ISSN 0956-9979 MAY 1995

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **An Apple a day.** A short overview of the virus scene in the *Apple Mac* world; which infectors are around, what they do, etc. See p.14.

• **Working towards cooperation.** *ITSEC* is an initiative being sponsored by the government. The organisers are attempting to form established information technology security guidelines. How does this apply to the anti-virus industry? Chris Baxter discusses the 'ins and outs'.

• **Upside-down viruses.** Viruses are a world-wide problem: how is the situation 'down under' in Australia right now? Roger Riordan, of *Cybec Pty*, tells the story from his point of view.

CONTENTS

EDITORIAL

… and in with the new

A bloody coup took place this month in the corridors (well, office) of *VB*, which led to the expulsion of Generalissimo Richard Ford. The power struggle had been brewing for several months, but the previous leadership has now been deported to the US, where he will live in sheltered accommodation and work for the *NCSA*.

Stepping into Richard's shoes is myself, Ian Whalley. Having ousted Richard from power, I can look forward to the work ahead of me. Whilst there may not be any immediate radical alteration to *Virus Bulletin* (a monthly feature on modems? Comparative reviews of sound cards? - I think not), the flavour of the magazine will change as I bring my own editorial touch to bear. In the longer term, however, there is much to be done: the scope of the tests which we perform, what we use in performing these tests, how we perform them, etc. The field is continually changing, and we must change with it.

"in 1989, there were fourteen viruses known for the IBM-PC"

When *VB* was first published in 1989, there were fourteen viruses known for the *IBM-PC*; something which makes me want to describe those days as 'halcyon'. Today, estimates on the number of known viruses vary greatly, fluctuating between 5500 and 7000, according to whose marketing literature you believe, or how you count. Whichever way you look at them, the figures are staggering. If virus writing groups were quoted on the Stock Exchange, their shares would have a world-beating growth record, and, presumably, dividends to match.

On the other side of the fence, things have also moved on. The number of companies involved in producing anti-virus software has risen in response to the growth of the threat, and now in recent years has shrunk again. This reduction was due not only to the increased complexity of viruses (which has favoured those whose technology has advanced ahead of the viruses), but also to large companies swallowing smaller ones (witness *Symantec*).

In this mire of competition, *Virus Bulletin* has a position to maintain. As an unbiased observer, we inform you, the users, what the different groups in the anti-virus community are doing to protect you from the virus threat. In addition, we keep you informed of the new ways in which the virus authors attempt to subvert this protection.

It is in this latter area that we are regularly subjected to 'the end is nigh' stories concerning whatever virus the tabloid press has latched onto this month. Most recently, it was the misinformation concerning the supposed 'Good Times' virus, which is still doing the rounds after about five months [see News article, p.3].

If ever an incident revealed the public's misconceptions about computer viruses, it was this one. Users of PCs, various types of *UNIX* box, and *Macintoshes* (amongst others) all received one form or another of electronic warning. Warnings claimed that the virus would be activated simply by reading an email message (by whatever means), and would destroy large amounts of data across multiple platforms. In reality, such a virus would be extremely difficult, if not impossible to write, and to implement in such a cross-platform manner.

Panic was widespread. Users believed that such a virus existed, and that it had been released across the world's electronic networks. Only a few more level-headed individuals actually stopped and thought, 'Wait a minute - *how* does it do all this?' However, if all the electronic networks have only one thing in common, it is the speed at which gossip spreads: this has made it very difficult to restore calm.

Information is power, the cliché states. This is true; indeed, we provide such information - in the case of the 'Good Times' fiasco, it is in an attempt to calm things down; in other circumstances it may be to warn of genuine danger. To do this (as Richard stated last month), we need your feedback. Please let me know what you would like to see in *VB* - I would welcome your suggestions, your contributions, and also your questions.

VIRUS BULLETIN **©1995 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)1235 555139. /95/$0.00+2.50**

NEWS

Norman Allies with ESaSS

At the *NCSA* conference this month, *Norman Data Defense Systems* revealed that they have signed a 'strategic alliance agreement' with *ESaSS*, the developer of *ThunderBYTE* anti-virus products. The agreement focuses around the technical resources present in both companies, and will form, according to J Arthur Olafsen, President and CEO of *Norman Data Defense Systems Holding AS*, 'one of the strongest R&D teams in the industry'.

Both companies cite the ever-increasing number of viruses requiring analysis, combined with the shortage of people sufficiently qualified to do that analysis, as one of the major reasons for this agreement.

Robin Bijland, president of *ESaSS BV*, stated that he was not interested in cooperation with a large company with many other interests besides data security. 'With *Norman*,' he said, 'it's the same situation as *ThunderBYTE* - they are 100% devoted to data security products'.

David Stang, President and CEO of *Norman Data Defense Systems Inc*, commented: 'The combination of these two companies means we've got more anti-virus programmers among us than probably any other company.' Both companies, he said, were very happy with the agreement.

Bijland said further: '*Norman* has acquired the rights to *ThunderBYTE* technology, and will be permitted to update their scanning engine with our technology. *Norman Virus Control* and *ThunderBYTE* will continue to be distributed through the existing channels.' ▮

Good Times, Bad Times

The panic over the so-called 'Good Times' virus is on the upswing again. This hoax, first reported in *VB* in January of this year, originated on *America On-Line* (*AOL*), and from there, warnings spread far and wide. For a few months it appeared that the furore had died down, but recently the stories have resurfaced, some more improbable than before. One such, received by the editor, stated:

```
This Virus is particularly nasty and opening
it will result in destruction of all data on
your hard disk. If the program is not stopped,
your computer's processor will be placed in an
nth-complexity infinite binary loop, which can
result in processor damage. The Virus spreads
by forwarding itself to every email address
you have in your inbox and sent mail folder.
```

In the continuing absence of any reliable evidence to the contrary, *Virus Bulletin* will continue to treat the story as fiction rather than fact. Any reader with information about the 'virus' is urged to contact *Virus Bulletin* through the usual means ▮

Virus Prevalence Table - March 1995

Virus	Incidents	(%) Reports
Monkey2	46	20.6%
AntiEXE.A	35	15.7%
Form	29	13.0%
Michelangelo	14	6.3%
Stoned	12	5.4%
AntiCMOS	10	4.4%
Sampo	10	4.4%
Natas	9	4.0%
JackRipper	8	3.6%
Parity_Boot	7	3.1%
Junkie	4	1.8%
Monkey1	4	1.8%
Spanish_Telecom	4	1.8%
V-Sign	4	1.8%
Cascade	3	1.3%
Tequila	3	1.3%
Viresc	3	1.3%
Stealth.B	2	0.9%
Taipan	2	0.9%
Green_Caterpillar	1	0.5%
Amse	1	0.5%
Angelina	1	0.5%
Athens	1	0.5%
Datacrime_II	1	0.5%
Diehard.2	1	0.5%
Fair	1	0.5%
Flip	1	0.5%
Halloween	1	0.5%
Jimi	1	0.5%
Joshi	1	0.5%
Keypress	1	0.5%
Tremor	1	0.5%
YMP	1	0.5%
Total	223	100%

Virus Statistics

Readers will notice that our prevalence table has been growing steadily over the past few months: this is not due to the fact that there are more virus reports; rather, that more bodies are reporting their statistics to us. This is a development we encourage: should any body (developers, major corporations, researchers, etc) wish to contribute their statistics, please contact *VB* ▮

IBM PC VIRUSES (UPDATE)

The following is a list of updates and amendments to the *Virus Bulletin Table of Known IBM-PC Viruses* as of 21 April 1995. 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 a dedicated scanner which contains a user-updatable pattern library.

Type Codes

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

Abraxas

CEN: A family of uninteresting overwriting viruses. The Abraxas.15xx pattern detects both the 1508-byte and the 1518-byte variant.

```
Abraxas.1304        33C0 8EC0 2683 3E18 0240 742A 26C7 0618 0240 0026 A120 0026
Abraxas.15xx        B902 00B4 4EBA A801 90CD 21B8 023C 33C9 BA9E 00CD 21B7 4093
```

Armagedon.1079.F

CR: A minor variant which requires a new search pattern.

```
Armagedon.1079.F    018C CBEA 9090 9000 8BC8 8EDB BE00 01BF 3705 8A05 8804 4647
```

Bupt.1261.B

CER: Detected with the Bupt (formerly Traveller) pattern.

Cascade.1701.AA

CR: A minor variant with a slightly modified decryption loop, requiring a new searchstring.

```
Cascade.1701.AA     012E F687 2A01 0174 018D B74D 01BC 8206 3134 3124 464C 75F8
```

Cascade.1701.AD

CR: Minor variant, detected with the Cascade (1) pattern.

Casino.D

CR: Detected with the Casino pattern.

Catholic

CR: The name of this 1129-byte encrypted virus derives from a text message it contains: 'ROB DA MOB FUCK YA ! CATHOLIC CORRUPTED GAY!'

```
Catholic            E800 005D 81ED 0300 B9FF FFAC 4975 FDB9 FFFF AC49 75FD 0E0E
```

Ear

CEN: There are two new variants of this virus: Ear.1024.C and 1026. Both are detected with the Ear-6 pattern. The variants are all flawed in the same way, and will damage almost every EXE file they attempt to infect.

Friday_the_13th.456

CN: Detected with the Friday_the_13th (formerly South_African) pattern.

Hello

CN: A family of several encrypted viruses also known as Hexametricx, as some contain the text 'Hello User, You have got The HEXAMETRICX Virus !!!'. The virus claims to be written in Langen, Germany. The variants are 402, 545 and 547 bytes long.

```
Hello.402           E800 0087 FE5D 87F7 8D76 1E90 E802 00EB 108A 968C 01B9 6E01
Hello.545.A         E800 005D 81ED 0801 8DB6 2601 E802 00EB 108B 9625 03B9 FF01
Hello.545.B         E800 005D 81ED 0701 8DB6 2501 E802 00EB 108B 9624 03B9 FF01
Hello.547           E800 005D 81ED 0801 8DB6 2601 E802 00EB 108B 9627 03B9 0102
```

Hellspawn.1141

P: This companion virus contains the text 'HellSpawn v0.91a (c) 1993 by Stormbringer'.

```
Hellspawn.1141      A113 04B1 06D3 E02D 1000 8EC0 1F1E BE00 01BF 0001 B9D2 04F2
```

HLLO.3853

EN: An overwriting HLL virus.

HLLP

EN: There are two new parasitic HLL viruses, which can both be disinfected: Kasienka and 6176.

HS

CER: This family has two viruses: one of 903 bytes containing the text 'HS86.2 Ver 4.06.Copyright by EA computers inc.1994', and one of 982 bytes, which claims to be version 4.05. This seems to indicate that there are more variants which have not yet been discovered. They are probably of Russian origin, as they appear to be targeted against one particular Russian anti-virus program.

```
HS.903              3DBA DC74 0E3D 004B 740E E82D 009D 2EFF 2E37 00B8 534E 9DCF
HS.982              9C3D 5348 7428 2E80 3EDC 0000 741A 80FC 3D74 2080 FC4B 741B
```

Industrial

CN: This 1841-byte virus, which contains a long text message, including text from the Dire Straits song 'Industrial Disease'.

```
Industrial          80FC 0374 26CD F173 2750 5306 B800 008E C0FA 26A1 C403 268B
```

Italian_Boy

CR: A 578-byte Italian virus, containing the text 'ITALY IS THE BEST COUNTRY IN THE WORLD'.

```
Italian_Boy         3D00 4B74 03E9 C200 9C50 5351 521E 0657 5655 E8BB 00B8 023D
```

Itv.462

CN: This virus, like the other members of its family, somewhat resembles the Vienna viruses, but is nevertheless considered sufficiently different to justify its placement in a separate family.

```
Itv.462          7302 EBAB 8A86 3A03 241E 3C1E 74EE 81BE 3E03 BDFB 77E6 8DB6
```

IVP

CEN, CN, EN: For some reason, the number of new IVP-generated viruses this month is unusually large: Executor.429 (EN), Executor.460 (EN), Executor.473 (EN), Executor.507 (EN), Executor.583 (EN), Infesto.522 (CEN), Infesto.561 (CEN), Infesto.604 (CEN), Infesto.679 (CEN), Infesto.697 (CEN), Replico.317 (CN), Replico.324 (CN), Replico.352 (CN), Replico.390 (CN), Replico.462 (CN) and Replico.478 (CN).

