ISSN 0956-9979 OCTOBER 1995

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Hacking around.** Our former editor, Richard Ford, put on a hacker's hat for a few days and visited a different kind of conference: read his report on p.9.

• **What's next, folks?** Much media hype has been made of *Microsoft*'s new operating system, *Windows 95*. We discover whether they noted our article on how viruses affect the system (June 1995) in a series of follow-up tests – see p.14.

• **Detecting a new way.** *IBM* has released a new version of its anti-virus software for *NetWare*: is it an improvement on their original? Turn to p.18 for our review.

CONTENTS

EDITORIAL

'…and it infects PDP11 executables as well…'

In recent weeks, an uncharacteristically high number of column inches in the media has been devoted to computer viruses. This time the culprit is WinWord.Concept, which is being presented to the public in a way that makes it appear to be the end of the Information Technology world as we know it.

"fear only comes from lack of understanding"

I am amazed by the seemingly total lack of comprehension with which daily newspapers, and even computing magazines, present the story. The devastatingly muddled stories provide ample evidence of the vast gap in understanding that technology is producing in the modern world. Let's have a quick look at some of what has been printed.

Both the *Sunday Times* (3 September 1995) and *Network Week* believe that the virus displays the message 'That's enough to prove my point': whilst the virus does *contain* this message, it is in the form of a comment, and is not displayed to the user.

In addition, *Network Week* states: 'After the initial infection, any other documents will be infected'. The virus only infects when the user selects 'Save As', so existing documents will be infected fairly slowly. Better yet, the article also says: 'The infected document on the CD [*Microsoft Windows 95 Compatibility Test* disk] is called Oemltr.doc and is believed to be the work of a professional Hungarian programmer'.

It's easy to see where this confusion originated: the code of WinWord.Concept is written in what is called Hungarian style, in which variable names are prefixed by a single character indicating their type (for example, a variable called sMessage would hold a string, whereas iCounter would hold an integer). This style (named after Charles Simonyi, a manager at *Microsoft* who is credited with creating the concept of code factories in the 1980s) is widely used by programmers working with *Microsoft* development tools, as *Microsoft* SDKs and DDKs use it throughout, and it is a convenient syntax to follow.

Representing the tabloid press, the *Daily Express* (29 August 1995): sandwiched between items about bare behinds on TV ads and couples on the M4, a piece entitled *Virus called Prank proves no joke for a computer*. The article contains the following statements: '…a new virus that could wreck computer equipment' (even given the risks associated with viral code, it's improbable that any hardware will be destroyed by WinWord.Concept); 'Programmes [*sic*] operating under *Windows 95*, *Windows 3.1*, or an *Apple Macintosh* can all become ridden with it' (programs are not infected, and 'ridden' is hardly a suitable word); 'Prank also spreads itself via the global Internet network by attaching itself to electronic mail' (it does not attach itself. A user may send another user a document which happens to be infected, but that is different); and finally: 'The infection hides itself in part of the computer programme known as a macro which opens the word processing file to the computer user. It overruns the macro and when the user opens the file, it is let loose into the entire computer, infecting any new documents.' No comment…

Whilst these errors may seem small and excessively technical, they all mount up. In addition, the tone of the stories suggests that information about the virus has been presented by parts of the anti-virus industry in a manner which the industry should definitely not be encouraging; rather working to prevent happening yet again. It must be ensured that technical matters are presented to the press in a manner which is both understandable and accurate.

Technology, even of the type that spawned WinWord.Concept, should not be feared. The fear only comes from lack of understanding, which in this case is aggravated by the mass media. I leave you with one more quote; *GQ* magazine interviewing Dr Alan Solomon, and talking about the computers he used in days gone by: 'Then, in quick succession, an *HP 3000*, a *VAX*, a *ZX-81*, a *Sinclair Spectrum* and – the real breakthrough – a *Lotus 1-2-3*'. Not so much a case of the journalist getting the wrong end of the stick; more getting the wrong end of a completely different stick.

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

NEWS

WinWord.Concept

The WinWord.Concept virus reported in last month's *Virus Bulletin* has been found on at least one CD-ROM. Shortly after the journal's September edition went to print, *VB* acquired a CD entitled *Snap-On Tools for the Windows NT Professional* from a UK company called *ServerWare*. The CD contains documents infected with WinWord.Concept.

It was shipped at the end of September to more than five and a half thousand *Windows NT* users. The infected documents on the CD-ROM are: custom~1.50\c1prod2.doc, html\netman.doc, intergra\intergra.doc, serverwr\ashwin.doc, serverwr\octopus.doc, serverwr\octposit.doc and serverwr\winport.doc.

ServerWare has since contacted all the customers who received infected disks informing them of the problem and of how to remove the virus, and has shipped a remastered CD, clearly distinguishable from the original: coloured bright blue, in comparison to the original's standard silver.

ServerWare is to be commended for its prompt and honest action in this matter, which will no doubt have gone a long way towards minimising the effect of this incident.

Meanwhile, *Microsoft* has finally admitted shipping the virus to OEMs on a *Windows 95* compatibility test CD-ROM. This has long been rumoured, but confirmation has been hard to obtain, due to problems in getting hold of the CD-ROM in question, *Windows 95 Software Compatibility Test Version 4.0*. It is not known precisely how many of these were shipped.

Various companies posted information on the Internet and provided fixes for the virus. As *VB* goes to print, we are aware of the following fixes:

```
Command Software
     ftp://ftp.commandcom.com/pub/fix/wvfix.zip
DataFellows
     http://www.datafellows.fi/macrovir.html
Datawatch
     http://www.zobkiw.datawatch.com/zob.html
FIRST http://www.first.org/first/resources/
      word.html
IBM  http://www.research.ibm.com/xw-D953-wconc
KAMI http://www.thenet.ch/metro/avpl/ww6macro
MS   http://www.microsoft.com/msoffice/
prank.htm
NCSA http://www.ncsa.com/wordvir1.html
Norman Data Defense
     http://www.norman.com/news.htm
Sophos Plc
     http://www.sophos.com/winwordvirus
S&S International PLC
     http://www.drsolomon.com/news/concept.htm
```

Companies with areas on *CompuServe* have also posted the information there. *VB* will, of course, keep abreast of the story, and will pass on information as it becomes available ▮

Virus Prevalence Table - August 1995

Virus	Incidents	(%) Reports
Parity_Boot	17	12.3%
Form	16	11.6%
AntiEXE	12	8.7%
Junkie	9	6.5%
AntiCMOS	8	5.8%
Jumper.B	8	5.8%
Monkey.B	8	5.8%
JackRipper	8	5.8%
Sampo	6	4.3%
WinWord.Concept	5	3.6%
Stoned.NoInt	4	2.9%
NYB	4	2.9%
BUPT	3	2.2%
Cascade.1701	3	2.2%
Stoned.Angelina	3	2.2%
Natas	2	1.5%
Telefonica	2	1.5%
V-Sign	2	1.5%
* Other	18	13.0%
Total	138	100%

* The Prevalence Table includes one report of each of the following viruses: Are-Three, Chameleon, Coffeeshop:MtE, Crazy_Boot, Flip, Friday_13th, HLLCa, Michelangelo, Monkey.Dropper, November_17th, One_Half, She_Has, Taipan.438, Taiwan, Tequila, UNashamed_naked, Wonka, Xeram.

