ISSN 0956-9979 OCTOBER 1993

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Protected Mode Boot Sector Viruses.** A sample of a protected mode boot sector virus has been sent to *Virus Bulletin*. The virus, PMBS, uses protected mode to give it complete memory and disk stealth. Will this change anti-virus software significantly?

• **Costing a virus attack**. *Rockwell International* goes 'on the record' in a frank discussion of the cost of a typical virus attack within its corporate environment.

• **Mutation Engine virus now in the wild**. In two separate confirmed reports, it is now clear that the Coffeeshop virus, which uses the Mutation Engine for encryption, is spreading in the wild.

CONTENTS

EDITORIAL

Kto Ne S Nami, Tot Protiv Nas...

Complaining bitterly about the actions of the virus authors and computer underground has become something of a *Virus Bulletin* editorial trademark, and rightly so: the journal has no wish to add in any way to the folk hero status of the hacker or virus writer. Unfortunately, such vituperations can be counter-productive, not to mention repetitive, and by and large the subject does not really contain enough meat for a satisfying editorial... unless something particularly raises the Editor's ire. The cause of this month's bout of righteous indignation is the latest edition of the so-called 'virus researcher's magazine', *Computer Virus Developments Quarterly.*

"Bug-free computing is difficult enough to achieve even without the aid of hackers..."

The press has already played an important role in shaping the way in which computer viruses have developed. The books by Ralf Burger have helped to make the much-hacked Vienna virus prevalent, and publications like *40Hex* have popularised techniques for writing more complex viruses. One of the latest journals of this ilk to grace the world's news stands is Mark Ludwig's *Computer Virus Developments Quarterly* - a publication which purports to give the true independent view on the virus problem. Another independent view is never a bad thing - except that Ludwig believes in explanation by example - and if his readers are incapable of typing his 'examples' correctly, a disk is available to aid them.

To give the reader an example of the type of material disseminated by Ludwig and his cronies, consider the latest issue of *CVDQ*. The main thrust of the journal this quarter is to do with the SS-386 virus (also known as PMBS, see page 9), but there are other items of interest, including a 'guided tour of VX BBS's - with phone numbers!' and the results of *'The First International Virus Writing Competition*'. Is this really the sort of material which should be freely available?

Ludwig inevitably argues that he has every right to publish *CVDQ*, and in this particular case, could argue that the SS-386 'virus' is not fully functioning and therefore holds no threat for either users or anti-virus software developers. Burger, too, claimed that he purposely introduced mistakes in the Vienna source code published in his book. Four years later there are some 200 variants in existence. Such deliberate mistakes are no defence: by tutoring his readers, Ludwig is actively encouraging them to write more sophisticated computer viruses.

This claim to legitimacy of virus research by virus writing is puzzling. If someone broke into a house, opened the filing cabinets and shredded every piece of paper, the owner would be outraged. Why is the sense of violation any less when the damage is done to computer data? If the arguments for this 'proof by example' are so compelling, then the world is very fortunate that Mr Ludwig is not attempting to illustrate the dangers of terrorist activity or explosives.

The role of the computer within society is growing more important by the day. Recent events within the UK have only served to underline this, with growing industry concern over the safety of the software controlling the Sizewell B nuclear plant. Bug-free computing is difficult enough to achieve even without the aid of hackers, virus authors and other assorted miscreants - it cannot be in anyone's interest to make the job of the computer vandal any easier.

Of course, none of this is new - both sides of the 'should we/shouldn't we publish virus code' battle feel that they have captured the moral high ground, and the arguments for each case have been flogged to death. However, what makes this particular issue so irksome is the comparative silence of those users who object to these activities but refuse to make their voice heard.

There is no apology if this places much of the blame on the average computer user - why should those who mumble quietly about the iniquity of computer law stand back and let others fight their battles for them? With the stance of the computer underground now much more clearly defined, one can do no better than to quote Lenin: 'He who is not for us, is against us.' The middle ground in the argument is rapidly disappearing, and those users who remain silent are adding their tacit support to the gradual legitimization of the computer underground. It is time to stand up and be counted.

NEWS

The Virus Bulletin Book

Over the last year, *Virus Bulletin* has received a record number of calls asking for information from some older back issues of *VB* (such as data on the Cascade virus). In addition to this, the market has been lacking good information, pitched at a level which the average computer user can understand and, most importantly, use.

Virus Bulletin is pleased to rectify this situation by publishing *The Survivor's Guide to Computer Viruses*. The book, edited by Victoria Lammer and available from *Virus Bulletin*, comprises over three hundred and fifty pages of essential information on computer viruses, anti-virus software, and anti-virus procedures.

Information included in the book includes a history of computer viruses, a tutorial on viruses and how they work, and a chapter on good anti-virus procedures, before embarking on examination of the twenty of the most important viruses discovered to date. With material from back issues of *VB* extensively updated, and new material written by Edward Wilding, Keith Jackson and Richard Ford, the book provides an instant one-shot authoritative reference on computer viruses.

The book may be purchased directly from *Virus Bulletin* and costs £19.95 (US $29.95). Discounts are available for bulk purchases; distributor enquiries should be made to Victoria Lammer at *Virus Bulletin* (Fax +44 235 559935).

NetWare 4 Security Loophole

According to an alert by the *Computer Incident Advisory Capability* (*CIAC*), there is a security problem in the *NetWare 4* LOGIN procedure which can allow users' accounts to be compromised. *CIAC* claims that no other versions of *NetWare* are affected.

The problem arises because the LOGIN program can temporarily swap a user's account name and password to disk during the login process on DOS machines with a small amount of memory. This could allow the account to be accessed by recovering this information.

A patch is available through *Novell* to fix this problem, and *CIAC* recommends that users replace the current LOGIN.EXE program with the 'fixed' version as soon as is practicable. The patch is also available via the anonymous FTP site, at *first.org*.

This discovery, coming so quickly after the release of *NetWare 4*, will doubtless cause some embarrassment to *Novell*, particularly in view of the great emphasis *Novell* has placed on the enhanced security of this release.

Virus Prevalence Table - August 1993

Virus	Incidents	(%) Reports
Form	23	43.4%
Spanish Telecom	5	9.4%
New Zealand 2	3	5.7%
NoInt	3	5.7%
Parity Boot	3	5.7%
Tequila	3	5.7%
V-Sign	3	5.7%
Vacsina	2	3.8%
Cascade	1	1.9%
Flip	1	1.9%
Jerusalem	1	1.9%
Joshi	1	1.9%
Italian	1	1.9%
Michelangelo	1	1.9%
Starship	1	1.9%
Yankee	1	1.9%
Total	53	100.0%

New Viruses In the Wild

The last four weeks have been bad in terms of the discovery of new viruses in the wild, with three new viruses being reported by users.

The first report concerns the Coffeeshop virus, which is reported to be spreading in South Africa. The Coffeeshop virus uses the Mutation Engine for its encryption, and is the first of the MtE viruses to be found in the wild.

The second report concerns the STB virus. This virus was sent in by a reader in Canada on an infected diskette. The virus, also known as Stealth 2 Boot, is a master boot sector virus, and contains no trigger routine. However, due to a programming error, infected diskettes may sometimes cause the message 'General Failure Error' to be displayed when the disks are used on an uninfected machine.

The last new virus to be reported this month is Satanbug, which is reported to be spreading rapidly in the United States. The virus is highly polymorphic, but contains no destructive trigger routine. Due to an error on the virus author's part, the virus will occasionally corrupt the header of EXE files which it has infected.

In each of these cases there is no cause for alarm, as up-to-date virus scanners should be capable of detecting the viruses. However, the general trend of more viruses appearing in the wild appears to be continuing - and at an ever-increasing rate.

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 27th September 1993. Each entry consists of the virus' name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or preferably a dedicated scanner which contains a user-updatable pattern library.

Type Codes

- **C** Infects COM files
- **E** Infects EXE files
- **D** Infects DOS Boot Sector (logical sector 0 on disk)
- **N** Not memory-resident
- **M** Infects Master Boot Sector (Track 0, Head 0, Sector 1)
- **R** Memory-resident after infection
- **P** Companion virus
- **L** Link virus

_604

CR: A 604 byte virus. Awaiting analysis.

```
_604            3D00 4B74 1280 FC2A 7403 E952 019C 2EFF 1E03 0149 E959 01FA
```

Arcv-companion

PN: A 346 byte 'companion' virus from the (now defunct) ARCV group. Creates hidden COM files corresponding to EXE files. The virus is encrypted, and the following search pattern should be used with care due to its short length.

```
ARCV.346        BF05 01B9 2301 8035 ??47 E2FA C3
```

Australian Parasite.143

CN: Very similar to the 142 byte variant. The virus damages the files which it infects, so disinfection is not possible.

```
Austr.Para.143  B802 3DBA 9E00 CD21 8BD8 BA55 FFB9 8F00 B43F CD21 803E 55FF
```

Beep

CER: A 2000 byte virus. Awaiting analysis.

```
Beep            502D 004B 7476 5850 80EC 4E74 0A58 5080 EC4F 7403 E98B 022E
```

Beer.3490

CR: Similar to the Beer.3164 virus, and detected with the same pattern.

Burger.560.K2

CN: New variants of this old and primitive overwriting virus keep appearing, possibly because they are being patched to avoid detection by known scanners. This variant is very similar to the 560.K version, and is detected with the Burger pattern. The same applies to the 498, 505.A, 505.B, 505.C, 505.D, 505.E, 505.F and 509 variants.