Jerusalem.1801.AR

CER: A minor variant, detected with the Jerusalem-1 pattern.

June_12th

CER: A 2660-byte virus, awaiting analysis.

```
June_12th        56BE 3103 2EF6 1446 81FE 3004 75F6 2E80 3658 0007 5EC3
```

Keypress

CER: There are several new variants of Keypress which are not detected with existing patterns for viruses in the family. The Keypress-gen pattern below will detect the 935- and 995-byte variants, but others (1258 and 1600 bytes long) require separate patterns.

```
Keypress-gen     8EDB C707 0100 B42A CD21 80FE 0475 2D33 D2B4 19CD 2133 DB8E
Keypress.1000    8EDB C707 0100 B42A CD21 80FE 0475 2B33 D2B4 19CD 2133 DB8E
Keypress.1258    FF2E 5C03 CF50 0633 C08E C0BE 4403 2681 3C07 0707 5875 052E
Keypress.1600    7405 C707 3901 F9F5 1FC3 F606 0C07 0174 0D8C C005 1000 0106
```

Khiznjak.823

CER: Of Eastern European origin, as the other Khiznjak viruses.

```
Khiznjak.823     B802 3DCD 2173 03E9 D700 A3FC 028D 1601 038B 1EFC 02B9 0300
```

Khrusha

ER: This 1505-byte virus is probably of Eastern European origin. Although it is fairly old (first discovered two years ago), it has not been listed before.

```
Khrusha          7476 80FC 4F74 7150 5351 5256 5755 1E06 80FC 4B74 4A80 FC40
```

KI

CR: This belongs to the small group of resident, overwriting viruses.

```
KI               601E 0680 FC00 7407 2EC7 0638 039C 9AB8 2435 CD21 5306 1E52
```

Leprosy

CN: There are three new variants this month, 666.J, 666.N, and 666.P, all of which are detected with the Leprosy-B pattern.

Mithrandir.450

P: An old companion virus containing the text ‘Mithrandir III’.

```
Mithrandir.450   264E 019C 2E89 1650 012E 8C1E 5201 3D00 4B75 03E8 D2FE 9DEA
```

Mithrandir.694

CR: This is presumably written by the same author as Mithrandir.450, as it contains the text ‘Mithrandir’, and the viruses share some code fragments. However, this is a standard COM-appending virus.

```
Mithrandir.694   3D76 4275 3B9D 5F07 2681 7D03 4459 7526 83EF 0BFC 061F 5657
```

November_17th.800.C

CER: Closely related to the ‘B’ variant and detected with the pattern given for that virus [*May 1994 p.6*].

PS-MPC

CEN/R: It should not come as a surprise that there are several new PS-MPC-generated viruses this month: 574.G (CEN), 574.H (CEN), 582.A (CEN), 582.B (CEN), 583 (CEN) and Toys.762 (CER).

Skid_Row

ER: This is a family of three EXE-cavity viruses, 415, 418 and 432 bytes long. The word ‘cavity’ refers to the technique of overwriting that section of the first 512 bytes of EXE files which is located directly after the actual header - this space is generally unused.

```
Skid_Row.415     B40D CD21 B452 CD21 FC26 C577 12C5 348C D850 4050 B902 008B
Skid_Row.418     B40D CD21 B452 CD21 26C5 7712 C534 1E8C D840 50FC B902 008B
Skid_Row.432     B40D CD21 B452 CD21 26C5 7712 C534 1E1E 5840 50FC B902 008B
```

Sybille.853

ER: What makes this virus unusual is the fact that it contains a batch file made up of the following three lines: ‘@echo off’, ‘echo Looking for Sibylle’, and ‘goto b’.

```
Sybille.853      3D00 4B75 F350 5351 5256 5755 1E06 1E52 0E1F E8A7 015A 1FB8
```

Syrian.241

CER: A family of two overwriting memory-resident viruses, containing the text ‘Syrian Brain II’.

```
Syrian.241       9C80 FC4B 7402 EB4C B802 3DCD 2172 458B D850 5351 521E 0E1F
Syrian.412       9C80 FC4B 7402 EB4E B802 3DCD 2172 478B D850 5351 521E 0E1F
```

Traveler_Jack

EN: We now have two new variants of this virus; 980 bytes and 1008 bytes long.

```
Traveler_Jack.980B  B98E 03BE 4600 8BFE 8A1E 0900 8A04 32C3 8805 4647 FEC3 4983
Traveler_Jack.1008  BCBC 0480 3E09 0000 7417 8A16 0900 BB41 00BF EF03 8A07 32C2
```

VCL

CN: This month brings a 417-byte variant.

WildFire

CER: A 2222-byte virus, not yet analysed, but containing the text ‘WildFire’.

```
WildFire         3D73 0B75 1281 FB73 7375 0C2E 803E A701 3775 04B4 739D CF9D
Clonewar.923 (3) BA1A 01B9 0000 B800 3DCD 21C3 BA1A 01B9 0000 B43C CD21 7262
```

INSIGHT

ITSEC and Anti-Virus

Chris Baxter

The *Information Technology Security Evaluation Criteria* (*ITSEC*) are a set of techniques agreed between European governments for use in the security testing of computer products and systems. In the UK, an Anti-Virus Working Group, made up of anti-virus developers, researchers, and government staff, is trying to set an *ITSEC* testing standard for anti-virus products. What are its aims, and what has it done so far?

The Problems

It might seem a simple process for users to measure the performance of anti-virus products: unlike many security products, the anti-virus tool is frequently used, and frequent use is the easiest way to uncover the failings in a product. It should be easy, but anti-virus history shows that it is not.

The problem changes rapidly, and rapidly changing problems are difficult to understand. Furthermore, it is difficult to seek impartial advice, as the main repository of expertise is amongst those with a commercial interest in the area, making 'unbiased' comment difficult to accept at face value! [*We try our best at VB! Ed*]

Current Approaches

Early attempts to evaluate anti-virus products illustrate the difficulties. Manufacturers were accused of supporting findings which enhanced their products, even if flawed. Magazines would evaluate many products using only one manufacturer's test collection, or independently proposed unworkable operations, e.g. testing against 'dead' viruses.

Such practices are now, thankfully, little more than history, but there is still no universally accepted approach. The best evaluations recognise this, and provide details of the methods used, which are usually tests against virus collections (one of the few ways of obtaining a swift result).

Interpretation of the output seems simple, especially if presented as a table of numbers of viruses detected. In reality, various questions arise. Is a product scoring 95% three per cent better than one scoring 92%? And if so, what does it signify? Expert studies of any product can also suffer if they dwell on individual aspects which have little or no application in reality.

Government Approaches

Government organisations have long had an interest in evaluating security of their computer systems. Early work in the US produced the *Trusted Computer Security Evaluation Criteria* (*TCSEC*), also known as the 'Orange Book'. This lists sets of security functions, and indicates levels of rigour for testing. Other countries found this a useful exercise, and responded by defining their own specifications.

In Europe, these individual regimes are being integrated into the *ITSEC*. This approach offers a less constraining process than *TCSEC*, though it remains broadly compatible. Briefly, the organisation requiring the evaluation (the Sponsor) makes claims (the Security Target) about the product to be evaluated. These are passed, with the product and its documentation, to a Government-licensed evaluator.

The evaluator tests the product according to a standard schedule. Any problems found are referred back to the Sponsor. If the product meets its Target, a report and a certificate are issued to the Sponsor. The process has a pass or fail finding, and the Sponsor pays for the evaluation.

ITSEC - the Aims

ITSEC offers independent, standardised evaluation. It is still evolving - while it works well in its current incarnation, there are aspects in which it can improve, particularly in such areas as anti-virus products. The process of evaluation assumes you know what the Security Target should be.

> *"cooperation enables us to gain a better appreciation of the threat posed by malicious software"*

Although there is provision for specifying a mandatory Security Target (or 'Functionality Class'), it is not widely used. Currently, it has no mechanics for monitoring dynamic change to the threat. *ITSEC* is, however, designed to be flexible, and provides a sound foundation for developing sophisticated structures for security evaluation.

Its principles depend on a statement of the required security functionality, the Security Target. The first requirements for successful evaluation are to find a method of defining the threat posed by malicious software, and to evaluate a product's capability for defence against that threat.

Such work is, of course, already being performed by experts in the anti-virus industry. The other important issue is that the anti-virus world changes every day: an evaluation made in January of one year will be out of date by February.

Establishing the Protocol

The virus threat comprises many instances of malicious software: this is why virus evaluations are usually carried out by running a product against a large collection. But

which viruses must be detected, and which is it acceptable to miss? Is it acceptable for a product to fail to detect a particular class of virus? How does one deal with a situation where the correct answers to these questions will be incorrect in a few months time?

What is needed is not a fixed statement, but the establishment of a process whereby data can be gathered, assessed and agreed, and used to produce a threat statement. We will use industry expertise and the independence of government to establish a body which has access to many views, and can merge them to produce mutually agreed findings.

Such a process will be of value to all: both evaluators and industry could use a standardised definition of the threat. *ITSEC* evaluators will decide what constitutes adequate defence, in a manner which is defendable and open to discussion by experts. It would not be difficult to alter decisions to accommodate new findings, as *ITSEC* would also be tracking threats and changing their tests constantly. Instead of trying to propose a test methodology, we would be laying groundwork for a process to produce agreement on what constitutes the threat, and to support testing methods.

Tracking the Threat

We need a substantial amount of information to be able to track the threat effectively: data on the number and spread of virus incidents, samples of all known viruses, knowledge of new and different attack techniques, and knowledge of features generally considered essential or 'Best Practice'. These four headings give us enough to quantify the threat, and to define responses which are mutually acceptable to industry, and independently justifiable to product users.

During 1994, we operated a UK-based industry/government working group. Industry members provided data specified above, which government representatives collated and circulated for checking. This has enabled us to develop statements about the threat, and to specify what is necessary in an anti-virus product.

Such requirements fall into two categories - static and dynamic. For instance, 'all common viruses will be defended against' is a static statement, but can only be made in conjunction with the data collection process we are operating, and which defines each requirement more precisely.

Armed with this data, we are constructing a 'Functionality Class' for anti-virus products, which we expect to have operating fully by the end of 1995. We will then be able to prioritise the requirements of a good anti-virus product, and to evaluate against them.

Defining the Tests

Defining a set of tests which ensures that any product meets the threat is only part of the story. To provide an adequate service, we would need to repeat the evaluation for each product, at quarterly or monthly intervals, to ensure that the developing product maintains its capability.

This, however, would be too expensive: we have therefore developed a different approach. Instead of testing the output from the company (the product) at frequent intervals, we will test the ability of the company to maintain the capability of the product directly as part of the initial evaluation.

It will be necessary to use a variety of tests and techniques to measure this capability. Such techniques already exist; for example, in the procedures used for accrediting test laboratories, such as the quality standard BS5750. In fact, we have an advantage over such tools, as we are measuring something less nebulous. We need to determine capabilities of the company for finding out new threats and dealing with them.

"end-users will soon be able to specify a quality standard for anti-virus products"

Such tests cannot be as precise as testing a product against a set of claims. They are more likely to involve tracking the processes in the company by which virus threat information is gathered and fed into the product. Critical dependencies or bottlenecks would be noted for later monitoring, and evidence would be required from the company to prove that these processes would continue to function.

This exercise cannot provide high levels of assurance that a product will always be good; thus we will require retesting at certain intervals. Testing the ability of the company to maintain their product will enable us to make the intervals longer, and at the same time have more assurance that the product will remain sound.

These processes are now essentially in place, and output such as lists of common viruses in the UK can now be produced. We are beginning detailed work on the second stage of the method; evaluating product maintenance.

The Future

At present our work is directed towards having our evaluation process in place by the end of the year. While this is currently only a UK initiative, we are working with Germany to develop these proposals, and hope they will be adopted as the standard in Europe at least.

We hope this approach to anti-virus evaluation will benefit all those involved. Cooperation of this sort enables us to gain a better appreciation of the threat posed by malicious software, and anti-virus companies may welcome clear statements of requirements for which they can develop new products. Most importantly, end-users will soon be able to specify a quality standard for anti-virus products involving independent and comprehensive testing against the nearest approach to the real-world threat which we can achieve.