Virus Awareness Campaign

The *National Computer Security Association* (*NCSA*), in conjunction with *CompuServe* and 16 anti-virus developers, sponsored an initiative in September to make the general public more aware of the threats posed by computer viruses. The campaign, which began on 8 September, lasted 20 days, and supported a *CompuServe* forum containing virus protection solutions provided by the participating vendors, and a toll-free help-line number for support.

Said Bob Bales, *NCSA's* Executive Director: 'We've learned a lot from doing this, which will all be applied to next year's Virus Awareness initiative. We've already started thinking about it, and plan to begin organisation very soon. We aim next year to make it a more participatory thing, and we'll be targeting big corporates as well as the end-user.'

Further information on the initiative, or on any other aspect of the *NCSA*, is available from Bales, or from Richard Ford (*NCSA's* Director of Research). Tel +1 717 258 1816 ▮

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 September 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

Assassin.4384

EN: An appending, encrypted, 4834-byte, fast, direct infector. It contains the message: 'FRöM HÉLL I CÆMÉ Tö TÆKÉ Æ LìFÉ ÆWÆY, Æ SöùLÉ I MùST CöLLÉCT YÖùR BLÖÖD MY WÆTÉR YÖùR FLÉSH MY BRÉÆD.. TöDÆY Yöù MÉÉT YöùR ÉND - ASSASSIN$'.

```
Assassin.4834       BD?? ??BB ??00 B9?? ??0E 1FD1 E981 37?? ??83 C302 E2F7
```

Bobas.754

ER: An appending, 754-byte virus containing the encrypted text: 'VIRUS :BOBAS v1.1'. The payload causes corruption of CMOS data, and displays a message intermittently in the centre of the screen.

```
Bobas.754           1E06 80FC FE75 0F81 FB52 5374 03E9 2801 071F 618B C3CF 80FC
```

CeCe.1998

CER: An appending, polymorphic, 1998-byte, new variant of the older 1994-byte virus, with a constant length decrypting procedure. It contains text beginning: 'Welcome to presentation of new program, named Ce-Ce!!!'. The patterns detect the viruses in memory.

```
CeCe.1998           B4CC CD21 3DCE CE75 13BF 0405 8BF7 B996 02F3 A60B C907 5E75
CeCe.1994           B4CC CD21 3DCE CE75 13BF 0205 8BF7 B995 02F3 A60B C907 5E75
```

Dei.1780

CER: An appending, encrypted, slightly polymorphic, 1870-byte virus with stealth capabilities. It contains the text: 'Devils & Evangels Inc. [DEI] MnemoniX v2.00\DEI.COM'. Virus structure in infected EXE and COM files differs: detection cannot rely on a simple template.

```
Dei.1780  (EXE)     BB?? ??B0 ??2E 3007 43F6 D881 FB?? ??76 F4
Dei.1780  (COM)     C706 ???? 2E30 B60E C706 ???? 0743 B??? ??E9
```

ExeHeader.453.A

ER: A group of 453-byte long viruses which insert their code in EXE file headers. They hook Int 13h and infect files when read, deleting all *.CHK files. Variant .A does not search subdirectories with names beginning with 'P'. The virus contains the text 'BOSCO'.

```
ExeHeader.453.A     B4F0 CD13 80FC 1974 108C D848 8ED8 2916 0300 2916 1200 E8D5
```

ExeHeader.453.B

ER: This variant searches through all subdirectories and deletes all *.CH? files. It contains the text: 'BOSCO D'SOUZA'.

```
ExeHeader.453.B     B4F0 CD13 80FC F974 138C D848 8ED8 2916 0300 2916 1200 E8DE
```

ExeHeader.453.C

ER: This variant searches through all subdirectories and deletes all *.CHK files. It contains the text: 'ROYDEN D'SOUZA'.

```
ExeHeader.453.C     B4F0 CD13 80FC F974 108C D848 8ED8 2916 1200 2916 0300 E8DB
```

ExeHeader.453.D

ER: Searches through all subdirectories, deleting all *.CHK files. It contains the text: 'BOSCO D'SOUZA'. It successfully replicates only on certain systems; when installing itself in memory it requires the string FA80 ???? ???? ???? ???? CD40 to be present in memory segment F000h.

```
ExeHeader.453.D     B4F0 CD13 80FC FA74 108C D848 8ED8 2916 1200 2916 0300 E81E
```

FTW.101

CN: An overwriting, 101-byte direct infector which infects one file at a time, hanging the system when all files in the current directory are infected. It contains the text: '*.COM FTW El Penga'.

```
FTW.101             B927 00BA 5201 CD21 720B E80B 0075 04B4 4FEB F3CD 20FA EBFE
```

FTW.192

CN: An overwriting, 192-byte direct infector which infects single files. It displays the message 'Copyright (c) 1992 Virtual Cortex', hanging the system when all files in the current directory are infected.

```
FTW.192             2172 0BE8 4F00 7504 B44F EBF3 CD20 FAB4 09BA 3501 CD21 EBFE
```

Girl_in_Green

CEN: An appending, encrypted, 1306-byte, direct infector. Its payload triggers on 3 June: the virus copies the boot sector from a floppy in drive A to the file BOOT.SEC, overwrites the boot sector with its code, and restarts the system. The message 'I ♥ YOU GIRL IN GREEN' is then displayed in an endless loop.

```
Girl_in_Green       0153 CD3B 551D 053C FC0E 07B9 1700 8DB6 E303 8BFE ACF6 D0AA E2FA C3
```

IVP.166

CN: A simple, appending, 166-byte direct infector, based on IVP. It contains the text: '*.c?m'.

```
IVP.166             B43F B903 008D 96A0 01CD 213E 80BE A001 E974 2F3E 8B86 C301
```