Burma

CEN: A primitive overwriting, 442 byte virus that conatins the text strings '[Tempest - `]' and 'Rangoon, Burma'.

```
Burma           2E01 E8EC 00E9 1501 B801 FABA 4559 CD16 C350 5351 5256 5716
```

Butterfly.Crusades

CN/EN: Two new variants of the Butterfly virus have been found, both 302 bytes like the original, but with the text message changed to 'Hurray the Crusades'. One of the variants infects files with an EXE extension, but as it does not recognize the EXE file format, infected programs will generally crash the machine. The new variants are detected with the Butterfly pattern.

Career

CR: Two variants are known, 446 and 697 bytes long.

```
Career          9C80 FC11 741B 80FC 1274 163D CDAB 7505 9DF8 CA02 003D 004B
```

Cascade.1701.Jojo.D

CR: Like the other variants in the Jojo group, this virus is not encrypted. It is not fully analysed, but does not appear to be significantly different from other related variants. Detected with the Jojo pattern.

Cascade.1701.Yap.B

CR: Internally the virus is virtually identical to the Yap variant, but the decryption code has been modified, presumably to avoid detection.

```
Yap.B           012E F687 2A01 0174 0F8D B74D 01B8 8206 3134 3104 4648 75F8
```

Cascade.1704.K

CR: The decryption loop of this variant has been modified slightly, but it is detected with the Cascade (1) pattern. The same pattern will also detect the 1704.M version, where the only difference is inside the encrypted part. Another new variant, 1704.I is detected with the Cascade-form pattern.

Cha-Cha

CER: A 2391 byte virus. Awaiting analysis.

```
Cha-cha         FB80 FCFF 7504 B834 12CF 5053 5152 5557 561E 062E 803E 4A06
```

Cinderella.C

CR: In this variant the text string has been altered to 'CindyRul.ez', and a few other changes have been made. Detected with the Cinderella pattern.

Dark_Avenger.1800.Ps!ko, Dark_Avenger.2000.Copy.B CER: Minor variants, 1800 and 2000 bytes long respectively, with the text messages at the start changed. Detected with the Dark Avenger pattern.

Datalock.828 **CER:** Detected with the Datalock pattern. This 828 byte virus does not seem to be capable of infecting all COM files correctly.

Flash.688.B **CER:** Awaiting analysis, but seems very similar to the original.

```
Flash.688.B     005E 8BDE 81C3 0F33 C000 FAD5 0A88 07EB 05EA ???? ???? FBC6
```

Hiperion **CR:** A 254 byte virus which does nothing but replicate.

```
Hiperion        9C50 80FC 4B75 1306 5351 561E 5255 33ED E80F 005D 5A1F 5E59
```

Infector.751 **CN:** This variant does not replicate properly, as infected files usually cause program execution to 'freeze'. It is detected with the Infector.726 pattern.

Intruder.1319.C **EN:** Some blocks of code have been moved around in this variant, but functionally it is similar to the other 1319 byte variants.

```
Intruder.1319.C 5F32 C0AA B001 0AC0 C35F 32C0 C3BA 2104 B41A CD21 BFCA 04BE
```

Keypress.1232.C **CER:** A minor variant, detected with the Keypress pattern.

Lockjaw **PN:** This 898 byte companion virus seems to share some parts of the code with the Proto-T group of viruses, perhaps indicating that they have the same author.

```
Lockjaw         9C06 1E50 5352 3D00 4B75 03E8 0E00 5A5B 581F 079D 2EFF 2E82
```

Malaise.1355.B **CER:** Very similar to the original 'A' variant, and detected with the Malaise pattern.

Mannequin.B **CER:** 778 bytes like the original, and detected with the Mannequin pattern.

Mark II **CN:** A 350 byte virus which does nothing but replicate.

```
Mark II         8A57 FC88 5600 8A57 FD88 5601 8A57 FE88 5602 53EB 0790 2A2E
```

Metallica II **CER:** It is not clear if this virus is at all related to the Metallica virus, but it is hard to give it any other name as it contains the text 'Metallica Ver 2.0'. The virus is 1129 bytes long, but has not been fully analysed yet. An 1103 byte variant also exists, and is detected with the same pattern.

```
Metallica II    9C06 5051 5352 1E8A C42C 4B74 13E9 FE02 83C4 18CF EA
```

Moose **CN:** A simple, 353 byte virus which does nothing but replicate. The virus contains the string 'Moose', but also 'MB', which might be the author's initials.

```
Moose           8BD8 33C9 8B84 5E01 8BD0 83EA 02B8 0042 CD21 8D94 0A02 B43F
```

Moose II **EN:** The author of the Moose virus also wrote two other viruses, 468 and 631 bytes long, which only infect EXE files, and are sufficiently different to justify placing them in a separate family.

```
Moose II.468    8BD8 B9FF FFBA FEFF B802 42CD 21BA B802 B43F B902 00CD 218B
Moose II.631    8BD8 B9FF FFBA E2FF B802 42CD 21BA 5B03 B43F B902 00CD 218B
```

Trident.611 **CR:** 611 bytes, not yet analysed. Contains the text strings '[TridenT]' and '{V1.1 Bugfix}'

```
Trident.611     3D00 4B74 1180 FC30 7507 E8DB FFBB 4342 CF2E FF2E C901 5053
```

Trident.90210 **CR:** This virus is 647 bytes long, but is awaiting full analysis. Contains the text strings '[90210 BH]' and 'John Tardy / TridenT'.

```
90210           3DAD DE75 04B8 AAAA CF80 FC11 743E 80FC 1274 3980 FC4E 7437
```

Trivial.Vootie **CN:** A simple, 66 byte overwriting virus.

```
Vootie          B42F CD21 89DE B801 4333 C98D 541E CD21 B802 3DCD 2193
```

VCL New VCL viruses keep appearing. This month brings three encrypted variants by the same author (BEv#A32 - CN, 562 bytes, BEv#A33 - CN, 519 bytes and BEv#A96 - CN, 516 bytes), which are detected by any program which detects the standard VCL encryption method. There are also two non-encrypted viruses, one 386 byte variant, which is detected with the VCL.394 pattern and VoCo (745 bytes, overwriting). The VoCo variant, as well as several other non-encrypted ones may be detected with the following generic string.

```
VCL.generic     B41A 8D56 80CD 21B4 4EB9 2700 5ACD 2172 09E8 0F00 7304 B44F
```

Vector **CR:** 441 bytes. Not yet analysed, but contains the text 'V3.0 [VECTOR] (c) Necros the Hacker Written Aug 1991 in Tralee, Ireland'.

```
Vector          3DF1 4B75 04B8 C0AB CF80 FC11 74C3 80FC 1274 BE80 FC40 7518
```

Willow.2013 **ER:** Somewhat longer than the original Willow virus, but detected with the same pattern.

Yankee_Doodle.Login.3096 **CER:** Very similar to the 3045 byte variant. Detected with the Yankee-Login pattern.

CONFERENCE REPORT

The Third International Virus Bulletin Conference

With the images of the *VB '92* conference still firmly implanted in one's mind, it is difficult to believe that all that Scottish merry-making happened over a year ago. Have 365 days really passed? Apparently so, as the conference went Dutch last month for *VB '93*.

The conference was held in The *Grand Hotel Krasnapolsky*, situated in the heart of Amsterdam. With over 150 delegates making the journey from twenty-four different countries, the conference took on not only a continental but a truly international flavour.

Man cannot live on viruses alone... or so the saying goes. With this in mind, the conference began with dinner for the speakers in the *Five Flies* restaurant, after a canal trip for both the delegates and the speakers, which gave everyone a chance to gain their bearings, and to sample the local brews. This trip was accompanied by a drizzly shower, which (with the Jenever flowing freely) dampened the coats but fortunately not the spirits of the delegates.

Conference Overview

According to many of the delegates at the conference, IT Managers now understand what they need to do in order to prevent virus attack, but want to know how to ensure that their carefully drawn-up policies are actually followed. 'We aren't interested in how Joe User's company places a copy of

Team *VB '93* (left to right): (Back row) Tim Winder, *Shell Nederland Informatiewerverking*, Stefano Toria, *CSI srl*, Jim Bates, *Bates Associates*. (Fourth Row) David Rischmiller, *Oxford University Computer Services*, John Walker, *ADS Computer Systems*, Jan Terpstra, *IBM Nederland NV*, George Guillory, *Paramax Space Systems*, Roger Marshallsay, *Secure Information Systems*, Rupert Goodwins, *PC Magazine*. (Third Row) Righard Zwienenberg, *Computer Security Engineers*, Steve White, *IBM*, Jan Hruska, *Sophos*, Philip Bancroft, *Digital Equipment Corporation*, Vesselin Bontchev, *Virus Test Centre*, Roger Riordan, *CYBEC Pty*. (Second Row) Fridrik Skulason, *Frisk Software*, Richard Ford, *Virus Bulletin*, Dmitry Gryaznov, *Russian Academy of Sciences*, Winn Schwartau, *InterPact Information Security*, Matthias Jänichen, *Virus Test Centre*, Ian Chambers, *ESA*, Rod Parkin, *Midland Bank*. (Front Row) Frans Veldman, *ESaSS BV*.

F-Prot on every workstation', commented one delegate. 'What we want is to understand how to enforce the rules, and what can go wrong.'

The conference attempted to answer some of these problems, but more than anything served to differentiate the needs of the users from those of the anti-virus industry. Exactly as last year, users are increasingly frustrated by the anti-virus manufacturers' schoolboy fascination with competing sizes of virus collections - what they need is a *solution*.

Up to Speed

The delegates had already been treated to the infamous Steve White-Jan Hruska Virus-101 course the evening before the conference began, but *IBM* wanted to reinforce this message. A good virus defence policy is built on several very simple precepts, and the opening talk by Jan Terpstra attempted to drum this maxim home.