Further information on the *ITSEC* initiative is available from the *Certification Body*, PO Box 152, Cheltenham, Gloucester, GL52 5UF, UK. Tel +44 (0)1242 238739 extension 5103.

VIRUS ANALYSIS 1

RMNS - The Perfect Couple

Eugene Kaspersky
Kami Associates

The evolution of the programs we call 'computer viruses' continues relentlessly. Today, there are at least two which have gone beyond a single-celled basis and started to replicate by dividing their code into two different components - they are known as 'multicellular' viruses (not to be confused with multipartite viruses).

The first of the 'multicellular' viruses is Dichotomy [*see Virus Bulletin, December 1994, p.8*], which has two components: 'odd' and 'even'. When a file infected with the 'odd' component is executed, the virus looks for a file infected with 'even' code, installing itself into memory only if that part is found.

Now, viruses may be abandoning their purely monosexual existence: the RMNS virus may be a further step towards more complex electronic creations. It appears that the word 'virus' may no longer be the best term to describe such programs - RMNS does not look like a biological virus, but more like an 'electronic creature'. This begins another branch of electronic evolution: the era of viruses of a specific sex.

The Virus and its (His?) Sex

RMNS gets its name from the internal text string which is placed at its end. Like Dichotomy, the code of the RMNS virus is divided into two parts ('male' and 'female'). Here, however, the similarity ends. The two parts of RMNS install themselves into memory independently of each other.

The names 'male' and 'female' derive from text descriptions held within the two parts of the virus:

male: `R.M.N.S Test virus R.M.N.S MW Man`
female: `R.M.N.S Test virus R.M.N.S MW Woman`

Infection can only take place if both sections of the code are resident in memory at the same time and on the same computer.

The male and female parts of the virus are very similar: they are both placed at the end of COM files, they both receive control when the infected file is executed, they both issue 'Are you there?' calls, and they both hook Int 21h and stay resident. They also have similar lengths - the male code is 297 bytes long, and the female is 353).

The differences between the two parts are few but important: the male does not infect files, but only intercepts their execution; the female does not intercept execution of the files, but infects them on request from the male.

Installation and Int 21h Hooking

When an infected file is executed, that part of the virus with which it is infected receives control with a JMP instruction. The virus then restores the three bytes at the beginning of the host program which were overwritten on infection.

Next, the virus decides whether or not to go resident: it is made up of two parts, and each part will only go resident in memory if it is not there already. The virus code in the infected program issues an 'Are you there?' call using Int 21h - for the male code, the AX register is set to 4BBCh, and for the female it is 4BBDh.

Both the male and female sections of the virus return the ID value BBB4h in the AX register to show that they are present in memory.

The segments of the virus install themselves at the top of system memory, using the standard methods of direct manipulation of Memory Control Blocks, and hooking Int 21h. After this, control passes to the beginning of the host program.

The male and female parts each intercept only one Int 21h function: AH=4Bh (Load and Execute). Both parts check the subfunctions of the Load and Execute call and execute the following corresponding routines:

- Male code:
  a) AL = BCh. 'Are you there?' call, returns BBB4h in the AX register.
  b) AL = 0, 1, 2, or 3. The file being loaded is checked, and the female part called, using Int 21h with 4BBEh in the AX register, to infect the file.
- Female code:
  a) AL = BDh. 'Are you there?' call, returns BBB4h in the AX register.
  b) AL = BEh. Performs the infection routine.

Note that only the male part intercepts the system generated Load and Execute subfunctions (i.e. 0, 1, 2 or 3).

Infection

On a Load and Execute call, the male part opens the corresponding file, reads three bytes from the beginning, and compares the first byte with the character 'M' in order to prevent infection of EXE files. Then the virus checks the date and time stamp of the file for the value 00FF00FFh (31.07.80; 12:07am). This is the virus' ID stamp, and if it is found, the file will not be infected.

If the file concerned is not an EXE file, and it is not yet infected, the virus calls the female part of its code with an 'Infect it' call (Int 21h, AX=4BB4h). The male part passes

its length in the CX register (CX=0129h), the segment address of its code in the DS register, and the file's handle in the BX register.

After receiving the 'Infect it' request, the female section of the virus checks the length of the file. If the file is longer than 65024 (FE00h) bytes, it will not be infected.

The infection routine then selects the part of the virus with which to infect the file, by using the system timer. It will, 50% of the time, write the male code (using the length and segment address received in the CX and DS registers), and 50% of the time it will write the female code (by overwriting the values in CX and DS with the appropriate values for the female section of the code).

Then, the infection routine overwrites the head of the file with a JMP VIRUS instruction, sets the file date and time stamp to 31.07.80, 12:07am, and returns control to the male part of the code. Thus, the file is infected either with the male or the female code, but not with both at the same time.

The virus does not perform standard virus routines, such as hooking Int 24h during infection to prevent the DOS error message whenever an attempt is made to write to a write-protected disk. It neither saves, clears, nor restores the file's attributes, and overwrites the file time and mask stamp with its ID value. However, the many minor defects in this virus cannot belittle its importance in the history of these electronic creatures.

RMNS

Aliases:	RMNS MW.
Type:	Memory-resident, parasitic file infector.
Infection:	COM files.
Self-recognition in Files:	Compares file's date and time stamp with 00FF 00FFh (31.07.80, 12:07am).
Self-recognition in Memory:	'Are you there?' calls with Int 21h, AX=4BBCh, AX=4BBDh. The memory-resident code returns BBB4h in the AX register.
Hex Pattern in Files and Memory:	`BF84 0101 F78A 05A2 0001 478B 05A3 0101 B8B? 4BCD 213D B4BB` (The wildcard is replaced in the 'woman' by D, and in the 'man' by C.)
Intercepts:	Int 21h for infection.
Trigger:	None.
Removal:	Under clean system conditions, identify and replace infected files.

FAREWELL

From the departing Technical Editor

I have been active in the anti-virus arena for more than six years, and Technical Editor of *Virus Bulletin* since early 1990. During this period, I have observed the explosive growth of viruses, and have tried to do my best to combat the problem in various ways. For various reasons, I have decided to leave my post at *Virus Bulletin*, knowing that a qualified successor will be appointed soon.

The computer virus field has changed dramatically, from being barely noticed back in 1989 through the Michelangelo media hype and into the current phase of 'old news'. Today, the user community is well aware that viruses exist, but most new viruses just provoke a 'ho..hum' reaction, although they may be much more advanced and difficult to handle than the viruses which originated a few years ago.

Many of those who were active in the anti-virus field six years ago have retired, and many of the products which used to exist have faded away as well. New products have occasionally appeared, but it is becoming obvious that starting from scratch is now almost impossible - six years ago, a single individual working part-time could develop a comprehensive anti-virus program, but today that would take a sizeable team.

The virus-writing field has changed as well - the virus writers and distributors are becoming ever bolder and more aggressive; the recent posting of over two thousand viruses to the alt.comp.virus newsgroup being the best example.

The problem is becoming ever harder to deal with, and although several companies are still able to produce anti-virus products which are more or less effective, there are growing signs of despair in the anti-virus community. Perhaps as a result of this the number of mergers and 'strategic alliances' between anti-virus companies seems likely to increase even more in the future.

I have as Technical Editor made various contributions to *Virus Bulletin*, and although I consider some of it to be of negligible value, I hope that a good deal of my work has been of use to the user community.

I would like to take this opportunity to dismiss any rumours of my permanent retirement from the anti-virus field: it does not seem likely that the virus problem will disappear in the next few years, and I see plenty of work waiting ahead.

So, good-bye and a virus-free future to you all!

Fridrik Skulason
Technical Editor

VIRUS ANALYSIS 2

Angelina: Angel of a Virus?

Benjamin Sidle
Sophos Plc

Yet another boot sector virus has joined the ranks of 'infectors at large': Angelina has become established in the wild, both in the UK and worldwide. In fact, the majority of viruses found in the wild are boot sector viruses: the most common method of transmission is by booting from an infected floppy which is not scanned before being used.

An Uninteresting Character

Apart from its one distinguishing characteristic (i.e. being in the wild), this virus is a completely unremarkable creature containing the usual childish style of message, which is feebly encoded and never displayed:

```
Greetings for ANGELINA!!!/

by Garfield/

Zielona Gora
```

The last line of this message also appears the file virus Reverse. 'Zielona Gora', the name of a town in Poland, is Polish for 'Green Hill'.

First Faltering Steps

When an infected floppy or hard disk is booted, the virus lowers the available memory by 1 Kilobyte, by altering the value at memory location 0000:0413h in the ROM BIOS data area. It then copies itself to this reserved area.

Next, Angelina stores the address of the original Int 13h handler in the same area where the copy of the virus code is located. The entry to the Interrupt Vector Table is then modified, making the Int 13h handler point to a new handler within the reserved area. Finally, the virus issues an Int 19h call (soft reboot), re-starting the boot procedure, but this time using the new Int 13h handler.

From now on, whenever an attempt is made to read sector 1, side 0, cylinder 0 of a disk (which is the boot sector for diskettes, and the Master Boot Sector of a hard disk) the read is intercepted. All other reads and Int 13h functions are passed straight to the original Int 13h handler.

Infection

Once this attempt to read the boot sector is intercepted, the virus reads the sector using the original Int 13h and checks to see whether or not it is infected, by comparing the word at offset 00F0h with C681h. If the disk is not infected, the virus will infect it.

In the case of a floppy disk, the original boot sector is copied to the last sector of the root directory. On a hard disk, the original Master Boot Sector is copied to sector 2, side 0, cylinder 0, making use of what would otherwise be 'dead' space in that area. However, if the disk is already infected, the read will be stealthed, and pointed to the copy of the original sector.

Conclusion

Angelina has no noteworthy features. It exists only to propagate, and is little more than another pointless 'wannabe' effort.

Although this virus does not carry a destructive payload, there are boot sector viruses in the wild which do. The importance of checking incoming diskettes for viruses cannot be overstressed: the few seconds spent scanning a disk may mean the difference between a fully operational PC and a minor catastrophe, or something worse.

In the UK, any virus attack can be reported to the *Computer Crime Unit* at *New Scotland Yard* on 0171 230 1177: only with the help of the user community can the activities of virus writers be stopped.

Angelina

Aliases:	None known.
Type:	Memory-resident Master Boot Sector virus with stealth capabilities.
Infection:	Master Boot Sector of hard disk, boot sector of diskette.
Self-recognition on Disk:	The word at offset 00F0h, which is set to C681h if the sector is infected.
Self-recognition in Memory:	None.
Self-recognition in Files:	None.
Hex Pattern:	`BB33 0080 8750 7D22 4B75 F8A1 4C00 26A3 8401 A14E 0026 A386`
Intercepts:	Int 13h for infection.
Trigger:	None.
Removal:	Under clean system conditions, use the FDISK /MBR command.

VIRUS ANALYSIS 3

N8FALL: The Nightmare Bug

Matt Brown

With a few exceptions, most viruses in the wild are simple, reliable and uncomplicated beasts: N8FALL, recently received from a user in Germany, is one of the exceptions. It is substantial (approximately 5,800 bytes long); worse, its operation is complicated, and its methods convoluted. A text string inside the body attributes the virus to 'Neurobasher', who is responsible for several other viruses in a similar vein, his best-known creation being Tremor.

Installation

The virus seeks itself in memory by testing the value at 0000:05E0h. When resident, this and the following bytes form a JMP VIRUS instruction. N8FALL checks this, testing the jump and the code to which it points.

If the virus is resident, the remainder of its code is skipped and the host program run. Otherwise, it calls a routine which checks the Int 13h, the Int 21h and the Int 2Ah vectors for a variety of anti-virus TSRs. If any are found, their code is modified to disable checking. Large amounts of N8FALL's code appears to have been written with *ThunderBYTE* in mind, as well as certain other products which try to identify any suspicious code.

Next, the virus checks to see if HIMEM.SYS is installed, in which case, N8FALL attempts to go resident in upper memory. Once it has relocated itself and transferred control, it adds itself to the Int 21h handlers, calling Int 21h, function 52h, to get the DOS List of Lists.

The virus is not interested in the list itself, only the segment where the DOS interrupt handlers reside. It searches this segment for the Int 21h handler, and overwrites the area immediately after the jump to the high-memory portion of the handler with a far call to the jump it had previously placed at 0000:05E0h. If DOS is not loaded into high memory, the virus overwrites the low-memory handler.