IVP.Skank.811

CEN: Appending, encrypted, 811-byte direct infector. Contains texts 'SkankySoso Man', '*.com *.exe'.

```
IVP.Skank.811       8D9C 1401 B905 032E 8A27 2E32 A42F 042E 8827 43E2 F2C3
```

MadSatan.647

CR: An appending, stealth, 647-byte virus containing the text 'Mad Satan', 'By [Mad Satan] v4.02'. When an infected file is run, the virus installs itself in memory without checking if it is already active, resulting in a system crash after a few infected files are executed. The seconds value of an infected file time stamp is 62.

```
MadSatan.647        50E8 0000 5D83 ED04 B8AD DECD 213D BBBB 7447 B821 35CD 212E
```

Mantis

CN: An appending, encrypted, direct infector targeting files with checksum information (deletes files *.MS). It contains the text '*.COM', 'COMMMAND.COM', '*.MS'. The 1215-byte variant contains the message 'Jesus saves all (except this lame computer!)'. After infecting three files, the virus plays a tune.

```
Mantis.1215         8B84 1000 EB02 ???? B94F 028D BC22 00EB 0031 0583 C702 E2F9
Mantis.1258         8B84 1000 EB02 ???? B964 028D BC22 00EB 0031 0583 C702 E2F9
```

MSD.331

CR: An appending, 331-byte virus. The text 'MSD93' is always found at the end of infected files.

```
MSD.331             3D00 4B74 0D80 FCFF 7403 E9BA 00B8 7707 9DCF 5053 5106 521E
```

November_17th.1061

CER: An appending, encrypted, 1061-byte virus targeting an anti-virus package. It contains the text: 'SCAN.CLEAN.COMEXE'.

```
November_17th.1061  9C00 0183 FB00 7410 FA8D BC25 01B9 0104 310D 311D 4743 E2F8
```

PS-MPC.331.B

EN: An appending, 331-byte, direct infector based on PS-MPC viruses but without an author's signature or message. All infected files have the character 'V' (56h) located at offset 12h.

```
PS-MPC.331.B        8D96 4B02 CD21 81BE 4B02 4D5A 7542 80BE 5D02 5674 3BE8 7300
```

PS-MPC.388

CN: An appending, 388-byte, direct infector based on PS-MPC. It contains the message: '(C) Copyright 1981-1994 Microsoft Corp Licenced Material - Property of Microsoft All rights reserved', '*.COM'.

```
PS-MPC.388          2E8B 86A3 022E 8B8E 8402 81C1 8701 3BC1 7428 2D03 002E 8986
```

PS-MPC.517

CN: An appending, 517-byte, PS-MPC-based, direct, fast infector. It contains the messages: '(C) Copyright "Desecrated Soul" 666bc-1994ad DΣSΣCRÆTΣD SOúL - Born: 18th May ----ad', 'Your PC is *NOT* stoned! It has been DΣSΣCRÆTΣD', 'Goodbye friendly little disk drives. Love, from: DΣSΣCRÆTΣD SOúL', '*.COM'

```
PS-MPC.517          2E8B 8624 032E 8B8E 0503 81C1 0802 3BC1 7428 2D03 002E 8986
```

Rabbit.292

CN: An appending, 292-byte, fast, direct infector. It contains the texts: '.COM', 'The Rabbit Virus By: Corrupt of Death Row'. All infected files have the character 'V' as the third byte of a file.

```
Rabbit.292          B43F B904 008D 9620 02CD 2180 BE23 0256 742F B802 4233 C933
```

ShineAway.620

EN: An appending, encrypted, 620-byte direct infector, containing the text: 'Eternal love, is like heaven, sometimes like eternal rain, sadness, deep inside pain! [Shine Away] oO 1995 by CoKe Oo Made in Luxembourg 1995'.

```
ShineAway.620       3E8B B64A 038D BE11 01B9 1C01 3135 83C7 02E2 F9C3 E8E9 FFB9
```

Sword.794.B

CER: An appending, slightly corrupted variant of Sword.794 containing the text: 'The POWER of my SWORD!!!'. It does not infect COM files properly. Its code is attached to the end of a file but not executed. The template below detects both variants.

```
Sword.794           B821 35CD 2189 1EBC 028C 06BE 02B8 2125 BAB3 02CD 211F 8CD8
```

Trivial.82

CN: A simple, overwriting, 82-byte direct infector. Since the virus is encrypted, there are only eight constant bytes. All infected files have date and time stamp set to 00.08.1981 and 00:00:00.

```
Trivial.82          BE09 0180 34?? 46E2 FA
```

V.814

ER: An appending, 814-byte virus. Infected files have the number of minutes in the time stamp set to 13. Instead of using the 'Are you there?' call, the virus checks for value 02CCh at location 0000:02CCh.

```
V.814               33DB 1E8E DB56 E857 0051 5789 1FFF 0E13 048B 1E13 04B1 06D3 E326
```

V.2435

CN: An appending, encrypted, 2435-byte direct infector. It corrupts some infected files. It contains the text: '*.COM', 'TEMP.TMP', 'HM\A'. The template is always found at offset 0600h.

```
V.2435              BA83 09BE ???? 8A1C D0C3 881C 464A 83FA 0075 F3B8 ???? FFE0
```

Zipper.B

CER: An appending, encrypted, slightly polymorphic 2779-byte variant of the Zipper virus. It contains the text: 'Dec 3 92 is my 20th birthday (V6)'. It differs from the original in that it does not contain the 'zipped' source code of another virus at the end of its code. Executing infected EXE programs hangs the system. Two templates cover all generated samples (the same stings may be used to detect Zipper.A).

```
Zipper (i)          E800 005E B9C4 0A8A 1E05 0183 EE03 8BEE 305C 1846 E2FA 9090
Zipper (ii)         B9C4 0AE8 0000 5B83 EB06 8A36 0501 8BEB 2877 1843 E2FA 9090
```

INSIGHT

Where My Eyes Look

Dmitry Gryaznov has come quite a distance to practise his art – thousands of kilometres, and across a whole continent, to be precise. Gryaznov was born and grew up in the former Soviet Union, and attributes *Perestroika* as the reason for his being in the UK at all.

'I was born, and spent my early childhood, in what is now the independent state of Kirghizstan. This is a relatively small republic, set in what was one of the former Soviet central Asian republics, on the border with China. It is a mountainous country.'

At eight years old, he moved with his mother, an English teacher, to the Ukraine. She still lives in Dnetropetrovsk, one of the biggest industrial centres of the former Soviet Union, where rocket engines for intercontinental ballistic missiles were built. After secondary school, Gryaznov went to Moscow to study at the *Physics and Technical Institute*. Here, he had his first real exposure to computers: he graduated in 1984, with a masters degree in science.