However, a far more thorny problem is that of what to do when something has gone wrong. A virus is loose on your computer system. It is not identified by current anti-virus software, and is highly destructive. What should you do now? This is exactly the situation David Rischmiller, from *Oxford University Computer Services*, found himself in.

Summing up the situation in early 1991 at *OUCS*, Rischmiller was disarmingly frank. 'At the start of 1991 we were aware of viruses; we had been subscribers to the *Virus-l* mailing list for some time; we were giving anti-virus advice to our users; and were taking simple anti-virus precautions with the machines under our control which were available for public use. We had even made a start on producing a document about computer viruses and their prevention ... for all that, we were naïve about the issues.'

Rischmiller then went on to explain about the unforeseen problems which the virus (in this case, Spanish Telecom) caused within the university. One interesting side-effect of the problem was that the users became increasingly paranoid about the nature of the virus infection which was spreading throughout the university - a problem which PC support staff will know all too well.

One problem which seems set to affect *OUCS* for the foreseeable future is that of 'haunting' by the Spanish Telecom virus, as machines become infected from one of the many infected floppy diskettes which are mouldering in a forgotten corner of an office. 'If there has been a serious outbreak,' explains Rischmiller, 'everyone is eager to do the right thing, but as the memory fades, so does the enthusiasm. I don't think there is any way of stopping this in a university environment. You can take a horse to water...'

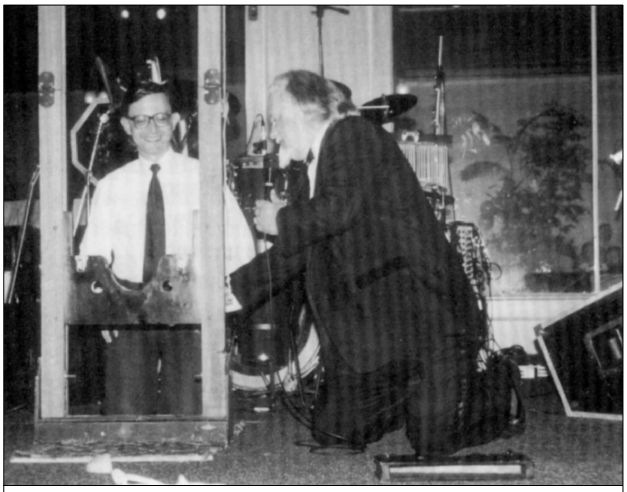
A Sense of Security

Not everyone in the anti-virus industry has the same perspective on how to go about preventing the spread of computer viruses. The most controversial talk of the conference was on an alternative approach to virus prevention.

Winn Schwartau, the cause of the furore, believes that the current approach to virus prevention is simply wrong, and that by using well known security techniques it is possible to limit the spread of computer viruses within an organisation. 'As most security professionals probably already know, I am not a big fan of virus busting', began Schwartau, before embarking on a no-holds-barred critique of the industry.

Schwartau argued that a better way to prevent viruses is to use a combination of the security systems one might find on a large mainframe system. He believes this is a better system for a number of reasons:

- It will cost less money than is currently spent on anti-virus software
- It will save the man-hours spent on keeping anti-virus software up to date
- It will provide protection against unknown viruses as well as known ones
- It will provide a number of additional benefits which are badly needed by the corporate IT manager.



As he was led to the Guillotine, Monsieur Bontchev was heard to mutter 'Let them use DEBUG...'

The delegates and speakers seemed to be divided by Schwartau's assertions. Vesselin Bontchev gave a lengthy multi-point argument against Schwartau (he did not agree with *any* of the points Schwartau raised!) and his views reflected those of several of the speakers and a proportion of the audience. However, the remainder of the delegates were very interested in what Schwartau's model had to offer. The acid test of his ideas will be how they fare on large systems over a period of time - meanwhile the jury is still out on this one. Debate over Schwartau's ideas continued through the rest of the conference.

Reviewing the Reviewers

On a more technical note, Vesselin Bontchev gave an informative account of how virus scanners should be tested. He explained that the biggest problem is maintaining a virus

collection: if the virus test-set used to examine anti-virus software is at fault, the test results are not valid.

However, the process of 'weeding' a large collection of the junk and joke programs which it contains is non-trivial. A typical 'virus collection' may consist of megabytes of data, much of which will not be of interest to the virus researcher - however, it all must be examined, in case it contains new viruses. Bontchev went on to describe how this should be done:

One of the most common mistakes to make when compiling a virus collection is the inclusion of first-generation virus droppers (which Bontchev further classifies as germs, droppers and injectors). The problem with such files is that although they replicate, they do not represent a typical infection, and therefore should not be included when testing scanners.

Bontchev concluded that even after many months of effort, the *Virus Test Centre* in Hamburg was still not ready to review products as thoroughly as he would like.

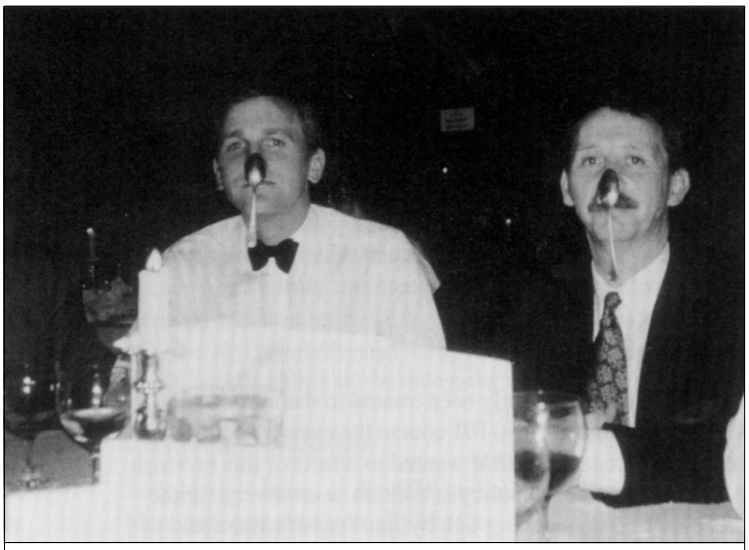
The approach adopted by *PC Magazine* was somewhat less scientific. The *PC Magazine* reviews weighted the usability of the software much more highly, explained Rupert Goodwins. Goodwins' virus detection tests were undoubtedly less rigorous, but gave his readers an idea of the 'feel' of the product. Goodwins then faced a barrage of questions from the more technically oriented members of the audience.

The ideal way to review anti-virus software has yet to be discovered, but such open discussions lead the way to better reviews for us all - the final recipe for the perfect review probably being a mixture of the *VTC's* scientific zeal and *PC Magazine's* 'touch and feel'.

New Virus Trends

Noticeably absent from this year's conference were some of the heavyweight technical papers presented in Edinburgh: hopefully there will be a stronger technical flavour to next year's event. However, the technical presentations were still one of the conference highlights.

One depressingly accurate talk was supplied by Tim Twaits, of *Sophos*. This examined a range of virus construction toolkits which seem to have grown in number overnight. Twaits cautioned that although the toolkits did not present too large a threat at this time, more 'products' were doubtless in the pipeline.



Schwartau recommends using a combination of security measures...
...delegates test his theory after the Gala Dinner.

One increasingly popular technique used for combating viruses is heuristic analysis - a method which has long been surrounded by an aura of black magic. Fortunately, Frans Veldman from *ESaSS* was intent on demystifying the entire heuristic procedure and explained to delegates how his company approached the issue... and unbelievers will be pleased to learn that there were no rams' heads, black candles or Latin incantations involved!

Blue Notes and Red Lights

On a closing note, the conference was not all work. With the venue being so close to the very heart of Amsterdam, there was much sightseeing and merry-making after hours.

The gala dinner proved to be less inflammable but at least as enjoyable as last year. Held in the Winter Garden restaurant at the hotel, the evening comprised a combination of fine food, music and entertainment, by the very capable magicians John and Saxon. The high point of the event was watching Vesselin Bontchev being placed on a working guillotine, although this was rivalled by the sight of *CPAV* product manager Tori Case seemingly floating in the air. The magician was not open to any bribes regarding either of his helpers' personal safety, and both Tori and Vesselin survived the evening unscathed!

After these excitements, the band led the party on until 1.00am - joined for the last few numbers by master saxman Jim Bates and the Editor of *VB*. 'It would never have happened in my day!' *Virus Bulletin's* erstwhile Editor, Edward Wilding, muttered darkly.

Once again, thanks are due to Petra Duffield, who consistently produces perfectly organised conferences, and all her helpers. Several people deserve the *Virus Bulletin* award for dedication well beyond the call of duty: namely Karen Richardson, Victoria Lammer, Rosalyn Rega at *Expotel International Groups* and all the staff at *Crypsys*.

Thanks are also well earned by the speakers, but particularly by all the delegates, whose lively discussions make the *Virus Bulletin* conference the event it is. Where will the conference be next year? Well - watch this space, as great plans are afoot...

VIRUS ANALYSIS 1

PMBS - Intentional Mayhem

Mike Lambert
Before Disaster Strikes

Amidst the usual flurry of viruses which cross my desk, a very unusual sample came to light this month: a Protected Mode Boot Sector virus (PMBS). Anti-virus researchers have been predicting something like this for many months, and to be frank, I am surprised that it has taken so long for a virus which uses protected mode to appear.

The fact that this is the first virus known to use protected mode means that it warrants a longer than usual discussion, and I will attempt to explain the testing sequence I went through to examine this virus, as it marked the beginning of a 'whole new ballgame' for me!

First Impressions

The virus arrived in the usual pile of diskettes and samples sent to me every month in the form of a dropper program (no infected floppy disk was supplied). When this program is run, it installs the virus on a floppy disk, placing its code in the boot sector and 12 other sectors of the disk. These are located in the first available data area of the floppy, and are marked as 'bad' in the FAT to ensure that they are not overwritten later.