N8FALL then opens COMMAND.COM and immediately closes it again: this will infect that file, as it passes through the Int 21h handler which has just been installed. Next, the virus decrypts the string 'C:\NCDTREE\NAVINFO.DAT' (the name of a file used by *Norton Anti-Virus*), stored near the end of the virus code: however, I could find no further reference to it. Control then returns to the host program.

Resident Behaviour

N8FALL's Int 21h handler intercepts a large number of DOS functions. The virus hides the increase in the size of infected files when a directory is scanned, using FindFirst/FindNext. Interestingly, if a directory listing of drive C is done from drive A, this increase is not hidden, but if the same listing is done from drive C, it is. The virus also stealths LSeek to End (AX=4202h), Read (AH=3Fh) and Write (AH=40h); in all cases to hide the increase in length of the infected file.

Int 21h, function 44h, subfunction 52h is also intercepted: if this is called, the virus stops stealthing FindFirst/FindNext calls. The only documented use for this function is a *DR-DOS* version check, but I am not sure if this is related.

FindFirst/FindNext stealthing is also avoided if a program calls Get Drive Parameter Block (Int 21h, AH=32h): this stops programs like CHKDSK being confused by the fact that the amount of space in use on the drive does not match that occupied by files. When the virus next attempts to infect anything, it will reactivate the stealth functions.

When the DOS function Set Interrupt Handler (Int 21h, AH=25h) is called, the virus patches the interrupt handler being installed in the same manner as that used on first execution of the virus, to disable a list of anti-virus TSRs.

The handler for the DOS functions Allocate Memory and Resize Memory Block (AH=48h, 4Ah) will lie about the amount of free memory available if a request fails due to lack of memory. This may cause software problems.

Files are infected via handlers for the DOS functions Exec (Int 21h AX=4B00h), Open (Int 21h AH=3Dh) and Close (Int 21h AH=3Eh). The handler is designed to be re-entrant: on entry to the infection routines, the code pointing to them is changed to return immediately; after they exit, it is changed back. This process allows the virus to perform Int 21h calls itself without fear of confusion.

Infection

The infection routine first checks to see if the filename contains one of a number of strings (including 'PL', 'AV', 'MI', and 'CH'), in which case it does not infect. Presumably, these are either anti-virus programs or other programs which may be damaged by an infection.

Then, it sets the DOS DTA (Data Transfer Area) in order to use the DOS functions FindFirst and FindNext (AH=11h, 12h, 4Eh, and 4Fh) without altering the contents of caller's DTA. After this, the Int 24h handler is set to inhibit DOS error messages, and Microsoft's TSR, VSAFE, is disabled. If the file to be infected is on a diskette, N8FALL tests to see if the disk is there and write-enabled, before proceeding.

Then virus looks at the lower five bits of the file length - if they are all set to 1, it may be infected, so further tests are done. The last 24 bytes of the file are read and decoded

using the following algorithm: the first 23 bytes are XOR-ed with the final byte (the encryption key), and their relative position subtracted from this result. The resulting bytes are then tested. If the file is found to be infected already, it is passed to another routine, which runs it.

If the file is not infected, the virus reads the first few bytes of the file, seeking the marker 'MZ' (or 'ZM'). This determines whether it is an EXE or a COM file: a slightly different path is followed for each.

Where a COM file is concerned, the virus attaches itself to the file end, replacing the first three bytes of the host program with a JMP VIRUS. In the case of an EXE file, it modifies the initial IP fields in the EXE header to point to the virus instead.

Either way, control eventually passes to the polymorphic header generation routines. These are quite involved, and use the stack to keep track of the many recursive calls. They produce most of the instructions in the *Intel* instruction set, including some only found on 386 and later processors.

However, the headers produced have distinctive patterns which make initial identification much easier than it might otherwise be. Despite its complexity, the polymorphic generator is not as clever as many other features inside the virus: merely generating a large proportion of the instruction set does not in itself make a virus hard to detect.

Companion Virus and Trigger Routine

Instead of infecting an EXE file, N8FALL will sometimes drop a 527-byte, non-polymorphic companion virus instead. This companion will replicate without the assistance of its 'parent', and is a fully functional virus in its own right. Its code is stored almost at the end of the parent virus, and is XOR-ed with the value 0033h. After the creation of the companion, the virus will occasionally print a text message.

The companion virus intercepts Int 21h functions 005Bh (Create New File), 003Ch (Create File) and 4B00h (Exec). Self-recognition in memory is performed by testing the word at memory location 0000:05D2 for the value 5832h: if it matches, the virus is already resident.

The Int 21h handler performs a few checks before moving on to its infection stage; if the function was 4B00h, it will not infect if the pathname begins with 'A:\' or 'B:\', or if the program name contains the letters 'F-' (presumably F-PROT.EXE).

It then creates a matching COM file, with attributes of System, Hidden and Read-only, to which it writes a copy of itself, setting file date/time to 11:55:00, 01 January 1994. Many of the operations in this part of the virus are done in a roundabout way, obviously to avoid heuristic detection.

The companion virus hides its presence in directory scans (Int 21h functions 11h, 12h, 4Eh, and 4Fh), but makes no other efforts at stealth.

N8FALL will occasionally, after dropping its companion virus, print the following message, and wait for a key to be pressed before continuing:

```
"Any means necessary for survival"
_ N8FALL/2XS _
"By the perception of illusion we experience
reality"
Art & Strategy by Neurobasher 1994 - Germany
"I don't think that the real violence has even
started yet"
```

Conclusions

N8FALL is an extremely complicated virus which took several days to analyse; I hope that this does not indicate another trend among virus writers. However, the sheer complexity of the virus, especially the manner in which the virus uses self-modifying code at almost every opportunity instead of maintaining status variables, is beyond the efforts of most programmers.

I must respect Neurobasher's programming ability, but at the same time I wonder what kind of person would devote so much time (for this can only represent many months of effort) to a task which is far easier for me to decipher than it must have been for him to create.

N8FALL

Aliases:	None known.
Type:	Memory-resident parasitic, polymorphic file infector with stealth capabilities. May also drop a 'child' companion virus.
Infection:	COM and EXE files.
Self-recognition in Files:	Lower five bits of file length are set to 1. There is an encrypted byte pattern in the last 24 bytes of file.
Self-recognition in Memory:	0000:05E0h contains a far jump to the Int 21h handler.
Hex Pattern in Files:	No searchstring possible.
Hex Pattern in Memory:	`891E 6C03 BB6C 0389 470A 8957 1A89 4F14 8977 0D89 7F17 8C47`
Intercepts:	Int 21 for infection, stealth and trigger routine.
Trigger:	Occasionally, after infection, prints a message and waits for a key to be pressed.
Removal:	Under clean system conditions, identify and replace infected files.

CONFERENCE REPORT

‘Security on the I-WAY’

Even as the air hostess pours out my second gin and tonic, and the plane reaches cruising altitude, my legs stretch into the luxurious five inches of space between seats you get in tourist class, and my mind stretches back - conference, conference, conference...

‘Security on the I-WAY’ was the title of this *NCSA* conference, which took place in Washington DC on 10/11 April. The first day kicked off with a keynote address by Peter Tippett (formerly of *Symantec*, now president of the *NCSA*). This focused on the concept of ‘convergence’ of digital systems, and the threats inherent therein. As technology becomes increasingly interlinked around us, we will be ever more susceptible to one form of digital attack or another.

After the keynote, the conference split into two streams. Track One was entitled ‘Viruses on the I-WAY’; Track Two, ‘Information Infrastructure’. Now, whilst I have at least a few talents, attending two presentations simultaneously is not one of them, so I will concentrate on the virus track, for the simple reason that I was there.

Highlights of this stream included presentations from, among others, Frans Veldman (of *ESaSS BV*), Alan Solomon (*S&S International*), and a joint talk from Sarah Gordon (now of *Command Software Systems*) and Richard Ford, *VB*’s outgoing editor.

Ford and Gordon - Double Trouble

Richard Ford (recently departed these hallowed halls for the *NCSA*) and Sarah Gordon made an interesting double act for the first presentation. Their topic, ‘Real world anti-virus product reviews and evaluation’, was discussed using several role-playing scenarios, in which Gordon played the part of a prospective buyer of a (fictional) anti-virus product, and Ford took various roles - ranging from the (entirely believable) friend, to the (frankly mind-boggling) editor of a specialist magazine in the field.

The problems with all forms of product recommendation and evaluation were convincingly demonstrated by both presenters. Ford closed with an overview of the *ITSEC* methodology (more specifically, the additions necessary for anti-virus products) used to certify software, currently under development in the UK (see article p.6).

Us versus Them: The Battle Continues

Frans Veldman spoke on ‘Virus Writing: High-tech Info Security Warfare’: his presentation examined the close relationship between developments in the rival fields of viruses and anti-virus products.

These two areas exhibit links which are, from a user’s point of view, disturbing - new methods in one lead rapidly to new methods in the other. The unsettling part comes when you realise that virus authors examine techniques used by anti-virus systems, incorporating knowledge gleaned in their programs to evade detection by anti-virus software.

Audience discussion on the topic was heated, not least because of some of Veldman’s theoretical escapes from this situation. His favoured way out is to claim that the most difficult viruses to detect present little challenge to the anti-virus developer, and vice-versa: this misinformation, fed to virus authors, encourages them to proceed with their development in a direction which makes detection easier.

32 Bits of What?

At the start of the second day, two talks were presented which addressed the virus problem on *Microsoft’s* new generation of operating systems - *Windows 95* and *Windows NT*. These operating systems were examined by Shane Coursen (*Symantec*) and Charles Rutstein (*Price Waterhouse*) respectively.

> *“virus authors [incorporate] knowledge gleaned in their programs to evade detection by anti-virus software”*

Audience participation after each of these talks was extensive, and proved that the topic of viruses on operating systems other than DOS is moving more and more into the public eye. With *Windows 95* still due for release in August, interest in the field can only increase.

Virus Evolution

Dr. Alan Solomon gave his ever-popular ‘The good, the bad, and the polymorphic’ talk, in which he discussed the development of the polymorphic virus in all its forms. Like polymorphic viruses, the talk is always changing - this time, it was the hats again. Solomon wore a black hat whenever he took the role of the virus author, and a white hat when he switched to being a product developer.

Lest We Forget

This conference, like most others, was not all work. Many theories were cooked up around the bar: perhaps the most notable (and reportable) of these was one which is surely to become one of the great conspiracy theories: ‘Is Peter Norton an urban legend?’. All in all, a conference well worth attending.

TUTORIAL

Mac Viruses: An Update

It is a fact that most computer viruses in existence today are targeted against the *IBM*-compatible PC. *IBM* has always been eager to see its technology spread, which has led to the existence of many clones: indeed, most people, when referring to a PC, mean an *IBM*-compatible.

This has an unfortunate side-effect: viruses which affect *IBM-PCs* also infect *IBM*-compatibles, allowing these infectors to disseminate widely. More research, more resources, more time has thus been expended in this area, by virus authors as well as anti-virus developers and analysts. The *Apple Macintosh*, on the other hand, has for various reasons remained a relatively restricted platform.

Macintosh Develops

The *Macintosh* is a more difficult machine to program (and thus to write viruses for) than the *IBM-PC*, and fewer 'hobbyists' have a *Mac*. Also, until now, *Apple* has resisted allowing its technology to be cloned. Both these factors mean that not only is the machine less widespread than the *IBM*-compatible, but that fewer people program for it.

Recent commercial developments in the *Mac* world indicate that at least the availability side will be changing. *Apple* has started licensing its *Mac* technology: *Daystar Digital*, *Power Computing*, *Radius*, and *Pioneer* have already begun to produce *Macintosh* clones. With the expected growth in ownership of the machines (and possible reduction in price?), we can expect to see a corresponding increase in *Macintosh* viruses.