Discovering the PC

As a child, most of Gryaznov's time was spent reading: 'My other hobby,' he reminisced, 'was mathematics and physics. So, I went to Moscow to study physics, and there, in the first or second year, I had a brief introductory course in programming. To most of the students, it was just another introductory course – me, I couldn't stop!'

Gryaznov's speciality at Moscow's *Physics and Technical Institute* was lasers and quantum electronics. This did not hold his interests as computers did, so, in the fourth year he altered his studies to include computer-related topics. Six months before the end of his degree, he decided to alter his thesis to writing a cross-macroassembler and an emulator for the *Intel 8080*.

It is policy in Russia that those young people who go on to institutes of higher education fulfil their obligations to the military during the course of their studies: while he was a student, Gryaznov completed one month of in-the-army training and a five-year course of Military Studies. The Russian *Ministry of Defence* also decides which educational institution prepares for which role: somewhat ironically, Gryaznov's *Institute* had been selected to study intercontinental ballistic missiles!

Although he is in theory still under obligation to the Russian Armed Services, as he is classed as an older man (at 34!), and has a family, he would not be called to serve except in the event of a full-scale war: 'Not for conflicts like those in Chechnya or Afghanistan,' he hastened to add.

After university, Gryaznov decamped to the newly-founded *Program Systems Institute* of the *Soviet Academy of Sciences*, in the ancient town of Pereslavl'-Zalesskij. 'One of the great figures of Russian history, Prince Aleksandr Nevsky, was born there,' explained Gryaznov. 'The town was his feudal seat. In his youth, Tsar Peter the Great used to play with his model navy on the lake there – under his leadership, the Russians later had one of the world's strongest navies.'

Gryaznov gained experience through a wide spectrum at the *Academy*, beginning by working as a system programmer on Unix kernels, and writing hardware device drivers. 'Another of my projects,' he said, 'was porting a Unix system to a computer that had only 56K of memory. I also worked with communications. The *Institute* had Bulgarian clones of *Apple II* computers which didn't have RS232 built in, but which had games ports: I modelled RS232 using software. I guess you could call this period my apprenticeship. I went on to write a C compiler for a Russian multi-processor.'

Foreign Bodies

An 'extra-curricular' activity for Gryaznov at the *Academy* was working with children; more precisely, teaching at a computer summer camp for children, which was sponsored by the *Academy*. This started as an initiative for Russian children, but with the advent of *Perestroika*, foreign children started attending. In the summer of 1988, there were students from as far afield as West Germany and the United States.

'At this camp, we noticed that some of the computers started to reboot on their own,' related Gryaznov. 'At first I didn't take much notice. I'd heard about viruses, but only knew that they were somewhere there in the West. It was the Vienna virus, of course; the very first reported case of a computer virus in the Soviet Union.

'Vienna was at the time very widespread in the West, and many of the kids had brought diskettes with them, computer games mostly. I think there was an infection on one of those; brought most likely by German kids. As Vienna infected only COM files, I wasn't very interested – most of the programming we did was with EXE files.

'About a month later, back at my job, I had this falling letters effect on my screen – it was Cascade, and it was very annoying. I was in the middle of a very important and interesting project. So I got angry, armed myself with a debugger, and for the first time disassembled a computer virus. Within half-an-hour I had my first scanner. Other viruses rapidly appeared – this was how it all started.'

Gradually, more and more of his time at work became dedicated to all aspects of viruses. Gryaznov was given more or less a free hand; allowed to work on viruses as long as his other projects

continued: 'In the Soviet Union we had a joke,' he said. 'They pretend to pay me a salary; I pretend to work! It wasn't like it would have been in a business company; they were research projects, where you cannot have clearly-defined tasks completed to a strict schedule.'

By the time Gryaznov left the *Soviet Academy of Sciences* in 1994, 99% of his time was spent working with viruses – this despite the fact that they were still very much an unofficial part of his job.

'They paid me for other things,' he explained. 'For example, I set up the Internet email for our *Institute*, and was in charge of that for a while. We also tried to make and sell our own Russian anti-virus product. For a while, some of my income was from selling this product – it was already more than just my salary from the *Academy*.'

None of this, Gryaznov feels, would have been possible without *Perestroika*: 'First, it became possible to discuss things more openly. I mean, I was never a dissident; in fact, I was a convinced Socialist, even a Communist. I had grown up in those circumstances, and all we could read or hear in the media were Soviet official things.

'*Perestroika* was started by the General Secretary of the *Communist Party*: the country was changing; the people were changing; I was growing and changing too, realising more and more things as more information was becoming more freely available.

'In 1990 I went abroad for the first time; I spent a month in the US on an exchange programme with the computer camp. I was interested in how Americans were using computers in school education, so I spent a lot of time with schoolteachers' families; two or three nights with each one. It affected my view of the world greatly; I saw that much of what we had been told about the West was lies. Lifestyles of teachers there were incomparably higher not only than their Russian equivalents, but also than *Communist Party* officials!'

Going West

The two greatest things achieved by *Perestroika*, in Gryaznov's opinion, were freedom of speech and freedom of travel. The latter enabled him to take up a post in England, at *S&S International PLC*, where he works on viruses.

'Freedom of travel is a relative thing: when the Iron Curtain was down, it was necessary to get a special permit to go abroad, even to Eastern Bloc countries. Then, Western countries welcomed people from the Soviet Union, helping them get there, and stay there if they wanted. When the Iron Curtain was removed, Western governments immediately built their own curtain from the other side. Now you don't need special permission to get out of Russia, but it's difficult for a Russian to get a visa for the West.'

Over a period of time, Gryaznov built up contacts with researchers in the West, such as Fridrik Skulason and Alan Solomon. In 1992, he planned his first professional trip to the US, presenting a paper at the *Ides of March* conference. Unfortunately, with the breakdown of the Communist regime, financing for such luxuries as conferences in the West also disappeared, and Gryaznov was faced with the prospect of being unable to present his paper.

Dmitry Gryaznov has brought a great deal of experience and knowledge with him from Russia, and plans to build on his expertise here in England.

At this point, Fridrik Skulason commissioned him to do a virus analysis, in return for which Skulason sponsored Gryaznov through the conference: he then did many other virus analyses; in the main for Skulason.

After much consideration, Gryaznov decided to look for a job in the West: 'I wasn't badly off in Russia,' he said, 'mainly because I was doing work for Western anti-virus researchers. This put me in a much better position than the average Russian, although I wasn't making much by Western standards. I realized that it was better to be an average person in a wealthy country than a wealthy person in a country of poverty.