The next step was to boot my test machine from this floppy disk, which caused the hard drive of my test machine to become infected. On an infected system, the virus stores a copy of the partition table information in the Master Boot Sector so that the hard drive is still accessible if the machine is booted from an uninfected floppy disk. The contents of the original MBS are stored in sector 13 of the disk, and the remainder of the virus body in sectors 2-12. This left me with a functioning copy of the virus with which to work.

When the system is subsequently booted from the hard drive, the virus code is loaded into a static area of memory, 32k long. This code is later relocated to extended memory. The virus then checks for the presence of a hard disk, infecting it if necessary, and sets up the machine prior to moving the processor into protected mode. There are two different sections of code in the virus: the real mode installation and infection routine, and the 32-bit protected mode code, which acts as the system monitor. The term 'monitor' is used because of the different action of the code in protected mode; all operations are monitored, rather than the virus simply hooking an interrupt vector.

Once the virus is safely installed in protected mode, it has complete control of every aspect of machine functionality, and can monitor the execution of any applications which run in a real mode DOS environment.

Monitoring the Monitor...

The monitor installed in extended memory is extremely simple, and does not seem to be a complete implementation of what is necessary for trouble-free execution - especially when one considers the tremendous compatibility issues raised. It appears to be unfinished (possibly indicative of a test version of the virus?) as it comes complete with its own 'debugging' messages which are presumably included so that the developer can spot exceptions and errors easily. It is these messages, and the compatibility issues, which make this virus so easy to spot. An example of these problems became evident as soon as I tried to boot my machine.

When the virus had first infected my PC, execution of any protected mode utilities, security software, and some conventional application software caused the PC to 'hang' with a mysterious protection fault of one kind or another. All these offending programs had to be disabled in order to get the machine to boot. These problems are caused by deficiencies of the monitor program, and an extensive amount of work would be required to make this a viable virus.

Once the test machine was rebooted with the offending applications disabled, I found that the presence of the virus on disk was stealthed - all attempts to read the MBS returned its original contents. However, there is no stealth protection on the floppy diskette, which seems unusual.

"Once the virus is safely installed in protected mode, it has complete control of the machine"

The virus could not be found in memory using my standard tools, and several utilities would not execute with the virus resident (my greatest concern is not being able to see and control the virus as I normally can with the standard real mode viruses - a new high priority is the construction of some new tools). A debugger or any other software which attempts either to control or to examine the whole machine causes the system to hang with a 'protection' violation message. Current AV products will almost certainly produce some conditions which the virus monitorfinds objectionable and therefore make it show its hand.

Protected Mode Behaviour

The protected mode monitor evaluates a number of interrupts and I/O instructions, one of which is floppy disk access (used to trigger the infection process), and another of which provides boot sector stealth. Token support is given to protected mode issues, mostly to ensure stability rather than to hide stealthily from inquisitive eyes.

The virus monitors Int 13h, read function 2 for a floppy disk, and when the virus finds a disk to infect, the virus moves its real mode code down for execution in real mode. The monitor takes note, transferring to the real mode code to accomplish the potential host evaluation, and when necessary, the infection. When this procedure is finished, it 'signals' the monitor by executing a monitor trapped interrupt (Int FFh). The monitor then cleans up and returns control to the original requester.

Hard disk accesses are screened for any MBS reads and are 'redirected' by telling the hardware where the MBS really is. Protected mode interrupt access is simply denied and extended memory portrayed as non-existent.

Propagation

The DIR, COPY commands, and some other floppy disk accesses, sometimes fail to coax the virus to propagate, but it is capable of infecting a floppy disk. Trying the virus on two different 386s, it was difficult to infect floppy disks on one, but the other infected quite easily. This is probably a result of the various ways that the different BIOSes spin up the floppy disk (the 'motor on' bit is checked in the infection routine).

I have used the term 'infect' rather loosely here: I mean that the virus writes itself to the floppy disk but is not a fully functioning virus. The resulting floppy *does* write code to other hard disks but not with a copy of the virus. On one test machine, a first generation floppy caused the system to reboot instead of the expected non-bootable disk message - I later found a section of the DOS kernel in the virus instead of the system monitor code. On another test machine, booting from the first generation floppy simply hung the system. So, there is no propagation in the normal sense of the word - this may be a compatibility issue, as it seems unlikely that this was not tested by the author.

Disinfection and Protection

Protected mode interrupt support is less than basic. Most calls are blocked by a few short stubs of code and it appears that everything else is passed back to real mode. The virus appears to contain no overtly malicious code. The only thing of any note is that the virus *does* go to some effort to find and protect its home on the floppy (the code is just over 500 bytes). This code could have been extensively improved upon, both in terms of size and implementation, so it seems that neither of these issues were of concern in the design.

Disinfection is the standard 'boot your clean disaster recovery disk and restore the MBS'. The ID the virus uses to indicate infection is the PMBS portion of the PMBSVIR text in the OEM ID area of the boot sector. Since the same code section is shared for hard disk and floppy infection, the hard disk MBS and first generation floppy boot sectors contain a partition table. This means the same string is in the same place in the MBS. In either case, the presence of a MBS lacking its error messages and the floppy boot sector missing all its messages make the virus easy to spot.

Pertinent Questions

As the concept of a protected mode virus is very new, I will attempt to anticipate some of the more obvious questions and tackle them here:

Is this a virus which will be found in the wild?

No; it was obviously never intended to be an 'in the wild' virus, at least in its current form. I did not even produce a viable first generation floppy infection during testing, so PMBS is on the edge of qualifying as a virus. My personal definition of a 'minimum' virus requires the code '...logically or physically to propagate without permission...' so it may not technically be a virus. Unfortunately, in the real world, the problems can be fixed, and this code can be made to propagate. In addition, it proves that a protected mode virus is possible, and the eventuality of more viruses of this type being developed needs to be addressed now.

What exactly is the purpose of PMBS?

PMBS looks like either:

1. Work in progress.

2. A demonstration virus. I say this because of the mixture of considerable expertise combined with the 'neglect to do a complete job'. Surely if someone is good enough to design and program a protected mode virus, they must also be capable of at least basic compatibility. Announcing such things as 'General Protection Fault', 'Unimplemented Interrupt', 'Offending Instruction', and then hanging the system is a long way from trying to hide from anybody! If you are trying to hide from the curious, why announce yourself by using the OEM field to store PMBSVIR?

While this version is not going anywhere (infecting people's computers without their knowledge), it *is* a persuasive demonstration that 'it' (a protected mode virus) can be done.

Author's note:

I have subsequently found that the subject of protected mode viruses was discussed in *Computer Virus Development Quarterly*, Vol 1 Number 4, published by American Eagle Press, PO Box 41401, Tucson, AZ 85717. The 'virus' I have analysed above is the subject of an article on the perils faced by those who use a protection system which cannot deal with the protected mode virus possibility. Protected mode and the virus implementation are fully explained in the article. This certainly clarifies the glaring discrepancies noted in the technology mix employed in the design and code. It appears that only the minimum 'virus support' was used around the protected mode theme. The author suggests hardware write-protection as the best defence and suggests that those not employing hardware write-protection should at least check the system immediately after it is booted to determine whether it is in protected mode when it should not be.

This is not the first virus written just to show that a certain technique is possible, and that may well be the reason for its existence. It is easy to make a case that this virus uses a more advanced form of stealth. We did not always have polymorphism, hardware stealth, and the myriad of current memory stealth techniques; someone did it first. In some cases it was done to show the anti-virus software developers where they were going to have to go in the future.

"protection needs to be purchased in one form or another, and the 'free lunch' of a TSR trying to compensate for the bad habits of the user community is gone"

This looks like the next logical step in the Virus technology vs Anti-Virus technology dance we see everyday (as any targeted anti-virus developer will tell you).

What if someone fixes this thing?

If the object is to produce a compatible, protected mode virus capable enough to spread in the wild, it might be better to redesign the virus. Remember the word 'monitor' and protected mode. There are in fact 'DOS Extenders' [*For example, Windows. Ed.*] which are protected mode monitors, and anyone developing such a thing knows full well how much work it is to code, test, and release compatible protected mode code! This is not a job for just any programmer, and if we take into consideration the necessary size of the resulting code, it will be a little difficult to hide (one person quipped that the virus would need to display the message 'Insert Protected Mode Virus Disk 2 and press return' in order to function invisibly!).

Is there any protection from these things?

Due to the nature of protected mode programs, solving this problem using 'vanilla' DOS will prove to be extremely difficult. However, there are solutions which will protect the system. Hardware write-protection of the boot sector and system files provide protection from all viruses incapable of removing cards or changing jumper settings. It is possible never to boot the system using the MBS - in this case, viruses can insert code without ever becoming active on the machine. The bottom line here is that protection needs to be purchased in one form or another, and the 'free lunch' of a TSR trying to compensate for the bad habits of the user community is gone (if it ever really existed).

Future Developments

It seems highly unlikely that there will be no more protected mode viruses, and development will probably parallel the lines of virus encryption and memory stealth - where one has gone, many will follow. Whatever the next step is in protected mode viruses, three things are guaranteed.

- The expertise of the author must be much greater than that typical of current virus authors.
- Protected mode viruses will be bigger, and harder to hide.
- Protected mode viruses will have compatibility problems and may be easier to find (in the early development stages) by accident than current real mode stealth viruses.

I do not think this is anything to cause immediate concern, unless you are sitting too far behind the development curve. The prudent protection provider, be he software developer or security consultant, will see what is coming and spend the time available planning for the inevitable. Fortunately, we do seem to have some time.

Panic Now?