Virus Protection on the Apple Macintosh

Although there are more anti-virus products for the *IBM-PC* than any other platform, *Mac*-specific scanners are available from a number of companies. Additionally, more and more companies whose products until recently excluded detection of *Apple Mac* viruses are investing in this area. *Macintosh* packages currently on the market include (with developers' names in parentheses):

- *MacTools* (*Central Point Software*)
- *Disinfectant* (freeware, John Norstad, Academic and Computing Services, *Northwestern University*, IL, USA)
- *GateKeeper* (freeware, Chris Johnson; Email chrisj@emx.cc.utexas.edu. The product, like *Disinfectant*, is available from many *Mac* sites on the *Internet*.)
- *Symantec AntiVirus for Macintosh - SAM* (*Symantec*)
- *Virex* (*Datawatch Corp.*, NC, USA) This program is also available with *Datawatch's Superset Utilities*

Viruses on the Apple Mac

The first *Apple Mac* virus, nVIR, appeared in 1987 in Europe, and in the USA in 1988. This infector also holds the position of one of the two most common *Mac* viruses; the other being WDEF.

In comparison with the *IBM*-compatible PC, there are very few *Apple Mac*-specific viruses; those known number in their dozens, as opposed to the thousands which have been written to target the *IBM*-compatible machine. The table on the opposite page shows a list of most *Mac* viruses known to be in the wild. In addition to the viruses, there are Trojan horses which affect the *Apple Mac*, including Merry Xmas, China Talk, Mosaic, and Font Finder.

New Mac Virus Variant

A new nVIR B variant named 'CLAP' was detected in the last week of March 1995. The nVIR B virus infects the System file, spreading immediately to applications. It can infect the Finder, though certain applications are immune. Once the System file is infected, a counter is initialised to 1000: this decrements on system startup and when an infected application is run. When the counter reaches zero, the *Mac* will beep randomly on power-up, or when running an infected application.

CLAP exhibits some of the usual characteristics seen in nVIR variants; the virus writer has simply modified the Resource type name from nVIR to CLAP. Specifically, this variant has been modified to avoid detection by the virus scanner *Disinfectant 3.5*. It would appear that the sole purpose of the changes to this virus has been to evade scanners and protection INITs. A new version of *Disinfectant*, 3.6, has been released, and is available through the usual electronic channels.

What's in a Name?

Many things can go wrong with a *Mac*; few will be virus-related. The most common problems are caused by software errors, damaged desktop displays (it should be noted, however, that the Scores virus will change the display of the Notepad and Scrapbook icons), and damaged applications.

Several things may indicate a virus infection, including growth of applications (not always a virus), distortion of pull-down menus, patterns inconsistent with normal operation across several files, and INITs and System extensions ceasing to load for no apparent reason.

Despite the fact that there are far fewer *Mac* than *IBM-PC* viruses, the regular use of anti-virus software is important: even viruses without destructive triggers can cause problems. Further discussion is planned for future issues.

Virus Family	Variants	Aliases	First seen	Origin	Payload	Side-Effects	Comments
nVIR	nVIR A, nVIR B	none known	1987	Europe	When pre-set counter decrements to zero, machine may beep or display the message: 'Don't Panic'	none known	Strains A & B can 'mate': resultant 'progeny' reported both as A & B. Other viruses in family: AIDS, Hpat, MRV#, nFLU, Jude, prod, Modm, Zero, F*CK, nCAM, CLAP, nVIR F
INIT 17	none known	none known	1993	Canada	Displays message: 'From the depths of Cyberspace'	Causes crashes, particularly on Macs with the 68000 processor	Spreads well on Systems 6 & 7. Triggers after 31.10.93
CODE 1	none known	none known	1993	USA	Renames hard drive 'Trent Saburo' on trigger date	Can cause crashes	Spreads well on Systems 6 & 7. Triggers on any October 31st
INIT 29	INIT 29A, INIT 29B	none known	A: 1988; B: 1994	unknown	When a locked diskette is inserted on an infected system, message displays: 'The disk "xxxx" needs minor repairs. Do you want to repair it?'	Problems printing from infected system. May have problems with MultiFinder (System 6), and incompatibilities with startup documents	Can infect applications when not actually running
INIT 9403	none known	SysX	1994	Italy	Crashes disks, attempts to erase disks connected to the system, to destroy disk information on connected hard drives, and to destroy boot volume	none known	Spreads under Systems 6 & 7; thus far known only on machines using the Italian Macintosh system
WDEF	WDEF A, WDEF B	none known	Dec 1989	Belgium	none known	Causes system crashes and problems with font style display; can damage disks	Will not infect System 7
Scores	none known	ERIC, San Jose Flu, NASA, VULT	1988	USA	Changes icon display	Technical errors may cause crashes	Some applications immune. Replaces resources in System software release 6.0.4
ANTI	ANTI A, B & O, ANDI-ANGE	none known	A: Feb 1989; B: Sept 1990	France	none known	Can damage applications, making repair difficult	Does not infect System 7, or System 6 under Multi-Finder
INIT 1984	none known	none known	Mar 1992	USA and NL	File/folder names change to random 1-8 character strings; file creators and types changed to random 4-character strings. Some files deleted	Affects all types of Macintosh. Causes crash on startup of old Macintoshes (128K, 512K, and XL)	Damages Systems 6 & 7
ZUC	ZUC-A, ZUC-B, & ZUC-C	none known	A: Mar 1990; B: Nov 1990; C: June 1991	Italy	Cursor moves diagonally across screen when mouse button depressed, changing direction and bouncing when any side is reached	A & B can cause disktop pattern to change. All three can cause long delays and unusually large amount of disk activity	Infects applications only, not System files or document files
MacMag	none known	Aldus, Brandon, Drew, Peace	Dec 1987	Montreal	Displays message of peace, then self-destructs	none known	Spreads only to System files
MDEF	MDEF A, MDEF B, MDEF C, & MDEF D	A: Garfield, B: TopCat	A: May 1990; B: Aug 1990; C: Oct 1990; D: Jan 1991	Ithaca, NY	none known	MDEF C contains an error which can cause crashes and other problems. MDEF D can damage some applications irreparably	Author of this virus also wrote CDEF virus
Frankie	none known	none known	unknown	unknown	Draws bomb icon, then displays message: 'Frankie says: No more piracy!', and causes system crash	Causes no damage to Apple Macintosh computers	Only affects some types of Mac emulators running on Atari computers. Applications do not have to run to be infected
CDEF	none known	none known	Aug 1990	Ithaca, NY	none known	Has been known to cause some problems	Infects only Desktop file (used by Finder), but not on System 7
MBDF	MBDF A, MBDF B	none known	Feb 1992	Wales	none known	Long delay on initial infection often prompts users to restart, which may result in an irreparably damaged System file	Installed by Trojan horse called 'Tetricycle' or 'tetris-rotating'
CODE 252	none known	none known	Apr 1992	California	Displays message: 'You have a virus / Ha Ha Ha Ha Ha Ha Ha / Now erasing all disks / Ha Ha Ha Ha Ha Ha / P.S. Have a nice day / Ha Ha Ha Ha Ha Ha Ha / (Click to continue...)'	An error can cause crashes, and may damage files under System 7	No files or directories are deleted by this virus. It spreads to new applications under System 6 without MultiFinder. Triggers between 6 June and 31 June in any year
T4	T4A, T4B, T4C, T4 Beta	none known	June 1992	Worldwide	May display message: 'Application is infected by the T4 virus'	Can damage applications irreparably	Systems 6/7: post-infection, INIT files and system extensions will not load. Trigger dates: T4A - after 15.08.92; T4B - after 26.06.92
INIT M	none known	none known	April 1993	USA	File/folder names and file creators/types changed to random strings. Changes icons; destroys association between programs and their documents. May delete files	Can cause problems with display of windows	Although damage caused by this virus is similar to that caused by INIT 1984, the viruses are technically very dissimilar. Triggers on any Friday 13th
Dukakis	none known	none known	August 1988	USA	Displays message: 'Greetings from the HyperAvenger! I am the first HyperCard virus ever. I was created by a mischievous 14-year-old, and am completely harmless. Dukakis for president in '88. Peace on earth and have a nice day.'	none known	Infects only HyperCard stacks
HC	none known	none known	unknown	unknown	none known	Causes machine to hum strangely. HC painting tool system appears onscreen in random places	Infects only HyperCard stacks

FEATURE 1

There's Viruses Down Under!

Roger Riordan and Jakub Kaminski
Cybec Pty

Things are fairly quiet on the virus front here in Australia; it is a year or more since we found a completely new virus, and the press has become bored with the whole business. However, our business is prospering, and we at *Cybec* are working hard to keep up with new developments.

We would probably never have got into the game were it not for the havoc wrought by Stoned in a labful of *Olivetti M24s*, in which the DOS boot sector and FAT followed immediately after the Master Boot Sector in track zero. Stoned, and its offspring, has discouraged the use of track zero, but these early clones have nearly disappeared, leaving such disasters as little more than folk history.

The recent introduction of high-capacity IDE drives (which rely on a special driver loaded from cylinder 0, track 0) has brought back this incompatibility. Many boot sector viruses clobber this driver, but virus writers are not the only ones who assume track zero is free for the taking: we have heard several cases where disk optimisers have locked up drive C.

A Local Point of View

By now many of you may be saying, 'So what's new?': indeed, you would probably feel at home here. However, there is significant local colour, due mainly to our relatively small population and geographical isolation from the rest of the world - though the significance of this is decreasing.

The number of locally written viruses is fairly small, but there are several groups of 'native' virus writers, including a group called VLAD (Virus Laboratories And Distribution): they have recently released a hypertext electronic magazine, also called *VLAD*, which can be found on various BBSs and on the *Internet*. It contains source code for new viruses, virus-related information on DOS/BIOS operating systems, and programming hints, with the usual news and gossip. Excerpts from the first issue give its flavour:

- Hello, and welcome to the 1st issue of the VLAD magazine, hopefully a number one Australian export :)
- …Just remember that spreading viruses is illegal and you shouldn't do it… Especially with Daddy which shouldn't be copied into an archive and uploaded to a lamer bbs because that would be wrong wouldn't it! :)

The Victorian state police have kept a low profile since a disastrous attempt to convict a student alleged to have distributed viruses. We understand that the national force, the Commonwealth Police, are only interested in the credit card and phone fraud they associate with these groups.

In the Press

The Australian press sometimes strikes something original: a recent issue of a local PC magazine published a letter signed 'Dark Fiber' (*AIH*), which included these words: 'Stephen W Hawking defends our actions: in his eyes we create artificial life forms. And that comes from a highly regarded scientist'. Anyone want to argue with Stephen Hawking?

There was excitement in February, when (according to the papers) the *Australian Tax Office's* entire computer network was shut down because of an outbreak of No_Frills. This is an old and boring virus, written several years ago by Clinton Haines, then a student at a Brisbane high school.

The affair inspired a local paper to publish a surprisingly good article on viruses, which they illustrated with a photo of Haines, with the heading, 'Writer will not quit'.

Now a science student at the *University of Queensland*, Haines claimed to be working on new and better viruses, but keeping them under wraps; adding to his collection with the aim of writing anti-virus programs. In his interview, he lamented that his notoriety may have killed his chances of getting a job in the computer industry. How unfair!

Distribution

Not infrequently, technology seems to conspire with viruses to allow them to take advantage of any momentary lapse; perhaps the fastest spreading outbreak on record occurred in a local university with a number of campuses. The university had a network of interconnected servers, with special software which distributed new software to all the servers in the network.

The 'nightmare possibility' happened: a supervisor logged in to the master server from an infected PC, and infected LOGIN.EXE. The updating software noted that the file had been updated, and distributed it to all the other servers. Within the hour, over 3,000 workstations on about five campuses were infected!

Another case involved a PC infected with about twenty different viruses, all breeding feverishly and filling the hard disk space with multiply-infected files. It is rare to find three variants of the Perfume virus running free on the same computer: the unfortunate victim had inherited the PC from an ex-employee who had been responsible for evaluating and selecting the firm's anti-virus protection, but had not removed the collection when he finished his tests.

On a related theme, we have been receiving for some months now batches of files infected with exotic viruses from different parts of Australia and New Zealand. All the files have been subsets of a particular collection of 71 files

VIRUS BULLETIN **©1995 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)1235 555139. /95/$0.00+2.50**

containing four boot sector droppers, uncommon specimens like Flash, Fumble, Kennedy and Emmie, and a corrupted and harmless program. Where did they originate?

It transpired that local agents for one anti-virus product had been supplying potential customers with a disk of selected viruses, and using unlicensed and outdated copies of competitors' products to demonstrate the superiority of their own software.

When one of their salesmen was questioned about the ethics of his actions he replied: 'I'm into selling; not ethics!' Regrettably, libel laws preclude us from naming the firm.