'Also, nowadays, to have more than average is just dangerous, because of what they call the "Mafia" – that was another reason I wanted to get out of Russia; I just didn't want to jeopardise my family.'

He recalled how difficult it was at first for him and his family in England: 'Neither my wife nor son could speak English. There is a rather barbaric method of teaching someone to swim: just take them to the water and throw them in. This is what happened to my family with English. We put my son straight into an English school – it worked well. The teachers asked my wife to go with him; to help him a little. She made friends with the mothers of his classmates, and both their English has improved.'

Gryaznov plans to stay in the West, and has no immediate plans to leave Europe for anywhere else: 'The American way of life is somewhat… different. Probably I could have adapted to it eventually, but I find the Western European, the English style of life much easier. Russians are still Europeans, after all!'

Working on the Wild Side

Gryaznov has now been in England, at *S&S International*, for about 19 months: 'My position is Senior Virus Research Analyst. In fact, in addition to routine virus analysis, I'm doing a lot of research projects. I've been very much involved in developing version seven of the *Toolkit*, helping it scan inside PKLITE and so on. My main project so far has been developing the heuristic capabilities for FindVirus.'

Virus trends are constantly changing, he feels: 'Soon after I moved here, we had a blast of polymorphic viruses; not just SMEG, but lots of others. We are currently seeing several new polymorphics each week. A polymorphic virus was a special event a couple of years ago.

'Nowadays, if a virus is *not* encrypted, this is something special. We have already also faced several viruses written specifically to avoid heuristics. We will see more of these because, as a result of the rapidly increasing numbers of viruses, more products use heuristics in detection.'

Many viruses are just as prevalent in Russia as in the West: Gryaznov believes that this is due to the fact that the best viruses replicate successfully anywhere.

> *"as a result of the rapidly increasing numbers of viruses, more products use heuristics"*

'Most viruses are primitive creations,' he explained, 'which are easily detected and eradicated, and reveal themselves very soon in the normal operation of the computer. Those which are successful eventually make it to other countries as well. We see some Russian viruses here in the West, but Russian viruses stay mainly in Russia. The same applies to other countries.'

Legal Issues

There is not much that can be done to counter the threat from virus authors, in Gryaznov's opinion: 'I don't really think there's a way to deal with them – there are skinheads, kids who smash windows… All kinds of teenagers do nasty things. It's a problem all over the world. Viruses are just another of that sort of problem. Legal redress would help, but it cannot eliminate the problem.

'Where properly enforced, legislation will help, but the problem will remain. Look at the Black Baron: I've seen his viruses, and the way he does them – he's 26 years old, but he has the psychology of a 13-year-old boy. He must be punished; they shouldn't let him go. He's supposed to be an adult; he should be taking responsibility for his actions.'

Gryaznov is only too aware of the problems involved in legislating against computer crime, both here and in Russia: 'We didn't have copyright laws in Russia until two or three years ago. Even now, these laws are probably impossible to enforce. You can go to any kiosk in Moscow and pick up a copy of almost any Western movie: it will be pirated, and dubbed into Russian. Nothing can be done about it – the government has more important things to do; you know, like taking Chechnya.

'Recently they also voted for a specific computer crime law in the Russian Parliament, but it's badly formulated. It's intended to protect data – anybody causing loss of data can be prosecuted. Even an electricity surge, however, can cause a loss of data, so if you follow the law to the letter, you could sue the manufacturers of whatever product caused the surge! There are two virus-related newsgroups in Russia, and I follow them, and read about such things. This law is just another tool for the government and the police, etc, to affect those they want to.'

From Culture to Culture

As a Russian Orthodox, Gryaznov sees many cultural differences between Russia and the West. England he views as a secular country; unlike his perception of the US. Russia, he feels, is in a state of flux: 'After the Russian revolution in 1917, my country was no longer religious – worship was strongly discouraged. Nowadays all that is changing; now it's almost a fashion for people to go to church and to pray in Russia.

'In fact, until recently, I myself wasn't at all religious – I was brought up with the Communist ideology. I'm very different from the Dmitry Gryaznov who joined the *Russian Academy of Sciences* in 1984.'

On the Home Front

Gryaznov feels comfortable here in the West: 'I like the variety and the freedom of choice: this applies not just to Britain, but to the whole Western world. Also I like the attitudes people have towards each other – if you pass someone in the street, they will often say hello. Russians don't do this. School education is different: the attitude here is much better than in Russia, but the quality of education is better in Russia than in England.

'One thing I discovered when I first went to the US – we were told that the environment in the West was awful, that no-one looked after it, and so on. When I got there, I found that this was not true. In the West, people actually take more care of the environment than in Russia. People here like the countryside better than in Russia.'

England is a land of contrasts to Gryaznov, and he spends much of his free time exploring these diversities with his family: 'We like to get in the car and drive just anywhere… As we say in Russian; "Where my eyes look".'

Where his eyes look has brought Dmitry Gryaznov to the Western world. For now, his eyes are firmly focused on England and on virus research – where they wander next remains to be seen.

CONFERENCE REPORT

DefCon: Fear and Loathing?

Richard A Ford

Picture if you will a room filled with over 250 people of many different ages… picture a casino; the desert sun beating down… picture people standing discussing friends, laughing and arguing, some huddled around a single computer, some not. Picture this, and you have some idea of the scene at *DefCon III*, held at the *Tropicana Resort and Casino*, Las Vegas, on 3–5 August. Ever wonder what happens at a 'hacker' conference? Read on to find out.

The State of Play

I cannot think of a better place for a hacker convention than Las Vegas, a town where the line between establishment and anarchy is fine indeed. That said, arriving at the *DefCon III* registration desk was much like signing up for a so-called 'real' conference. For my US$40, I was given a multi-colour badge, with spaces for name and email address, a free copy of *2600*, and a booklet detailing what was happening when.

No real name was required, and apart from one attendee who took one look at me and said 'He's a Fed', registration was trouble free. It granted me access to a large room, lined with tables on which a few people were selling their warez (sorry, I mean wares – I saw nothing illegal going on, and by and large the behaviour of delegates was no better nor worse than one would see at any conference). A notable display was provided by *Secure Computing* (not the UK magazine, but the US company which sells *SideWinder*, the firewall), which was challenging any and all to hack its product.

