disk in 26 seconds. A Virus **Newsline service**, including a monthly **Virus Journal** are also envisaged. UK Tel 0628 776343, USA Tel 408 253 2167.

The Sophos Ltd product **Vaccine** has been awarded UKL1 certification from CESG, the *Communications Electronics Security Group* of GCHQ. It is the first anti-virus product to receive UK certification. Vaccine was also the recommended ‘best buy’ in the *Which Computer?* survey of security software, July 1990. Tel 0235 559933.

Microcom are offering a “two for the price of one” software deal on the **Virex** (Macintosh) and **Virex-PC** anti-virus packages. The offer closes in November 1990. The Virex-PC package, (developed by Ross Greenberg - author of Flushot +) will be the subject of a *VB* technical evaluation in October. UK Tel 0483 740763, USA Tel 919 490 1277.

S&S Ltd, UK, is holding a two day **seminar on data recovery** (4-5 October) and **a seminar on the virus threat** (8-9 November). The company has also released **QFV** (Quick Find Virus), a virus scanning program which employs VIRTRAN (Virus Transaction Language) to speed search times. Tel 0494 791900.

Information Systems Integrity & Security Ltd (ISIS) is a **new British company** specialising in countering computer virus outbreaks and enhancing system security. Consultancy, on-site disinfection and programming contract work in UNIX, Macintosh and DOS security are among the services available. Tel 0831 223 120. EMAIL:isis@cs.hw.ac.uk

VIRUS BULLETIN

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

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, of 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.