We deal with frequent reports of 'boring' old viruses like Junkie, Monkey, Anti-CMOS, W-Boot (a local variant of Stoned), Form, Anti-EXE, Lemming and Tai-Pan. In recent weeks, BUPT9146 and Boot-437 have been 'climbing the charts'. Variety is provided by isolated outbreaks of new or rare viruses, often only reported from a single firm or department, and presumably caused by careless or malicious release of material downloaded from a suspect BBS.

Recent one-off outbreaks in Melbourne were caused by Da'Boys (DOS boot sector virus), Shin (boot sector virus probably written at a local university), CHAOS.1241 (file infector with a destructive payload) and Fat_Avenger.

Cultural Influences

In the last fifty years, our society has been completely transformed. Once a British colony, in which even third generation Australians would speak of 'going home' when they visited England, we are now truly multi-cultural.

Many of our new citizens visit their former homes, often bringing back the latest local software with the latest local viruses. Recently, there was a significant outbreak of the Fairz (or Khobar) virus within the Egyptian community in Melbourne, after a copy had been brought from Cairo, where it was common a year ago.

Anticad was thought to have been introduced by a visiting group of Chinese academics, and a number of other Asian viruses are known to have been introduced by visitors.

Other Sources

Most of our computer hardware is imported from the South Asian countries: many major virus outbreaks have been due to the mass distribution of accompanying infected software. The delays in delivery are short, so we frequently get these viruses before they are detected in Europe or the USA.

Regrettably, many PCs are delivered with viruses pre-installed, and a substantial percentage of disks accompanying new hardware is still infected. Australia recently received a large shipment of preformatted hard disks, all infected with the Sampo virus, and there have been many cases of large shipments of infected preformatted (and even nominally blank) floppy disks.

Our schools still act as major virus distributors, with infected games the most common vehicle. Doom.II.Death is spreading happily, thanks to infected copies of DOOM cheat files on CDs imported from the United States and uploaded to various BBSs. The employee with children at school, who takes disks home to work on the family PC, is still the most common single source of viruses in the workplace.

Shareware of all sorts is a wonderful source of useful (and sometimes useless) software, but often it comes with surprising extras. We had a call reporting Barrotes.1310 on a file downloaded from a BBS while we were writing this article - the first report we have had of it in the wild.

Recovery

Our Data Recovery Service gets a number of requests to restore data on computers with IDE hard disks. Most have been hit by harmless viruses, but one had been 'optimised' with a popular utility. In all but one case, we were able to recover all data successfully. In the exception, the owner had had a go with a utility, and mangled all copies of the partition information. However, we were able to reconfigure the drive as a normal IDE drive, and access the first 520MB, which contained almost all the files the customer needed.

"many of our new citizens visit their former homes, often bringing back the latest local software and the latest local viruses"

Few users realise the dangers inherent in these drives, and even fewer dealers offer any alternative to the traditional 'You'll have to reformat it and start again.'

Our support staff lists one of the most frustrating aspects of the job (after false alarms) as listening to the stories of users who, when their PCs have caught a trivial virus, have reformatted their disks, and then rung to report the virus. They could have fixed everything in five minutes, with no loss of data, had they rung us first.

Other highlights include tales of the user who insisted there was a virus 'that vibrated the hard disk heads in such a way that the PC walked sideways till the PC fell off the bench'; the user confused by the name VET, who ended a conversation with one of our staff: 'by the way, I've got a problem with the dog'; and the lady who asked, 'I've read about the new virus - should I bring the boys in to be inoculated?'

Generally speaking, the Australian scene doesn't look much different from the rest of the world. However, we have been lucky: we have not had disasters such as those caused by Tremor and EXE_Bug, and Natas is almost exclusively reported by our overseas customers. So, it's not all bad down on this side of the equator.

FEATURE 2

Enhancing your Chances

Jonathan Burchell

With the introduction of *Windows v3.1*, *Microsoft* introduced a new way of accessing the local hard disk, called FastDisk, or 32-bit disk access. This was billed as one of the many speed improvements of *Windows v3.1*, and although in that version it is disabled by default, when *Windows for Workgroups* (*WFWG*) and *Windows v3.11* shipped, *Microsoft* felt sufficiently confident in the stability of FastDisk to enable it by default. As a result, any user of *WFWG* or *Windows v3.11* with compatible hardware is probably benefiting from this feature without even knowing it.

FastDisk brings many improvements to the speed of disk access under *Windows*. Machines which are relatively slow in terms of disk access can suddenly spring to life when FastDisk is enabled; however, one little-documented and little-discussed fact is that the use of 32-bit access almost certainly blocks or severely disables the action of most (if not all) TSR virus detectors and behaviour blockers.

Consequently, many companies believe that because they have such utilities loaded they have adequate workstation protection. In fact, they have left themselves far more open to a viral attack than they realise.

Disk Access Under DOS

In order to understand why this is the case, we must first reacquaint ourselves with exactly what happens when an application wants to access the disk drive under DOS.

Firstly, the application issues a call which accesses the disk (File Open, Read, Write etc) via the normal DOS Int 21h interface. This call is then processed by DOS, which determines where on the disk the requested information exists. *MS-DOS* cannot talk directly to the disk controller hardware; however, disk controllers come with a standard BIOS interface via Int 13h. This interface is often referred to as the 'Disk BIOS'.

Finally, the code called by Int 13h talks directly to the hard drive controller: it is this code which knows how to access the disk controller physically, and to read or write the specified sectors. Each drive controller in a system requires a specific Disk BIOS.

Although this sequence may seem complicated, the divorce of DOS from the physical disk driver has enabled DOS machines to support many different types of disk drive (ST-506, SCSI, ESDI, IDE, Optical, CD-ROM etc).

The interface to Int 13h is completely standardised. It is relatively simple (little more than read/write a disk sector), and well understood. As long as your new disk controller for 'the super-duper terabyte magneto-optical drive' can provide an Int 13h interface, DOS will automatically be able to read and write files to it.

For many standard disk controllers, the Disk BIOS is part of the PC's motherboard BIOS. Other disk controllers (such as SCSI devices) will provide an Int 13h BIOS on the controller card, which is configured into the system by the System BIOS on start-up.

Finally, some Int 13h BIOS drivers exist as software drivers loaded from CONFIG.SYS. Just such a device driver is the CD-ROM support module MSCDEX. These drivers watch Int 13h: when they see a call for the device they handle, they intercept and process it, rather than passing it to the standard ROM handler.

Disk Access Installation in Windows

When *Windows* is running in standard mode, disk access proceeds along the lines described above and is completely identical in operation to disk access from DOS.

When *Windows* is running in enhanced or protected mode (probably the norm these days, as the only real requirement is a 386 or above computer), *Windows* itself, and some of its applications, will also be running in protected or 386 mode. Protected mode is a special mode of the 386 (and later), solving many problems to do with memory access, process interaction and multi-tasking. However, it is not possible to access real mode code directly from protected mode.

Two prime examples of real mode code are DOS itself and the BIOS. In order to access either of these components, *Windows* must switch the processor from protected mode to real mode (a special mode called virtual 8086 is used) to make the call and then back into protected mode when the call has completed. The process is illustrated in Figure 1.

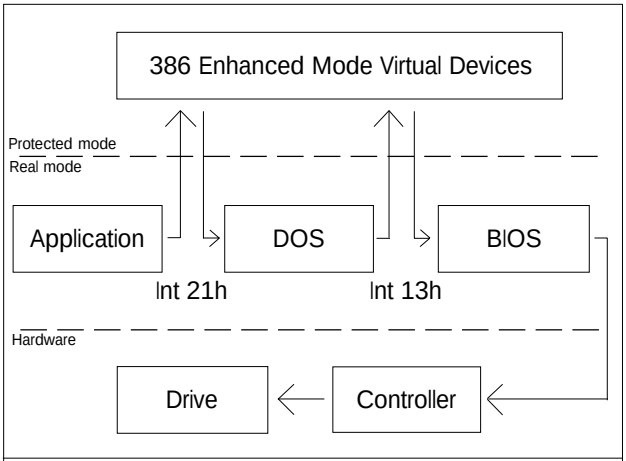


Figure 1: Disk access in 386 mode *Windows* without FastDisk.

First, an application makes a call to read from a file, then *Windows* traps this interrupt and switches to protected mode, where several virtual device drivers (such as the 32-bit network drivers in *Windows v3.11*) check the call to see if they want to handle it directly.

If none do, *Windows* switches back to virtual 8086 mode and calls DOS via Int 21h. As before, DOS works out the location on disk of the requested information, and generates a series of Int 13h calls to access the disk. Again, *Windows* traps the call and switches to protected mode (it is a requirement of being in protected mode that all interrupt calls cause a transfer from the virtual 8086 machine back to the protected mode handler - hence the name protected mode).

After some processing, *Windows* switches back to virtual 8086 mode and passes the call to the Disk BIOS via Int 13h. The BIOS talks to the disk controller and starts the physical read. When it has completed, it returns, forcing another transition to protected mode.

The *Windows* kernel does some more processing before returning to virtual 8086 mode, so that *MS-DOS*, which generated the original Int 13h call, can see the return and complete its processing. When the Int 21h call completes, a final virtual 8086 mode to protected mode transition is performed, and the *Windows* kernel returns to the application which started the whole thing off.

Even if the above explanation is a little too technical for you, two points should by now have been made clear: first, essentially the same thing is happening in standard mode *Windows* and at the DOS prompt. A file I/O request accesses DOS via Int 21h, which then accesses the Disk BIOS via Int 13h. Second, there is a lot of switching between real and protected modes.

Switching between real and protected modes is time-consuming, and is also a major obstacle to improved performance. Additionally, neither DOS nor the Disk BIOS are re-entrant, which means that only one task at a time may use them. This lack of re-entrancy inhibits the ability to multi-task properly, despite being in enhanced mode, and is also the reason why systems may be unable to run multiple DOS applications despite having apparently oodles of free virtual memory.

FastDisk

In an attempt to overcome these limitations, *Microsoft* introduced the FastDisk, or 32-bit disk access system, which, put quite simply, is an Int 13h protected mode replacement for the disk BIOS. This driver provides a 'standard' Int 13h interface for *MS-DOS*, but is located within the *Windows* executable.

Figure 2 illustrates the call flow for disk access in enhanced mode when 32-bit access is enabled. To begin with, it proceeds exactly as described above. The application makes a file I/O request; the *Windows* kernel looks to see if any of the protected mode virtual device drivers want to handle it. If they do not, it switches back to virtual 8086 mode and passes the call to DOS via Int 21h.

DOS processes the call and eventually makes disk access calls via Int 13h. The change happens here: instead of passing the call back to the previous disk BIOS, via Int 13h, *Windows* passes the call, in protected mode, to the FastDisk driver, which talks directly to the disk controller hardware.

Performance is improved because several transitions to and from protected mode are saved. The FastDisk driver itself is highly optimised for the task at hand (Disk BIOSes may not be, as they have often been written to cope with multiple hardware configurations and operating environments - this can lead to unwieldy and slow code). In addition, the FastDisk driver is re-entrant, so multi-tasking is improved.

One of the first things people may notice when 32-bit access is enabled is the ability to run more DOS applications: this is because they can now be safely given virtual memory by the *Windows* kernel without fear that a page-swap request will occur whilst the task itself is accessing the disk, and cause the machine to hang.