The Shape of Things to Come

As with all conferences, the quality of speakers and their topics varied widely. Again, though, one should not leap to conclusions: the content and presentation skills of most of the speakers was high, and sessions were interesting. Space restrictions preclude me giving a full review, so apologies in advance to speakers I fail to mention – I will simply look at points that show the conference in its many colours. Topics covered a wide range, from *Why Hacking Sucks* (Steven Cobb) to *Social Engineering and Psychological Subversion* (Susan Thunder): something for everyone!

Perhaps the most memorable for me was Karen Coyle, from *CPSR* (*Computer Professionals for Social Responsibility*), arguing against aspects of copyright and the Internet, being waylaid by attendees who thought their work deserved to be protected. Another stereotype bites the dust.

Although some talks were what one might expect at such a gathering, there were several of a serious nature: one by Koresh, entitled *Hacking the Real World*, discussed the use of hacking skills in real life to get a job, where you can work legally with systems. Other speakers included 'Dead Addict' and 'Veggie'. Dead Addict talked about the ramifications of computer-based technology on the mindset of 'the people', pointing out that we are ultimately responsible for who we are. Veggie's talk – less philosophical, but practical – related tales of his less-than-gratifying interactions with the press, provoking audience dialogue on how to find a journalist who does not bow to the almighty dollar.

As a group, the attitude of those present was largely against illegal behaviour – I observed none of the drunken melée I expected after tales of other such conventions. (Even at Winn Schwartau's excellent 'Hacker Jeopardy' quiz, where a good time was had by all, people were controlled and considerate. High spirited, yes. Badly behaved, no.)

People

Like most conferences, many of the most interesting moments happened away from the hall. Sarah Gordon (aka Theora, a regular *DefCon* speaker), still 'Caught in the Crossfire' [*see VB March 1995 p.7*], introduced me to some virus writers, and some anti-virus product developers I had not met. Although I cannot name names (a condition of the introductions), the virus writers to whom I spoke seemed of above average intelligence, eloquent and likeable.

While I will never agree that virus writing is a 'good thing', meeting some of the 'other side' taught me something: there is room for considered and well-placed dialogue. I came away from these meetings wiser, and somewhat surprised.

Closing Thoughts

I had a good time at *DefCon III*, and learned a lot there. Congratulations to Dark Tangent, the conference organiser, for keeping things running smoothly, and putting together an interesting program. To those who think hackers are a bunch of irresponsibles, or stupid and uninformed, I would say go to *DefCon* next year… but with an open mind.

If you have something to say, this is the perfect forum. While those present may or may not agree with you, your ideas will be listened to. You won't agree with some of what is said, and people will talk on subjects you may not like, but this is balanced by the fact that you will be given time to air your views. I found it well worth the trip. The future lies in agreement and understanding: this can be achieved only through meaningful dialogue.

And if you were wondering, there was no 'As I sat back in the plane and sipped an ice-cold Gin and Tonic, I recalled...' as I left *DefCon* – only the current Editor can write that. What I can say is that I expected to see fear and loathing, but what I saw was fascination and learning… on all sides.

VIRUS ANALYSIS 1

Burglar: The New Pretender

Eugene Kaspersky
KAMI Associates

One of the most difficult problems in the war against viruses is polymorphism. It is well known that to adapt the anti-virus scanner to detect new polymorphic viruses is a very complex task; more complicated than adapting the scanner to detect dozens of new non-polymorphic viruses. But the polymorphics keep on coming, using new engines, bringing new problems. The latest in this sequence is Burglar.

Burglar is the obvious choice for this virus' name – there is a text line in the virus code, the second part of which reads:

```
written by Burglar in Taipei, Taiwan
```

I see that Taiwan is in the lead not only in the field of PC manufacture, but also in that of computer viruses – many of these have recently been received from Taiwan. The first part of the text line is:

```
Hello! This is [Super Virus-2]
```

The next Great Pretender to be the Super Virus on the computer virus scene. Not the first, not the last, and certainly not 'super'.

Installation

On execution of an infected file, the system passes control to the polymorphic decryption routine, which restores the virus code in its original form. After decryption, the virus installation code starts to infect the system.

As is usual for memory-resident viruses, Burglar checks the system to ensure that it is not already resident by performing an 'Are you there?' call (Int 21h with AX=ABCDh). If the system returns the value DCBAh in the AX register, the virus immediately returns control to the host program.

If there is no TSR copy, Burglar installs itself into memory. It frees part of the system memory with a ChangeMem DOS call, then allocates a new block of the memory with an AllocMem call. It marks that block as 'system' (value 0008h in MCB Owner field), and copies itself there.

The method used by Burglar can be called 'good form' – the virus installs itself into UMB blocks, if present. To do that, Burglar performs several calls to the DOS Memory Allocation Strategy functions (Int 21h, AH=58h).

The first virus to use this UMB route to install itself into memory was Tremor. That was quite long time ago, but did not lead to all new viruses using this method. It may show that the average virus author is a very lazy person – installing the virus into the UMB is not difficult, but the authors do not take the time to do this. Hundreds of new viruses still use old methods to install themselves into memory – the 'DOS version 3.30' methods were used in Cascade, Yankee Doodle and Dark Avenger (Eddie) viruses several years ago.

The author of Burglar is not lazy: the virus includes much more than just the UMB infection routine. There are complex methods for interrupt hooking, the trigger routine, and the polymorphic engine.

The virus hooks Int 21h to infect files. It obtains and saves the Int 13h vector, and uses that value in the infection routine. Depending on the system date, the virus will also hook Int 08h.

To obtain the original addresses of Int 13h and Int 21h, the virus uses several undocumented calls to the system. To obtain the original Int 13h address, the virus performs Int 2Fh, AH=13h call which returns the interrupt address that usually points directly to the BIOS.

To obtain the Int 21h address, the virus uses the address of the CP/M handler. That address is compatible with very old applications only. I know of no applications which use that at this time (except viruses, of course), but this 'back door' is present in all versions of DOS.

Using the address of the CP/M call, the virus scans memory for the code of the original Int 21h handler, and patches it with two FAR JMPs. I see no reason to patch the DOS Int 21h handler with *two* far jumps, but the fact is that the virus overwrites two five-byte blocks of Int 21h handler with two FAR JMP instructions, and there are two corresponding Int 21h handlers in the virus code.

Burglar uses an unusual method to make the system appear uncorrupted: it overwrites the Int 21h handler with 10 bytes of the two FAR JMP instructions. Usually, viruses use the 'restore/overwrite-back' method to patch the handlers during installation. They then receive control from the patched code, and restore that code before returning to the original handler.