In its present incarnation, a scan string is not needed to find this virus: if a computer can still execute the usual DOS extensions (eg memory-managers) and the system operates normally, it is not infected. In addition to this, because the virus does not replicate correctly, it will not spread.

The virus is interesting in that it is a protected mode virus, but other than this, it uses no new technology. No destructive trigger routines are included, and it is unlikely that it will cause extensive damage either to floppy or to hard disks. This is the place neither for moral evaluation of such creations nor a review of the quality of coding: I do not feel qualified to do one or the other. The reader is left to make up his own mind on the trends and whims which drive the virus world along.

PMBS

Aliases:	None known
Type:	Protected Mode, Master Boot Sector
Self-Recognition on Disk:	
	OEM Name set to 'PMBSVIR'
Self-Recognition in Memory:	
	None necessary
Hex Pattern:	
	`E80F EEE8 82F0 E801` `F1FA E883 F1E8 67F1`
Intercepts:	In the usual sense of the word, none. However, the monitor uses Int FFh internally, and contains code to intercept Int 13h, subfunction 2 and Int 15h, subfunctions 87h and 88h.
Trigger:	None, but causes extensive disruption of many applications.
Removal:	Under clean system conditions, replace contents of Master Boot Sector using FDISK /MBR.

VIRUS ANALYSIS 2

Sibel Sheep: Crying Wolf?

Jim Bates

It has always been my suspicion that virus writers are mentally abnormal. This is confirmed by the contents of a recent virus which suggests that the writer's mind is not simply twisted, but actually sprained!

The virus has been in circulation for several months, but has recently been reported at large in the UK and it has therefore been necessary to disassemble and analyse it in the usual manner. For reasons which will become apparent, the virus has been called Sheep (or Sibel Sheep) and while it represents no more than the usual nuisance value, it is interesting because it exemplifies all the classic facets of most viruses which cross my desk.

Our star virus writer this month appears obsessed with sheep and also attempts to grab our interest with what seems to be some nonsensical reference to cars. Those readers with nothing better to do might attempt to decipher this gibberish but if you succeed, please don't tell me - my own cerebral processor has been sorely overloaded for some months now!

Overview

The Sibel Sheep virus is a parasitic, resident, COM/EXE infector which deliberately corrupts DOC, TXT, ARJ, and BAK files on a pseudo-random basis. The code is encrypted by a laughable attempt at polymorphism. The actual virus length is 2352 bytes although this is increased by a random amount during infection. COMMAND.COM is infected, but this will undoubtedly cause system malfunction.

The DOS Interrupt Service Routine (Int 21h) is intercepted, but apart from servicing the ubiquitous 'Are you there?' call, only function 4Eh (FCB Find First) is subverted.

Initial Operation

When the virus first executes, it decrypts the virus body. It shuffles some register contents, and issues an 'Are you there?' call. This involves placing a value of D4h into the AH register and issuing an Int 21h request. If the virus is resident, the D4h value is incremented before the request is returned and processing jumps to the exit routine. Otherwise, processing passes on to the pre-installation routine.

This is the first error in the virus: the 'Are you there?' call interception will cause serious malfunction on *Novell* and *Banyan VINES* networks, since they use a similar call for access control of their logical records. This type of conflict exemplifies the poor level of technical competence displayed by the virus author - and I hope that if he reads this he feels suitably sheepish.

The pre-installation routine examines the machine environment for the name of the file specified in the COMSPEC variable (if not otherwise set, this will be the usual command interpreter, COMMAND.COM). This field is checked to see if it matches the pattern 'c??cO???.??' and if it does, the virus issues an Int 05h call before jumping to the exit routine. I am not aware of a COMSPEC variable which matches this pattern, particularly considering the lower case letters involved. This may be an attempt to avoid or disable a protection mechanism, or avoid infection of the virus author's own machine. At this point, if the COMSPEC file is in the default directory, it will be infected by the virus.

"the 'Are you there?' call interception will cause serious malfunction on Novell and Banyan VINES networks"

The virus then herds the existing Int 21h vector into its data area and checks the DOS version number. If this is earlier than version 4, the code is relocated to a point 128k below the top of available memory. Processing then continues by making a further test on the COMSPEC variable (in a similar fashion to before) for a pattern of '??NDOS?.???'. In this case, if a match is found, the virus is relocated in memory to segment 8AA0:0100h (instead of 9000:0100h).

Whether the virus code is relocated or not, no attempt is made to protect it from being overwritten by subsequent system activity.

Once relocated, the virus first hooks itself into the Int 21h service routine and then collects a value from the system clock hundredths of a second field. If this value is zero, the computer will hang. Otherwise, it goes on to check the system date. If this is 7th May (any year), the following message is displayed on the screen and the computer hangs.

```
KIRYAT MOSKIN!!!
LOCAL PROCESS INDUSTRY.
VIRUS DONE BY:
SIBEL ,TEACHES
 HOW TO MANAGE SHEEP?
Thanks for using Turbo Anti Virus.
PLEASE JMP FE00:0
```

On any other date the virus exits to the host program. The 'KIRYAT MOSKIN!!!' message here may be a greeting (or an insult) in a foreign language, a magic spell to ward off evil spirits or even the name of the writer's favourite sheep (whose birthday just happens to be on May 7th). Whatever it means is not really of the slightest interest except for identification purposes.

Resident Operation

Once resident, this virus intercepts and subverts all requests for Int 21h subfunction 4Eh (FCB Find First). After extracting the target filename, the virus checks to see if the request is directed at drive C and if so, it checks for the existence of a directory named 'the Great'. If this is found, infection is terminated and processing is returned to the system. In all other cases, the virus continues by saving the caller's filename and issuing its own search for any available file. Once found, this file is checked and treated accordingly.

If the file found has the extension BAK, the following text is inserted at the beginning of the file:

```
... What is backup for anyway??? BackUp is
usually unnecessary ! End..
```

If the target file has an ARJ extension, a corrupting jump instruction is inserted at the beginning of the file. For files which have the extension DOC or TXT, the following message is inserted at a point halfway along the file:

```
'What's 455260 MI COUNTACH 5000 CC???
Instead of reading this junk, think about it!'
```

Once again I have made no attempt to unravel this gibberish.

If the file has a COM or EXE extension, it is passed to the infection routine. The virus maintains a counter to try and infect two files during each interception.

"If the target file has an ARJ extension, a corrupting jump instruction is inserted at the beginning of the file."

Infection Routine

This routine processes both EXE and COM files and detects the difference by the usual expedient of checking for the 'MZ' header which identifies EXE file structures. No check is made of absolute file size, so COM files greater than approximately 63k will be irreparably damaged. An abortive attempt is made to check whether there are any resource areas attached to EXE files, but this code is so riddled with mistakes that there will certainly be damage to such files if they become infected.

Infection is achieved by appending the virus code to the host file and modifying the file header to ensure that it gains immediate control. In an apparent attempt to avoid some of the simpler anti-virus controls, the virus renames a potential target file to the extension VZQ before infecting it and then renames it again.

Once successfully infected and renamed, the target file seconds field within the Date/Time stamp is changed to the value 13h, which represents 38 seconds. It may be worth noting that corrupted BAK, ARJ, DOC and TXT files also have this seconds value set.

Just prior to writing the virus code to the target file, an encryption toggle algorithm is generated to make the virus polymorphic. This particular virus writer was obviously too preoccupied with other things to give this much thought, since there are just 2 algorithms with 8 variations.

Even including the garbage code, generated on a pseudo-random basis, the code derives a grand total of only 16,384 possible variations. Compare this with something around 3×10^{18} for the Mutation Engine and you will appreciate the skill of our sheep molester.

Conclusions

There are many conclusions which can be drawn from examining this virus, most of which the Editor would not print. However, it is noteworthy that most of the code seems to be original. I did not recognise any obvious similarity between sections of this code and other viruses which I have analysed. The mutilation of the various file formats attacked is nothing more than 'computer vandalism'. In addition to the deliberate damage caused, the bugs which the virus contains can cause serious problems.

To all present and prospective virus writers let me plead - don't waste your time, viruses are a dead end and it is much more fun to write productive programs.

To the author of this virus I can only suggest - 'return to your sheep, she's probably missing ewe!' [*Groan. Ed.*]

SIBEL SHEEP

Aliases:	SHEEP
Type:	Parasitic file infector
Infection:	COM and EXE files (including COMMAND.COM)
Self-Recognition in File:	Seconds field set to 38
Self-Recognition in Memory:	Issue INT 21h call with D4h in AH, returns D5h in AH
Hex Pattern:	(On disk and in memory) 9C80 FCD4 7504 FEC4 9DCF 80FC 4E74 03E9 7701 5053 5152 5657
Intercepts:	Int 21h Function 4Eh for infection.
Trigger:	Random action - corrupts ARJ, BAK, DOC and TXT files
Removal:	Under clean system conditions identify and replace infected files.

FEATURE

Computer Viruses in the Corporate Arena

Micki Krause
Rockwell International

Computer viruses have been of increasing concern at *Rockwell International*, with hundreds of incidents reported over the past five years. Most recently, two large business units suffered infections on sizeable local area networks, rendering computer resources unavailable, and hindering business operation. Subsequently, the virus problem has escalated to present a serious business risk.

Moreover, the nature and implied intent of computer viruses have significantly evolved in recent years. The seeming innocence of the Cookie Monster virus has been overshadowed by the stealth-like, self-encrypting viruses of today. This transformation, and the profound impact it has on information processing, have changed forever the way we plan, design, implement and manage the distributed computing environment.

Being Prepared

In late 1988, I participated in one of the earliest computer virus symposiums, sponsored by *Deloitte & Touche* in New York. *Rockwell International* had a vested interest in my attendance and participation. Although the majority of symposium attendees had only read about viruses, we had already experienced virus attacks on Macintosh computers.