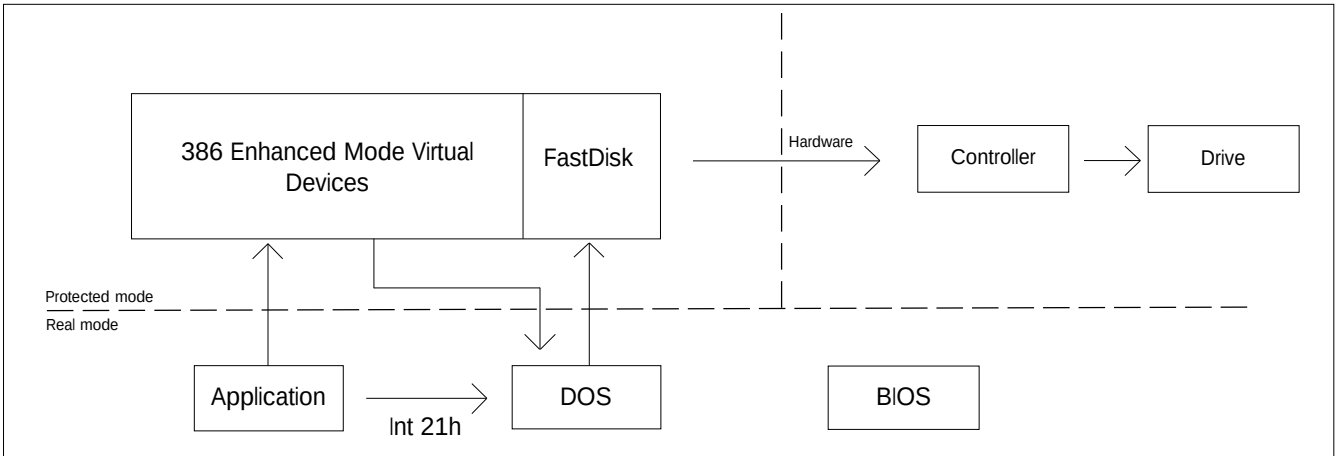


Figure 2: Call flow in 386 mode *Windows* with FastDisk.

Enabling/Disabling FastDisk and Component Parts

32-bit disk access is enabled and disabled via a check-box or the virtual memory settings of the 386 Enhanced dialog. This is accessed via the control panel. The check-box will only be present if your system supports 32-bit access. As *Windows* must go directly to the controller hardware for 32-bit access, currently only a subset of disk controllers (those which are *Western Digital*-compatible) is supported. The actual 32-bit access system consists of the following enhanced mode virtual device drivers:

- WDCtrl - this is the FastDisk device which talks to standard *Western Digital* 1003 or ST506 hard drive controllers (circa 90% of existing disk controllers). This device is only installed if *Windows* Setup detects a compatible hard drive controller
- BlockDev - this co-ordinates devices calling block I/O services and the FastDisk devices which provide those services for specific hard drive controllers. This device is always installed
- PageFile - this handles the virtual memory paging file. It calls through BlockDev if any FastDisk devices are available. This device is always installed. Int 13h, which traps and emulates Int 13h BIOS calls by calling BlockDev. This device is only installed when at least one FastDisk device is present

Because WDCtrl is the only FastDisk device included with *Windows v3.1*, the standard components only support 32-bit disk access on *Western Digital*-compatible controllers. When you run Setup, it automatically detects these controllers and, if present, adds the following lines to the [386enh] section of SYSTEM.INI:

```
32BitDiskAccess=ON
device=*int13
device=*wdctrl
```

No actual files relating to WDCtrl or Int13h can be found, because the code is built into WIN386.EXE. This is signified by the leading '*' in the device name. 32-bit access can be turned off simply by changing the above option: it is not necessary to remove the 'device=' lines. Note that for *Windows v3.1* the default is off, whereas for *WFWG* and *Windows v3.11* the default is on.

Microsoft only ships 32-bit drivers for *Western Digital* compatible systems; however, many third-party drive manufacturers also supply 32-bit *Windows* drivers for their own systems.

FastDisk, Viral Monitors and Viruses

To summarize, the major benefit of 32-bit disk access is greatly improved disk performance and multi-tasking. This is achieved by replacing the standard BIOS Int 13h interface with a driver located within *Windows* itself. It is this replacement of Int 13h which represents an area of serious concern for virus monitoring.

Many viruses use Int 13h to access the disk directly, thus avoiding behaviour monitors which are watching at the DOS level (Int 21h) for suspicious activity. To counter this, many behaviour monitors also watch Int 13h for suspicious activities; however, when *Windows* is started in enhanced mode with 32-bit access enabled, such monitoring becomes impossible, because the behaviour monitor will never 'see' Int 13h. Instead, *Windows* will pass it to the FastDisk driver, which certainly has no virus detection built in.

As a result, such systems are considerably weakened. A virus which performs disk I/O via Interrupts 21h or 13h will definitely 'work' if executed within *Windows*. If it avoids any activity monitors on Int 21h (e.g. performing disk I/O via Int 13h), it will not be detected.

If the virus has an infection mechanism which relies on Int 13h, it will not do much more, because it will see no further Int 13h calls. However, it will still have been executed, have infected whatever it wants on first execution, be resident, and (possibly) have released its payload.

On the other hand, if it has an infection mechanism which monitors Int 21h, it will continue to see disk I/O requests, and will be free to cause further infections. Stealthing via Int 21h will also be successful, and may prevent the virus from being recognised via scanners and casual observation.

BIOS Blockers

Many BIOSes have a feature which inhibits writes to the master and partition boot sectors, enabling them to provide some level of virus protection.

The whole point of FastDisk is to prevent disk access ever passing through the BIOS: if the only form of protection on a system is at the BIOS level, problems will arise. This form of BIOS protection is completely unable to prevent writes to these 'critical' sectors when *Windows* is in 32-bit disk access mode.

The irony is that many manufacturers use the anti-virus capabilities of the BIOS (which are very powerful under normal circumstances) as a selling point. They then ship the machines with *Windows v3.11* or *WFWG*, which has 32-bit disk access enabled by default (or even a custom installation of *Windows v3.1* with it enabled), thus effectively removing the protection.

What to Do?

Firstly, and most obviously, if you are relying on the functions of your computer's BIOS to provide protection against boot sector viruses, you should disable 32-bit disk access. You will suffer a degradation in performance, but at least your protection will remain intact.

If you are using some form of TSR-based detection system or behaviour blocker, you should test it with 32-bit disk access enabled. It may well be that you are not as well-protected as you had thought.

PRODUCT REVIEW

S&S: The Anti-Virus Toolkit

Dr Keith Jackson

There is a saying that life goes round full circle. This article is proof of that theory: *Dr. Solomon's Anti-Virus Toolkit* was the first product ever reviewed by *VB* (July 1989). An update review was published in June 1991, with the next review in November 1992, on release of the first *Windows*-specific version. And here I am, reviewing this product for the fourth time, in its newest release (version 7) - doesn't time fly when you're having fun!

The *AVTK* operates under DOS, *Windows* or *OS/2*. As I run neither *OS/2* nor a network, this review must unfortunately ignore the features solely relevant to those platforms.

Dr. Solomon's AVTK has been written about *ad infinitum*, and always receives consistently good reviews: it came out top in *VB's* most recent comparative DOS scanner review (January 1995). I do not intend to go over old ground, so I will concentrate on features new to version 7: a revamped graphical user interface, single button repair, archive file and compressed file support, changes to memory-resident program and scanner, and international language support.

The single button repair feature is described in the manual, but the README file included with my copy explained that this does not yet work with infected Master Boot Sectors and/or partition boot sectors.

Documentation

The package includes a manual covering both DOS and *Windows* versions, and a copy of the latest version of the *Virus Encyclopædia*. The manual is a 222-page, A5 book, which seems to have been pruned down over the years. Although I do not have the old documentation to hand, I get the impression that superfluous information has been removed and/or abbreviated. In any case, it is easy to comprehend, reasonably well-indexed, and appears to be a balanced effort, though I would recommend adding a detailed explanation of all possible errors which may occur.

There are a large number of command-line switches in the FindVirus scanner: 37 are explained in the manual, but 52 others were undocumented. Even if it is necessary to mark some 'Internal use only', all these options should at least be mentioned: this would be preferable to the present conspicuous lack of information.

The *Virus Encyclopædia* provides a short description of each known virus, explanations of what each can do, and instructions on eradicating viruses. It contains a paragraph of relevant information about many of the viruses known to the *AVTK*, which now claims to scan for 5816 viruses.

The on-line version of the *Virus Encyclopædia* contains information about 2957 uniquely-named viruses. However, the printed copy has information about only 314 unique viruses, coupled with mention of some 800 variants on this core set. The virus deluge is having an effect! Previous versions had information on most viruses known to it: the *Virus Encyclopædia* seems to be finding it hard to keep up.

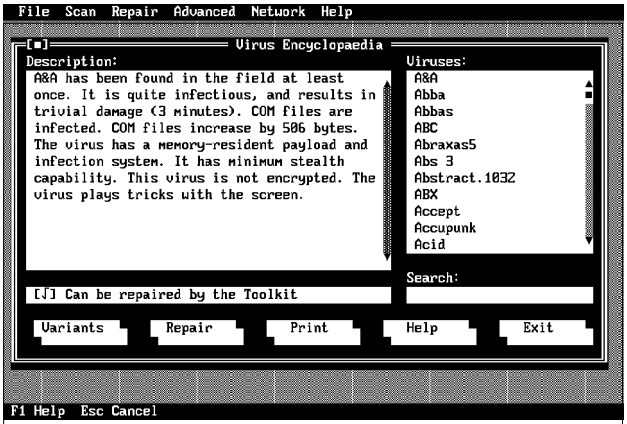
Installation

The *AVTK* is provided only on three 3.5-inch, 1.44 Mbyte floppy disks. It was formerly available on various types of diskette, but anything other than the default is now sent only on special request. Installation can be carried out for DOS, for *Windows* (including all DOS components), or for *OS/2*.

Installing the *AVTK* under *Windows* proved no more difficult than executing the SETUP program and answering a few questions. The installation program requires knowledge of the drive where it is to be installed, followed by the subdirectory location. Why the drive and the name of the desired subdirectory have to be entered separately is beyond me. Why not just ask for a path name to be entered? This is a more standard technique.

At the appropriate time, the installation program requests that the other floppy disks are inserted. After file copying is complete, it asks whether the memory-resident program (VirusGuard) should be installed, and whether changes to AUTOEXEC.BAT and CONFIG.SYS are permitted.

VirusGuard can be set up to offer maximum security (all files will be checked during copying and/or writing), standard security (executable files will be checked whilst being copied), or minimum security (only files on floppy disk are scanned). No matter which mode is specified, VirusGuard scans all executables before they are run, and



The *AVTK* gives information on many infectors in its online *Virus Encyclopædia*, and is far more comprehensive than the printed version.

the boot sector of all diskettes. The security level can be changed by altering one line in the AUTOEXEC.BAT file and rebooting the PC.

During the final stages of installation, a message appears stating that the necessary *Windows* icons will be created the next time *Windows* is restarted. Why not just create them immediately? The installation program then scans the local hard disk drives to make sure they are virus-free, and explains how to create a 'Rescue Disk' (you must do this manually). Installation is then complete.

The *Toolkit* installation process has always been simple to use, and it still is. It is difficult to find any real fault with it. I do, however, have two small gripes, both of which concern installation under *Windows*.

Creation of the *Toolkit* icons only works if the files associated with *Microsoft Windows* are on the DOS PATH: on my PC this is usually not true. This problem only shows itself as a short message which states, without explanation, that the *Windows* icons could not be created: it took detective work to find out exactly what was happening.

Another quibble is that the files MAKEICON.EXE and MAKEICON.INI are left behind in the *Windows* subdirectory by the installation program. Why doesn't it clear up after itself? There is enough rubbish in my *Windows* subdirectory without adding to it.

When fully installed (DOS and *Windows*), the *AVTK* needed 3.44 MB of hard disk space. I have said before, and shall say it again: 'spending' this much hard disk space on what is at heart merely a utility is faintly ridiculous. It is obvious that bells and whistles count for a lot in today's market.

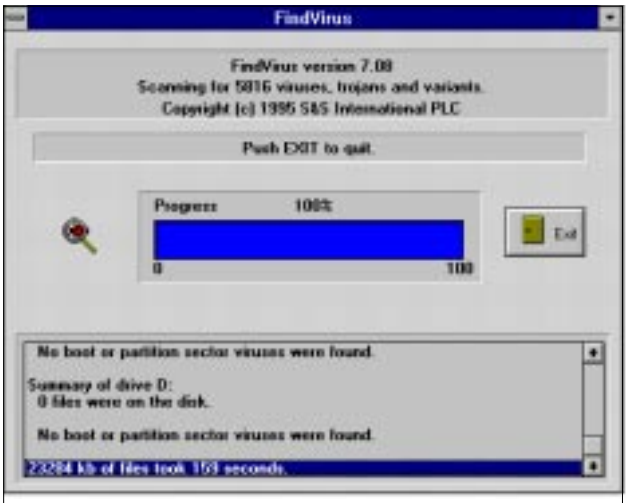
Scanning

The *AVTK* for DOS checked the hard disk of my test PC in 1 minute 34 seconds - 679 files in total (24.9 Mbytes), of which 331 were actually checked. Under the *Windows* version, scan time increased to 2 minutes 14 seconds.