A.	B.	C.
CLI	CLI	CMP AH,6Ch
CMP AH,6Ch	JMP far Virus	JA Loc_c
JA Loc_a		JMP far Loc_b
Loc_b: CMP AH,33h	Loc_b: CMP AH,33h	Loc_c: JMP far Loc_a
…	…	
Loc_a: …	Loc_a: …	

Figure 1: A. The Int 21h handler before being patched.
B. The Int 21h handler after being patched.
C. The virus handler's return code.

Next, these viruses again overwrite the same bytes of the handler and take control on the next interrupt call (see Figure 1, left).

Burglar does not restore the Int 21h code on returning to the original handler. It uses another method: before patching, the virus copies the overwritten bytes into the virus' Int 21h handler code. On Int 21h calls, the virus executes that code in its (virus) handler, and passes control to the original Int 21h code.

Two Int 21h Handlers

The first Int 21h virus handler hooks four functions: one for the 'Are you there?' call (AX=ABCDh); FindFirst/Next (AH=11h, 12h); and ExtendedOpen (AH=6Ch). The second handler intercepts Lseek Home/End (AX=4200h/4202h), Open (AH=3Dh), Get/Set File Attributes (AH=43h), Execute (AH=4Bh), and Rename (AH=56h).

When a file is accessed, the virus checks whether it is already infected, using one of the most popular algorithms, the '100 years' stamp. During infection, the virus adds 100 to the year field in the file's date and time stamp. On access, the virus checks that field for the '100 years' label.

On FindFirst/Next calls, the virus checks the file to see if its code is present (using the '100 years' label), and passes control to the infection routine if the file is clean. It does this also on Open, Execute, Rename, and Get/SetFileAttribute calls. In the last case, if the file is not infected, the virus subtracts 100 from the year field and returns the result – this is part of its stealth functionality.

On Lseek calls, the virus uses the second part of the stealth routine to make it impossible to Lseek to the virus code in infected files.

Infection

The infection routine checks the file name and extension. If the first part of the file name is COMMAND, or if the file extension is not COM or EXE (i.e. it does not have an executable extension), the infection routine aborts.

Then the virus calls the polymorphic engine, encrypts the virus body, and writes the polymorphic decryptor and the encrypted file body into the file – at the beginning of a COM file, or at the end of an EXE file. The original contents of the COM file are shifted down (the Jerusalem virus also does this) by the length of the encrypted virus.

The length of infected files is increased by a random value – the polymorphic routine generates decryption code of a random length, so files increase by the length of the virus (3260 bytes) plus the length of the decryptor.

To bypass memory resident anti-virus utilities, and prevent an error message if it attempts to write to a write-protected disk, the virus temporarily sets the Int 13h handler to its original value (stored during installation), and hooks the Int 24h DOS Error Message handler. The virus also obtains and restores the file attributes and the date and time stamp (the virus increases that stamp if infection is successful).

Trigger Routine

On the days when the date is equal to the number of the month (1 January, 2 February, etc) the virus hooks Int 08h and displays a blinking message at the top of the screen:

```
Hello! This is [Super Virus-2] ... written
by Burglar in Taipei, Taiwan
```

There is a bug in this virus: it overwrites the original address of the Int 08h handler during infection of the next file, and, as a result, the computer hangs.

The Polymorphic Engine

Burglar is the first (and, at the moment, only) virus to use the PME – 'Phantasie Mutation Engine'. This engine contains the internal text string:

```
PME v1.01 (C) Feb 1995 By Burglar
```

This leads one to think that both the virus and the engine were written by the same person. From a technical point of view, the engine brings no great surprises. It generates a simple decryption (XOR) loop with numbers of junk instructions.

Burglar

Aliases:	None known.
Type:	Memory-resident parasitic infector, polymorphic.
Infection:	COM and EXE files only.
Self recognition in Memory:	The virus returns DCBAh in the AX register in response to an 'Are you there?' call (Int 21h, AX=ABCDh).
Self-recognition in Files:	Adds 100 years to the file's date stamp.
Hex Pattern in Memory:	(NB: this virus is polymorphic, with no simple hex pattern in files) `E800 005B 80FF 0374 032D 1000 5068 1600 CBB8 CDAB CD21 3DBA`
Intercepts:	Int 21h for infection, Int 08h (timer) for trigger routine.
Trigger:	Displays message, halts PC.
Removal:	Under clean system conditions, identify and replace infected files.

VIRUS ANALYSIS 2

DiskWasher

Kevin Powis
Precise Publishing Ltd

DiskWasher is a boot sector virus which is to be found 'in the wild'. Its payload corrupts the first four sides of a hard disk and both sides of a floppy. On a hard disk, the last 66 bytes of the MBS are reserved for the PC's partition table and the standard two byte boot signature (55AAh), but the virus fits easily in the remaining space.

During the boot process, the PC's firmware loads the first sector from the boot disk (the MBS) into memory at offset 7C00h in segment zero and passes control to that address to start the boot program (in this case, the virus).

DiskWasher is structured as follows: installer, hard disk infection code, interrupt vector hooking, original boot processing, and a payload interrupt handler (which contains trigger detection and floppy infection code).

Installer

The Installer is invoked by the firmware as described above. This decrements a word in low memory by 1. This word is used by the PC to hold the size of conventional memory in 1K chunks. By decrementing this, a 640K PC appears to only have 639K. The missing 1K is at the top of memory: the virus can reside there without fear of being overwritten. The virus next copies an image of itself to this area, and passes control to that image.

Once relocated, the virus checks to see if the hard disk is infected, attempting to read head 0, cylinder 0, sector 10 – an area which is normally unused. On an infected PC, this holds the original boot sector, so if the last two bytes are the boot program signature (55AAh), the virus assumes that infection has taken place and hooks the interrupt vectors.

If the read fails, the virus assumes it is working on a 'floppy-only' system, and sets an internal flag (located at offset 10 in the virus body) to 1 to indicate this. It then retrieves the floppy boot sector (head 1, cylinder 0, sector 3), and hooks the interrupt vectors. The flag is set to 3 if a hard disk is deemed present.

Hard Disk Infection Code

If the read worked, but the necessary signature is not present, the virus begins its infection routine. It reads in the current clean boot sector and writes it, unchanged, to sector 10. This original sector will be required by the virus to complete each boot sequence and, once stored in sector 10, will indicate to the virus on subsequent reboots that the PC is already infected.

With the clean boot sector stored away, the virus takes the partition table entries from this sector and copies them into its own body at the correct offset (1BEh). These are the 66 reserved bytes mentioned earlier. To complete hard disk infection, the virus image (complete with stolen partition table) is written to the PC's MBS. The virus then sets about putting its own interrupt handler in place.