At that early meeting, security professionals were hard pressed to agree on a common definition for a computer virus. In fact, for two years after that, debates ensued over whether or not viruses were a fad that would become passé, or a real threat to be taken seriously.

Five-Year Tracking Record

Although actual virus incidents were reported in 1988, computer viruses were not considered a serious business concern at the time. Many people thought they were a joke - a novelty - something that could not affect a real computer. Viruses were an enigma. Their actions were a mystery; their origin was a puzzle; no real damage could be attributed to them. Awareness of security and prophylactic software for viruses was impossible to sell at this time.

Throughout 1989, we saw increased infections, still Macintosh-related, and still perceived as a mere nuisance. Attempts at educating and informing users met with resistance and/or denial. Many computer-literate folk looked us squarely in the eye as they said 'Computer virus - there is no such thing.'

Viruses Within Rockwell

In 1990, primarily due to increased interconnecting of computers, we experienced an explosion of virus incidents. Macintosh viruses which had existed on disk and stand-alone systems were now being propagated through Macintosh networks. PC viruses, such as Stoned and Jerusalem, made their corporate debut. Over 600 incidents were reported throughout Rockwell in 1990. Subsequently, we made our first major investment in anti-virus software.

During late 1990 and throughout 1991, we marketed the installation and use of anti-virus software throughout the corporation. The software we purchased as our corporate standard came as a suite of programs which included scanning and cleaning executables, and Terminate and Stay Resident programs (TSRs) for activity and anomaly checking. Since many of our installations are local area network based, with an already over-encumbered TSR environment, we chose to implement the scanner executable, run from AUTOEXEC.BAT, with a configuration file which included a date parameter.

The scanner was thus kicked off only at the initial boot up every day, regardless of how many times the machine was booted during the day. On networked PCs, the scan was performed prior to network connection.

> *"As with the preceding Monkey virus, this virus was new and our anti-virus software did not recognise its signature"*

We reinforced our anti-virus campaign with comprehensive management briefings, virus alerts and security newsletters highlighting the new and recurring viruses. We increased the internal availability of anti-virus software by storing it on multiple platforms to enable a broader distribution. Our software licensing agreement allowed it to be used at home by employees, so that disks used both within the company and at home were not a potential source of infection.

As anti-virus software was installed throughout the company, viruses were discovered lying dormant on PCs. Security awareness was heightened, and by mid-1992, we began to see a decrease in virus incidents. We attributed our success to a strong awareness campaign and a marked increase in the use of anti-virus software.

By the end of 1992, virus incidents had decreased substantially and we thought we had finally had our arms around the problem. Unfortunately, we were lulled into what we now know was a false sense of security.

Complex Viruses Discovered

In late 1992, information was disseminated about more sophisticated, more dangerous viruses - viruses which change system attributes and evade anti-virus software; stealth, polymorphic and crypto-engine viruses; and increasingly, viruses which originated in Western Europe, Eastern Europe and beyond. We were coming to realize that the products upon which we were now dependent, whose strengths lay in scanning for known virus signatures, could soon become obsolete. In a worryingly short time, those products began to fail us.

In early 1993, the Monkey virus was discovered in a southern California business unit. Several PCs were infected, and our standard anti-virus software product had not detected it. Although the impact of the incident was not quantified in terms of lost data or system downtime, it got the attention of our user community. They began to demand a better anti-virus solution.

Scanner Exhaustion

Two months later, in April 1993, we were blind-sided once again, this time with the Hi virus. As with the preceding Monkey virus, this was new and our anti-virus software did not recognise it.

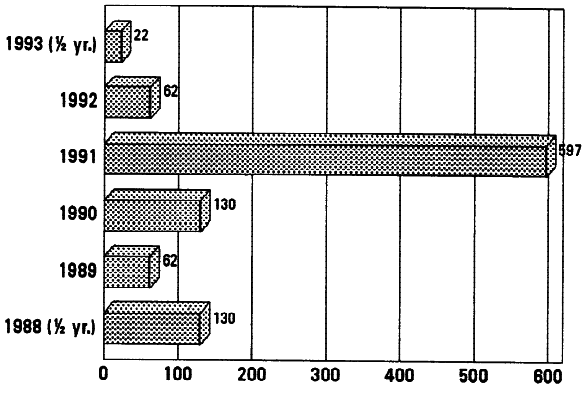
The Hi virus infects memory and executables. It does not carry malicious instructions to delete data or destroy disks. Regardless, it took its toll. The Hi struck a very large business unit located in the US. This division is heavily networked, with 9 file servers and 630 PCs in one location, and connections to 30 other US sites and 34 European sites.

This particular division had had its share of viruses in the past; thus, the PC/network support personnel were very virus-aware. Anti-virus software had been installed on all of the networked PCs. Floppy disks were scanned prior to using them. By all accounts, this division had taken all of the right steps to protect itself against viruses.

After some investigation, it was discovered that the Hi virus had arrived on program disks received from a legitimate business partner in Switzerland. According to division personnel, the disks were scanned according to procedure prior to being loaded onto a production network. A day later, systems began to go awry. Using an auxiliary anti-virus product, technicians found the Hi virus on file servers, floppy disks and multiple PCs. Despite efforts to contain and eradicate the virus, it continued to travel around the network throughout the entire month of May.

Financial Impact of the Hi Virus

At my urging, and in order to justify the cost of additional anti-virus software, division management quantified the cost of the virus. The following is an approximate cost breakdown, according to the Manager of Information Systems (Dates of infection: April 29,1993 - June 2, 1993).



Breakdown of virus incidents reported by *Rockwell International* for the period 1988 to 1993.

(1) *Rockwell* internal PC and LAN support technicians spent approximately 160 hours, at $45.00 per hour, to identify the virus infections, consult the users, scan with anti-virus software, and delete and restore the infected files. A week and a half passed before our vendor provided us with a recognition string of the Hi virus. Having received it, we were able to clean the infected files, a process requiring less time than deleting and restoring files.

1. $7,200 *Rockwell* PC/network support (160 hrs @ $45/hr)

(2) External contractor support was hired for 200 hours at $40.00 per hour to work with *Rockwell* employees.

2. $8,000 Contractor support (200 hrs @ $40/hr)

(3) One file server was disconnected from the network, to prevent the virus from spreading through the LAN. This server was unavailable for an entire day while the origin and spread of the virus was being determined, files were cleaned or restored, and other servers were scanned for a sign of the infection.

Approximately 100 employees relied on the one server for the resources required to perform their regular job duties.

$36,000 server downtime (100 users @ $45/hr - 8 hours)

On the average, the users accessed the network for about 25% of the normal work day.

3. $9,000 $36,000 x 25%

(4) Management assessed the costs of purchasing additional anti-virus software. Approximate cost of software for each file server = $900; approximate cost of each individual workstation = $20.00.

4. $19,800 Additional AV software required.

The grand total cost of the incident was $44,000.

For the sake of comparison, if we multiply the $44,000 by ten incidents (a reasonable assumption, considering the number of divisions a large company may have), that number explodes to $440,000 dollars, which equates to a major bank robbery or a fraudulent electronic funds transfer. This is the magnitude of the cost which we are facing.

Unfortunately, many executives find it difficult to relate unavailability of resources to a bottom line dollar cost. And yet, the biggest impact from computer viruses has been and continues to be the unavailability of resources. In defining information security, availability of resources is included as an integral component. Thus the unavailability of our computer resources reduces computer and data security - leading to a direct dollar cost.

Although we have had instances where files were lost, the overwhelming impact of viruses has been the resulting unavailability of computers and resident data. Especially when a virus is propagated through a network, multiple users are put out of work and administrators and support staff are forced to stop performing regular job duties to work on the problem.

The proper response to a virus incident calls for the system(s) in question to be isolated. In many cases, virus outbreaks have affected multiple file servers, making them inoperable for unacceptable periods of time.

Many professionals believe strongly that virus legislation should be enacted and that the punishment should fit the crime. Someonce once referred to a virus as a tax that we pay on the cost of using a computer. I submit that the burden is heavy, and becoming heavier.

Actions and Recommendations

My briefing to the US Congress was prepared to lend support for anti-virus legislation; thus one of my recommendations for solutions to the computer virus epidemic is to enact laws which would penalize the virus writer. However, I believe that legislation is our last resort, to be used when all else fails. Therefore, I submit the following action items for consideration:

• Improved Quality Assurance and Control. Commercial hardware and software vendors must adopt more stringent methods to assure that the systems are not contaminated prior to shipment. Too often, we discover viruses in commercial shrink-wrapped software or in systems which we receive on a turnkey basis from hardware vendors. Additionally, hardware/software service units must upgrade the quality of their diagnostic tools to ensure that diskettes carried by service technicians from customer to customer are not infected.

• Integrity and security should be built in to application software and operating system software. Depending on the sensitivity of the system/data, we find it necessary to use additional security and assurance products because we cannot rely on the integrity of the core system. It has become inefficient, ineffective and very expensive to layer security products; the burden on the user, the computing resources, and the company is becoming unbearable.

• We need an independent, unbiased source of product evaluations. Companies are being bombarded with security and anti-virus products of all shapes and sizes. Not only are we unable to test these products on all computing platforms used within our companies, but it is impossible for customers to test products against the thousands of existing strains of computer viruses. We need a sense of assurance that the product will perform as advertised.

" Despite efforts to contain and eradicate the Hi virus, it continued to travel around the network throughout the entire month of May."

• A centralised resource for all incident tracking, education, and security alerts. Security professionals need a repository for reliable information concerning virus incidents and information to be able to educate users and the community at large. Although many companies are reluctant to admit that viruses have invaded their computing resources, surveys and industry studies show that a majority of companies *have* been infected. As opposed to being an embarrassment to the company, I believe that the willingness to share our experiences indicates a sense of community and industry responsibility. I hope this will encourage others to do the same.