In comparison, when executing under DOS and performing the same scan, *Sophos' SWEEP* took 2 minutes 40 seconds in quick mode, and 8 minutes 49 seconds in full mode. *McAfee's SCAN* took 1 minute 49 seconds to perform the same task. These times are excellent, and will maintain the product's reputation for being swift as well as secure.

In common with several other scanners, scan time reported onscreen by the *AVTK* is less than the actual time taken to perform the scan - for example, the time taken by the *AVTK* to scan my hard disk under DOS was recorded by a stopwatch at 2 minutes 14 seconds; 43% more than the onscreen time reported.

Onscreen time refers to the file scan time, which did not commence until 48 seconds had elapsed. In similar manner, elapsed *Windows* scan time was 3 minutes 8 seconds (40% more than stated onscreen time).



Version 7 of the *Anti-Virus Toolkit* includes FindVirus version 7.08, and has the ability to look inside archived and dynamically compressed files.

One clever feature of the scanner is that it will not check inside files less than 20 bytes in length, stating that 'File too small to have known virus'. This may well cut down scan times on some systems.

Accuracy

When pitted against the test-set described in the *Technical Details* section below, FindVirus detected 100% of the 248 virus infected test samples. Not bad by anybody's standards.

When tested against the 500 positively replicating Mutation Engine (MtE) samples, all but two were correctly detected as infected. This detection rate of 99.6% is eminently acceptable; however, I have no idea why just these two MtE samples in particular are not detected.

Activating the option to scan inside compressed files to look inside a ZIP file gave the same results; the *AVTK* correctly detects all bar two MtE samples. This test did, however, have a drastic effect on the scan time for DOS, which rose to 6 minutes 15 seconds (an increase of almost 400%).

The *AVTK* will only scan inside ZIP and ARJ archive files, but these are the most often-used PC compression programs. For some unknown reason, this option is only available if the PC contains a 386 (or higher) CPU. I wonder why? The documentation states the point, but does not explain it.

Scanning inside dynamically compressed files (PKLITE, LZEXE) had a similar effect on DOS scan time, which rose to 6 minutes 57 seconds.

There is a bug in the way the ZIP/ARJ and PKLITE/LZEXE buttons are handled in the *Windows* version of the *Toolkit*. These options can be correctly selected using a mouse, but if the down-arrow key on the keyboard is used, the options are bypassed, and cannot be directly selected. Neither the DOS user interface, nor version 6 of the *AVTK*, exhibits this bug. It is definitely new, but a minor problem, and rectifiable.

Memory-resident Program

The memory-resident component of the *Anti-Virus Toolkit*, VirusGuard, occupies only 9 Kbytes of low memory when installed; parsimonious by any standards. It detects fewer viruses than the stand-alone scanner, but of the test samples listed in the *Technical Details*, it failed, on copying, to detect only seven: Butterfly, WinVir14, Coffeeshop, NukeHard, V2P6, Tremor and Satanbug. These viruses are mainly polymorphic and/or encrypted.

No matter which level of security is chosen (see installation above), none of the MtE-infected files are detected whilst being copied from one drive to another. The manual states that detecting the more polymorphic viruses with VirusGuard is only possible after the file has been run, which accounts for the behaviour described.

The overhead introduced by VirusGuard was measured by timing how long it took to copy 40 files (1.25 Mbytes) from one subdirectory to another. Without VirusGuard, the files could be copied in 23 seconds: this rose to 26 seconds with minimum security, 58 seconds with standard security, and 1 minute 45 seconds under maximum security. Only when maximum security was used did the PC 'feel' slow - I would guess that this option is only really usable on a very fast PC.

Conclusions

Functions are also available within the *AVTK* for detecting changes in file checksums, repairing infected files etc, but as promised at the start of this review, I have concentrated on the features which have changed in this new version.

This product has a long-standing reputation for high detection rates, and performs consistently well in DOS scanner comparative reviews. It remains excellent at detection, and extra facilities for inspecting within compressed and archive files add to its impressive armoury.

Although VirusGuard introduces a large overhead when set up in its maximum security mode, a balance can readily be struck between security and intrusiveness. At first glance, the results reported for VirusGuard may not look very good, but it performs much better than many of its competitors.

Whilst the measured scan times are much increased when inspecting compressed and/or archived files, this is not particularly surprising. Such features are used mainly to scan incoming files, not to perform routine scanning, and so this should not matter overmuch. Rejecting this product on the grounds of the speed at which it scans compressed files would be foolhardy.

I still harbour a lingering doubt over *Windows* versions of anti-virus software - how is it possible to boot and execute *Windows* solely from a floppy disk when the hard disk has become thoroughly infected? Providing both a DOS and a *Windows* version of the same product does mitigate this objection, and even unreformed Luddites such as myself admit that the *Windows* version of the *AVTK* is easy to use.

Version 7 of *Dr. Solomon's AVTK* gets a hearty recommendation from me.

Technical Details

Product: *Dr.Solomon's AVTK*.

Developer/Vendor: *S&S International plc*, Alton House, Gatehouse Way, Aylesbury, Buckinghamshire, HP19 3XU, UK, Tel +44 (1296) 318700, fax +44 (1296) 318777, Email: sales@sands.co.uk.

Availability: Any *IBM* or compatible PC, with DOS v3.1 or higher. 330 KB of RAM is used during program execution. The DOS version requires 2.5 MB of hard disk space; the *Windows* version requires 4 MB.

Version evaluated: *AVTK v7.00*, FindVirus v7.08.

Serial number: None visible.

Price: Single user licence, £55 per PC. Site licences: 11-100 users, £55/PC; 101-250 users, £42.50/PC; 251-500 users, £34/PC. For larger site licences, apply to *S&S*. All software comes with free quarterly upgrades for one year.

Hardware used: A Toshiba 3100SX laptop PC (16MHz 386) with one 3.5-inch (1.4 Mbyte) floppy disk drive, 5 MB of RAM, and a 40 MB hard disk, running under *MS-DOS v5.00*.

Viruses used for testing purposes: This suite of 158 unique viruses (according to the virus-naming convention employed by *VB*), spread across 247 individual virus samples, is the current standard test-set. A specific test is also made against 500 viruses generated by the Mutation Engine (which are particularly difficult to detect with certainty).

The test-set contains nine boot sector viruses (Brain, Form, Italian, Michelangelo, Monkey, New_Zealand_2, Quox, Spanish_Telecom, V-Sign), and 239 samples of 150 parasitic viruses (Spanish_Telecom appears in both lists). There is more than one example of many viruses, ranging up to 12 variants of the Tiny virus. The parasitic viruses used for testing are listed below. Where more than one variant virus is available, the number of examples of each virus is shown in brackets. For a complete explanation of each virus, and the nomenclature used, please refer to the list of PC viruses published regularly in *VB*:

1049, 1260, 12_TRICKS, 1575, 1600, 2100 (2), 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 777, 800, 8888, 8 TUNES, 905, 948, AIDS, AIDS_II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), AntiCAD (2), Anti-Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Butterfly, Captain_Trips (2), Cascade (2), Casper, Coffeeshop, Dark_Avenger, Darth_Vader (3), Datacrime, Datacrime_II (2), Datalock (2), December_24th, Destructor, Diamond (2), Dir, Diskjeb, DOShunter, Dot_Killer, Durban, Eddie, Eddie_2, Fellowship, Fish_1100, Fish 6 (2), Flash, Flip (2), Fu-Manchu (2), Halley, Hallöchen, Helloween (2), Hide_Nowt, Hymn (2), Icelandic (3), Internal, Invisible_Man (2), Itavir, Jerusalem (2), Jocker, Jo-Jo, July_13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (5), LoveChild, Lozinsky, Macho (2), Maltese_Amoeba, MIX1 (2), MLTI, Monxla, Murphy (2), Necropolis, Nina, Nomenklatura (2), Nuke_Hard, Number_of_the_Beast (5), Oropax, Parity, PcVrsDs(2), Perfume, Pitch, Piter, Polish_217, Power_Pump, Pretoria, Prudents, Rat, Satan_Bug (2), Shake, Sibel_Sheep (2), Slow, Spanish_Telecom (2), Spanz, Starship (2), Subliminal, Sunday (2), Suomi, Suriv_1.01, Suriv_2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Syslock, Taiwan (2), Tequila, Terror, Tiny (12), Todor, Traceback (2), Tremor, TUQ, Turbo_488, Typo, V2P6, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virdem, Virus-101 (2), Virus-90, Voronezh (2), VP, V-Sign, V-1, W13 (2), Willow, WinVirus_14, Whale, Yankee (7), Zero_Bug.

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, Oxfordshire, OX14 3YS, England

Tel 01235 555139, International Tel. +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email virusbtn@vax.ox.ac.uk
CompuServe 100070,1340@compuserve.com

US subscriptions only:

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

Tel +1 203 431 8720, fax +1 203 431 8165

END NOTES AND NEWS

The 5th Annual Network Security in the Open Environment (*NetSec 95*) conference will be held from 12-14 June 1995 in New Orleans, Louisiana, USA. Tutorials, vendor exhibits, and seminars will be included. Further information is available from the Computer Security Institute on Tel +1 415 905 2626, fax +1 415 905 2218.

The next round of **anti-virus workshops from *Sophos Plc*** will be held on 24/25 May, at the training suite in Abingdon. Day one is an introduction to computer viruses; day two, an advanced virus workshop. One session costs £325.00; both, £595.00. Contact Karen Richardson on Tel +44 1235 559933 for details.

Norman Data Defense Systems has announced the launch of the ***Norman Automatic Virus Analysis System***. This service, which can be accessed at ftp.norman.com or via BBS (+1 703 573 8990), will be free of charge, and is intended to analyse files suspected by users to be virus-infected.

March is Michelangelo month, and (as shown in the Virus Prevalence Table on p.3), infections have occurred this year. Although we have had only one report of it here in the UK, Germany and the USA both reported incidents.

The *British Standard Code of Practice for Information Security Management* has now been released by the *British Standards Institution*, and deals with a variety of topics, including **equipment security, user responsibilities, and protection from malicious software** (i.e. viruses and related programs). Information can be obtained from Nick Clark (BSi press office): Tel +44 181 996 9000, fax +44 181 996 7400.

Yet another virus has been released; this time at the *Computer Shopper Show* in Birmingham, UK (which took place from 16-19 May 1995). In this case, a CD-ROM called ***The Gates of the Underworld*** **was infected by two viruses, Tai-Pan and Goldbug**. The company which produced the CD, Home Grown Productions Ltd, had prepared it by downloading shareware from various BBSs. The infected software has now been withdrawn from sale.

GEC-Marconi and the *Merseyside Police Fraud Squad* have joined forces to present a conference on computer crime: ***Computer Crime - Secure IT***. It will be held on 18 May 1995 in Liverpool, UK. Further information is available from the conference organiser, Lynda Moore, on Tel +44 151 231 3440, fax +44 151 707 0423.

The fifth annual *Virus Bulletin* conference, ***VB 95*, will be held at the *Park Plaza Hotel* in Boston, Massachusetts**, from 20-22 September 1995. Internationally-renowned virus and security experts will address the problems of virus protection in the 1990s. For more information, contact Petra Duffield, Conference Manager, on +44 1235 555139.

***LAN/SEC 95 (Europe)* will be held in London, UK**, from 23-25 May, with optional workshops on 22 May. The conference is on network security, and is sponsored by *MIS Training Institute* in association with *Euromoney Publications*. Contact Mandy Moore, Tel +44 171 779 8795, fax +44 171 779 8944, for details.

***S&S International*'s next computer virus workshop** will be held at *Ashridge Management College* (Berkhamsted, Hertfordshire, UK) on 15/16 May. Cost for the two-day course is £680. Contact *S&S* on Tel +44 1296 318700, fax +44 1296 318777 for further information. The company has also opened a corporate US office in Burlington, Mass; Tel +1 617 273 7400, fax +1 617 273 7474.

A 'hacker' conference is scheduled to take place at *King's College*, London in July. Under the title *Access All Areas*, organisers have lined up such speakers as Emmanuel Goldstein (editor of *2600*) and ex-hacker Rop Gonggrijp.