In Summary

The nature and implied content of computer viruses have evolved significantly in recent years. This transformation has had a profound impact on information processing and has changed forever the way we plan, design, implement, and manage the distributed computing environment.

Viruses have become a serious threat to computing. The incidents to date have been costly, primarily due to the extent that our businesses have been disrupted and the unavailibility of our resources. The migration of major applications from the traditionally protected and secure mainframe environment to the inherently insecure PC and local area network causes serious exposure.

The risk is ever-increasing, thanks to the virus authors whose creations are continuing to become more sophisticated and more dangerous. The risk to a company such as *Rockwell* is even greater than to some other companies, because of our extensive domestic and foreign networking, closer to the origins of some of the newest and meanest viruses. We hope that briefings and articles such as this will perpetuate cooperative efforts and bring us closer to real world solutions.

PRODUCT REVIEW

A Clean Sweep

Keith Jackson

Sweep for DOS is an anti-virus program which can scan hard drives, floppy disks and networks for the presence of viruses. It currently claims to be capable of detecting 2834 viruses, a rapidly rising total. *Sweep* is updated every month to keep up with this remorseless increase in the number of known viruses. No checksumming facilities are included with *Sweep*; these are provided by other *Sophos* products.

Documentation

Sweep comes in a 'standard' size slipcase which contains a copy of a book entitled 'Data Security Reference Guide', a user manual, permanently write-protected 3.5 inch and 5.25 inch floppy disks, numerous bumph sheets for other *Sophos* products - and an advert for *Virus Bulletin*! One nice touch is that the package contains two sheets of pre-printed sticky labels which can be used to mark disks as being virus-infected, or to indicate when a disk was last scanned.

Sweep is available for *OS/2*, *Novell Netware*, and *OpenVMS* as well as *MS-DOS*, though this review will look only at the *MS-DOS* version (unless somebody cares to donate a *VAX* to me, in which case I will duly extend my testing in return). The point of having the software available on other operating systems seems to be that they are often used as file servers for networked PCs, and *Sweep* can perform anti-virus checks directly on a file server.

The *Sweep* user manual is a well-written 100 page A5 wire-bound manual which is thoroughly indexed and which contains a voluminous fifteen page Glossary of technical terms. It is seemingly free of the marketing rubbish affecting so many anti-virus product user manuals.

A quick start tutorial is provided, as well as a thorough description of using *Sweep* either as a stand-alone product (command line driven), or via the 'interactive shell', turning *Sweep* into a mouse, keyboard, and drop-down menu type product which can work under either DOS or *Windows*. Personally, I prefer the former method of operation, but rodent-addicted users will doubtless opt for the latter.

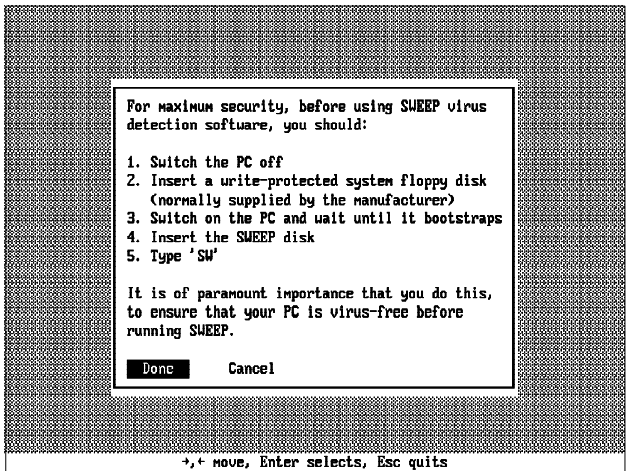
The user manual contains chapters on what do when problems prevent correct operation, and what to do if a virus is detected. My only real beef with the manual is that several terms defined in the Glossary have nothing at all to do with the matter in hand. For instance, do access security models such as the Bell-LaPadula model and the Biba model really matter to someone scanning for viruses on a PC running MS-DOS? I think not. In similar vein, terms such as Virus Description Language (VDL) are used in the User manual without being explained in the Glossary.

The developers of *Sweep* do not believe that 'cleaning' a virus from an infected file is a good idea, and to this end the advice in the *Sweep* manual is always to replace an infected file with a copy which is known to be virus-free. This is sound advice, but it may prove onerous on networks where the same infected file is present on PCs situated at geographically remote sites which have to updated locally rather than via the network. Even so, I agree with this stance. Virus infected files should always be eradicated, rather than merely tinkered with, but I am fully aware that this is a very personal viewpoint.

Installation

Installation of *Sweep* to hard disk, as far as DOS is concerned, is very straightforward: a few files are copied, and there is really nothing more to be said about it. *Sweep* can be executed directly from floppy disk; the manual explains in detail how to perform a boot from a 'known clean' floppy disk, and how to execute *Sweep* from floppy disk. As the manual quite clearly states, such an approach provides maximum security, but it is impossible if the files do not fit onto a floppy disk - such an Herculean task is not possible for scanners which require *Windows* to be present.

A *Windows* installation program is also provided with *Sweep*; this caused me a few problems. For starters, the installation program is incredibly slow. It begins with a huge hard disk thrash (the function of which is mysteriously unexplained); then, after asking for the *Sweep* disk itself, the installation program took 2 minutes 44 seconds to copy four files (580 Kbytes in total) to the hard disk... and this was on a 25 MHz 486! To give some idea of how poor this performance is, the DOS COPY command can copy the same files from floppy disk to hard disk in just 33 seconds, which is almost 5 times as fast.



Better advice would be hard to come by - but how many users will actually follow the on-screen instructions?

The *Windows* installation program changes the date/time of all installed files so that the date is set to 1/1/80, and the time is set to 0:00. As this alteration prevents users from seeing at a glance the age of the installed version of *Sweep*, it is potentially quite damaging and should be fixed as soon as possible. I fail to understand why the *Windows* installation disk contains 16 files, seven of which seem to be *Windows* DLL files, yet (not counting the files contained on the DOS *Sweep* disk) the *Windows* installation program merely installs an icon file and a *Windows* PIF file. What do all the other files do? Users should be told.

As for explaining the *Windows* installation process, the manual is no help whatsoever - it merely says that 'The installation program will display the screen, telling you which files are being copied, as well as other pertinent information'. A tad terse perhaps?

Do not be misled into thinking that *Sweep* comes in two flavours, a DOS version and *Windows* version. Only one version of the scanning program is provided: this may be executed as a command line driven program, or the interactive shell may be used. The latter can be executed in a DOS box under *Windows*. Although I personally prefer using scanners directly from the DOS command line, I fully understand why many users wish to have menus/mouse/keyboard driven option selection. As long as such a program does not become so bloated that it will not readily fit onto floppy disk, this is not deleterious.

On the Menu

The interactive shell provided with *Sweep* provides the usual plethora of drop-down menus, and works very effectively. I found no real problems with it, beyond a desire for the mouse to be able to click on the explanatory help provided on the bottom line of the screen, and a lack of short-cut keys to provide a quick path through the various menu options. I particularly like the on-line virus information (available as long as the file SW.DAT is present), which could save much digging around for long-lost paper manuals when a virus is detected and identified.

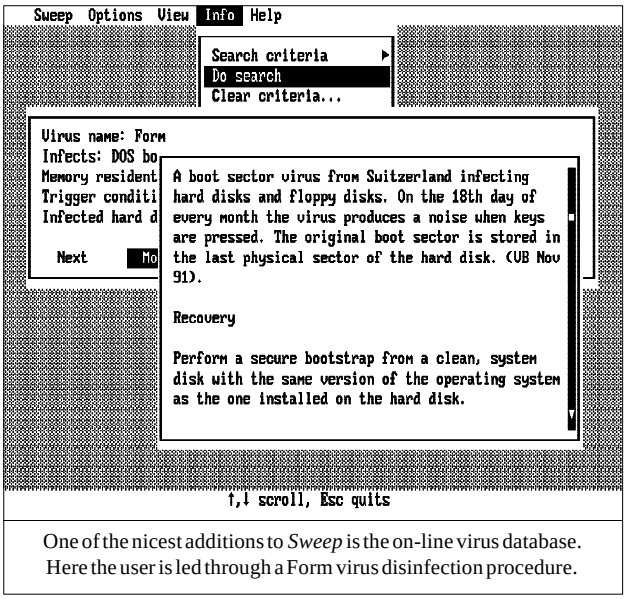
A word about disks is in order at this point. *Sweep* was provided on two 3.5 inch disks (720 Kbyte), but only one 5.25 inch disk (360 Kbyte). Of the two 3.5 inch disks, one disk contains all of the *Sweep* files, and one contains the *Windows* installation files. The 5.25 inch disk only contains some of the *Sweep* files, in particular the file SW.DAT which provides the on-line virus information is only supplied on 3.5 inch disks. I am afraid that I cannot understand this at all. Do the developers of *Sweep* believe that users who have 5.25 inch disk drives only want to perform a basic disk scan, and do not want access to all of *Sweep's* features?

If I purchase a package which does not have a huge pile of disks, I have now become conditioned to it containing both types of floppy disk, and for all of the features to be available on either set of disks. Skimping on this point for the price of a couple of floppy disks is not helping anybody.

Sell by Date

When *Sweep* is executed, it provides an on-screen warning if the software is more than 4 months out of date. This warning still works although the *Windows* installation of *Sweep* has changed all the file dates to 01/01/80 (see above). The normal frequency of update is every month, so barring a wrongly set clock, most users will never see this warning message. Note that users are not prevented from using an out-of-date version, they are merely warned that its 'shelf life' has expired.

Sweep can check any chosen part of any designated disk. The user can specify that *Sweep's* attentions should be concentrated either upon a whole disk, individual file(s), logical disk sectors, absolute disk sectors, the boot sector or even a memory range. Most users will just ask for a whole disk (or several disks) to be scanned, but the extra specification features could prove invaluable if a virus is detected, and more than a routine disk scan is then required.



One of the nicest additions to *Sweep* is the on-line virus database. Here the user is led through a Form virus disinfection procedure.

The speed with which a scanner operates is always difficult to describe in terms that can be usefully comprehended. Knowing the time taken to scan my hard disk is of no use whatsoever to anyone else as far as their own system is concerned. The only meaningful test is to compare a scanner against other well-known scanners, and measure their relative performance.

Sweep scanned my hard disk, containing 758 files (23 Mbytes) spread across 28 subdirectories, in 1 minute 51 seconds. For comparison purposes, *Dr. Solomon's Anti-Virus Toolkit for DOS* (*AVTK*) needed 25 seconds, and *McAfee's SCAN* program needed 1 minute 43 seconds to scan the same hard disk. *Sweep* also has a quick mode of scanning ('full' scanning, the default mode, examines every byte of each file), which required 40 seconds to scan the same hard disk. When the same tests were performed under *Windows*, *Sweep* required 2 minutes 44 seconds for a 'full'

scan, *AVTK* required 40 seconds, and *SCAN* required 2 minutes 52 seconds. This shows the same proportional increase in time for each product. Thank you, *Windows*.

When I tested the same scanners against a hard disk volume which used the *Stacker* data compression system, the results were somewhat different. Using DOS alone, and a hard disk containing 2224 files (79 Mbytes) spread across 130 subdirectories, *Sweep* required 3 minutes 8 seconds (33 seconds when 'quick' scanning), the *AVTK* required 27 seconds, and *SCAN* required 1 minute 27 seconds. Note the large time increase when *Sweep* is carrying out a 'full' scan. Either the presence of *Stacker*, or the large number of files, is causing *Sweep* to perform more slowly. It is instructive that 'Quick' *Sweep* is almost unaffected by all this.

"Sweep is constantly vying for the best detection results in many of the comparative reviews published both here and elsewhere"

The virus detection capabilities of *Sweep* were tested against all the viruses listed in the *Technical Details* section (see below). It correctly detected 100% in both test-sets. Note that the 1024 Mutation Engine samples were all detected correctly, though the complexity of detecting this virus did lead to a very slow scan time. It took 88 minutes 5 seconds to scan the hard disk of the 4.77 MHz PC on which the Mutation Engine samples are stored.

This scan time illustrates clearly that modern powerful hardware has masked just how much work a scanner is really doing when scanning an infected disk rather than a clean one. Fortunately (it is to be hoped) in daily use, a scanner will not be used in such a virus-riddled environment.

As *Virus Bulletin* and *Sophos* (the developers of *Sweep*) obtain their virus test samples from the same sources, the 100% virus detection result is unsurprising. Indeed any result less than 100% would point towards very poor quality control on the part of the software developers.

Bits and Bobs

Included with *Sweep* is a program called the '*Sophos Utilities*'. This is rather like a stripped down version of *PC Tools* or *The Norton Utilities*, in that facilities are provided to inspect and/or manipulate disks, disk sectors, files etc. Quite rightly the manual states that this program is not intended as a replacement for commercially available programs, but it is very simple to use, and is at least available instantly (as long as you do not use 5.25 inch floppy disks!) if the programs on the *Sweep* floppy disk are used in anger and detect the presence of a virus.

'Dangerous' features such as copying sectors or clusters have to be explicitly enabled from the command line whenever they are required. This is a fail-safe feature not offered by products such as *Norton* or *PC Tools*, and prevents a user in panic mode from making things worse.

An option is provided whereby a warning can be issued whenever a file is found which has been previously compressed using one of the common data compression programs (ZIP, ARC and ZOO are the ones mentioned in the manual). This is not really good enough, and there really should be an option where the contents of compressed files actually can be scanned. I am fully aware that the proliferation of several different types of data compression complicates this, and that it introduces a large scanning time overhead, but data compression software is used so frequently that this omission may be a real disadvantage.

Conclusions and Thoughts

In conclusion, *Sweep* is not the fastest scanner around (that honour is probably still held by *Thunderbyte*; see last month's *VB* review), but it provides a scanning speed which is perfectly adequate for most purposes. Testing of *Sweep's* virus detection capabilities showed a perfect score of 100%, and even allowing for the fact that my test-set and *Sophos's* test viruses are from a common pool, *Sweep* is constantly vying for the best detection results in many of the comparative reviews published both here and elsewhere.

Obviously *Sweep's* developers are coping well with the relentless growth in the total number of known viruses. The problems with the *Windows* installation part of *Sweep* are not catastrophic but they do need putting right.

Long-term readers of my reviews will have noticed that the *AVTK* and *Sweep* have been the two 'commercial benchmark' programs against which I compare other scanners. This review bolsters my opinion that *Sweep* should remain in this position.

Technical Details

Product: *Sweep*

Developer: Sophos Plc, 21 The Quadrant, Abingdon, OX14 3YS, England, Tel: +44 (235) 559933, Fax: +44 (235) 559935

Availability: MS-DOS 2.0 and above

Version evaluated: 2.53

Serial number: None visible

Price: £295 for a roving licence with monthly updates.

Hardware used: (a) *Toshiba* 4400C, a 25MHz 486 notebook, with 4 Mbytes of RAM, one 3.5 inch (1.44M) floppy disk drive, and a 120 Mbyte hard disk, running under MS-DOS v5.0 (b) 4.77MHz 8088, with one 3.5 inch (720K) floppy disk drive, two 5.25 inch (360K) floppy disk drives, and a 32 Mbyte hard card, running under MS-DOS v3.30

Viruses used for testing purposes: This suite of 143 unique viruses (according to the virus naming convention employed by VB), spread across 228 individual virus samples, is the current standard test-set. A specific test is also made against 1024 viruses generated by the Mutation Engine (which are particularly difficult to detect with certainty).

Full details of the test-sets used are printed in *Virus Bulletin*, August 1993, p.19.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA

Tel. 203 431 8720, Fax. 203 431 8165

END NOTES AND NEWS

Proceedings of the *Third International Virus Bulletin Conference* are available from *Virus Bulletin* containing all papers presented at the 1993 conference, written by the leading names in the anti-virus industry. The cost of the proceedings is £50 + postage and packing. To order contact Victoria Lammer. Tel. +44 (235) 555139.

Central Point* has announced that its *OS/2* version of *CPAV has entered Beta test. The product claims to be the 'Industry's only true 32-bit *OS/2* application for virus protection that supports key *OS/2* capabilities'. *CPAV for OS/2* is expected to be launched in the Autumn of 1993. Pricing information on the package will be announced at that time. Tel. +44 (81) 848 1414.

According to a report in the *Weekend Australian*, two men are attempting to escape trial for hacking a *NASA* computer by claiming the crime took place in America, one hundredth of a second before the information appeared on their terminal. This would mean that the crime would have to be tried in the USA, and the Australian charges would be dropped. With an ever increasing number of computer misuse cases, it looks likely that solicitors will come up with a complete new range of defences.

Patricia Hoffman's *VSUM* ratings for August: 1. *F-Prot Professional 2.09*, 95.6%, 2. *McAfee ViruScan V106*, 94.9%, 3. *VirusNet 2.08a*, 91.7%, 4. *Dr Solomon's AVTK*, 89.0%, 5. *Fifth Generation UTScan* 28.02S, 83.0%. **NLMs:** *McAfee NetShield v106*, 93.3%, 2. *Net-Prot 1.00s*, 71.0%, 3. *Cheyenne's Inoculan 2.18g*, 67.7%, 4. *Intel LanProtect 1.53+1/93S*, 54.1%.

According to an *American Bankers Association* report, more than nine out of every ten medium-sized banks carry insurance policies which cover computer systems and the electronic transfer of money. The survey indicates that the risk associated with electronic fraud is greater than the physical risk to cash and documents.

S&S International is holding a seminar on Network security on 25th-26th October in Edinburgh. Tel. +44 (442) 877877.

CSI's 20th Annual Computer Security Conference and Exhibition will be held in Anaheim, California, on November 11th-12th. For more information, contact Patrice Rapalus. Tel. +1 (415) 905 2310.

The entire virus problem is solved proclaims the press release! Where has such a revolutionary announcement come from? Frimley (near Bracknell). The press release is from *Pacific Associates*, and is announcing the launch of 'their revolutionary new anti-virus system *Oyster*' which 'claims to be able to protect PCs from attack by all existing viruses and all unknown viruses.' Anyone out there feel as if they have been here before? Tel. +44 (256) 479277.

RG Software has announced that its flagship product, ***Vi-Spy Professional* is now available in Western Europe**. Although *Vi-Spy* has been available in the United States and Canada since 1989, this is the first time purchasers on the other side of the pond will have a chance to examine this well-regarded product. 'From the very beginning *Vi-Spy* has been designed, marketed and supported with the corporate environment in mind', said *RG Software's* founder and President Ray Glath. 'In Europe, we are taking our strategy a step further. We are marketing only into corporate environments that can accommodate a site licence of 100 users or more.' Tel. +33 (1) 3973 9668.

Further details about the forthcoming release of *Novell DOS 7* have been released. The product will be shipped with the *Stacker* data compression program. *Stac* has granted *Novell* a license for *Novell* networks and operating systems which could double the storage available on file servers. *Stacker* will also fully support the new *DOS Protected Mode Service*s, which allows device drivers and TSRs to reside in extended memory on AT computers.