Interrupt Vector Hooking and Boot Processing

Int 13h is used by the PC as an API to access the BIOS disk routines. So, under DOS and *Windows 3.x*, file activity eventually falls to a request for disk access channelling through Int 13h. With this in mind, DiskWasher takes a copy of the current vector, using it to build a far pointer to BIOS routines. Then it overwrites this vector with a pointer to its interrupt handler, thus taking control of Int 13h just before it enters the BIOS.

"one strange side effect is that when you boot from an infected floppy, the PC appears to boot from the hard disk as normal"

Once the disk handler is in place, DiskWasher resets an internal counter at offset 8 in the virus body to zero. This controls the trigger mechanism and is described later.

The virus is now installed and the PC continues booting as normal: control passes to the original clean boot sector's image, which is still in memory. One strange side effect is that when you boot from an infected floppy, the PC appears to boot from the hard disk as normal – users do not even get the message, 'Non-System disk or disk error' which might alert them that they have attempted to boot from a floppy.

Interrupt Handler – Trigger Detection

When the PC is up and running, the virus takes control briefly with every disk access courtesy of the Int 13h hook. The handler starts by incrementing the word at offset 8 in its own body. This, you may recall, was reset to zero as part of initialisation. When word 8 reaches a value of fifty thousand, the payload (described later) is released, and all is lost.

If this trigger condition is not satisfied and the interrupted disk access is destined for a hard disk, the request is passed unhindered down through the interrupt chain. The virus then goes to sleep, waiting for the next disk access.

If the interrupted access relates to a floppy, a second variable is incremented: a byte at offset 7 in the virus body. While this byte has a value less than eight, the disk action is passed on

unhindered. When the counter reaches 8, it is reset to zero and DiskWasher checks the floppy to see if it is uninfected. If so, it will then attempt to infect the floppy.

DiskWasher is not, however, just infecting every eighth floppy. Even a simple DIR listing will result in more than eight disk accesses; thus, it is almost a certainty that every disk used will be infected.

To determine whether or not the floppy is infected, DiskWasher reads in the floppy boot sector. It does this twice to cater for potential read failures due to the drive not being up to speed. It then compares the word at offset 0178h in the boot sector to the value 019Eh. This will be the case for an infected floppy because offset 0178h is within the snippet of virus code 'MOV CX,19eh'.

If this identifier is absent, the floppy is deemed uninfected: the virus then copies the clean floppy boot sector to head 1, cylinder 0, sector 3, placing its own image over a selected part of the clean boot sector in memory. This effectively converts it to a virus image, while preserving values that must be retained, such as the Bios Parameter Block (which describes the floppy's structure). Once done, rewriting the amended image to the floppy boot sector completes infection.

It is worth pointing out that on a 1.44MB 3.5-inch diskette, the sector DiskWasher uses to store the clean boot sector is the second sector in the root directory area. Thus, any such floppy with more than 16 files in the root (including volume label) will be damaged. On a 720K 3.5-inch floppy, this sector falls further down the root directory, giving a better chance to avoid damage during infection. DiskWasher will infect floppies accessed in any drive, not just the A: drive.

Once the floppy has been checked (and possibly infected), DiskWasher allows the original interrupted disk request to pass down the interrupt chain and waits for the next access.

Payload

As mentioned, the payload is invoked on the 50,000th disk access of any session. This routine picks up the contents of offset 10 of the virus body (set up at installation) – 1 for floppy-only; 3 for hard disks. This controls the target drive and the head for the format routine to follow. If the value is 3, the target is forced to be drive 80h (the first hard disk) – otherwise it remains as set by the original disk request. The value is then used as the start head for the format. On a hard disk the format starts on head 3; on a floppy, head 1.

The destruction in the payload is caused when DiskWasher enters a continual loop using the BIOS to format 5 sectors from the start of each track before decrementing the head and repeating while the head value is greater or equal to zero. It then displays the message: '♥ From Diskwashef', followed by a few garbage characters. [*This appears to be a variant of another form of DiskWasher, which triggers at the 5000th disk accesses and displays the message '♥ from DiskWasher with love ♥'. Ed.*]

After the message is displayed, DiskWasher moves to the next track, resets the head to 3 or 1 and repeats the trigger. The message will be displayed each time the virus moves to the next head. If it is not interrupted by the user, the process continues until all available tracks on the first four heads are covered. The loop then keeps going, but as format calls are now illegal, disk activity ceases.

When formatting tracks via the BIOS, the programmer passes across a pointer to a table that controls how sectors are formatted: virus authors seem immune from such mundane tasks. DiskWasher passes across a pointer that is uninitialised, so will point to random bytes in memory. This effectively makes the disk non-standard, ensuring that, after format has taken place, you cannot even access the disk with a disk sector editor in an attempt to salvage something.

On hard disks with more than four heads, some data will escape the payload. Unfortunately, as the virus has corrupted contents and structure of the first four sides of the hard disk – including File Allocation Table (FAT) and Directory Area – recovery will be decidedly non-trivial.

Conclusion

DiskWasher is well written and effective. It does not employ any stealth capabilities. It will infect floppies used in any drive and moves from the floppy to the hard disk without even raising a murmur. It is in the wild and dangerous.

DiskWasher

Aliases:	None known.
Type:	Boot sector infector.
Infection:	Floppy and hard disks.
Self-recognition on Hard Disks:	Head 0, Cylinder 0, Sector 10 contains valid boot sector as signified by 55AA being last two bytes.
Self-recognition on Floppy Disks:	Head 0, Cylinder 0, Sector 3 has word value 19Eh at offset 0178h.
Hex Pattern (hard/floppy disks, and in memory):	`B840 008E D8A1 1300 48A3 1300` `33DB 531F B106 D3E0 8EC0 33C0`
Intercepts:	Interrupt 13h.
Trigger:	50,000th disk access in any session.
Payload:	Format of first 4 sides of hard disk or both sides of floppy.
Removal:	Hard disk – under clean system conditions, use FDISK /MBR. Floppies – salvage required files, then format.

TUTORIAL

Windows 95: Even Better than the Real Thing?

On 24 August, with a crash, a bang, and a huge marketing budget, *Microsoft* unleashed *Windows 95*. The full release is the result of what must have been frantic effort within the hallowed halls of Redmond before the mastering; this has, amongst other things, moved the build number up to 950.

I was curious to see if the product had changed since I examined what happened when it was infected with boot sector viruses [*see VB, June 1995 p.15*], and to try some standard file infecting viruses in *Windows 95*. Once again, my overworked Opus 386/25 was used as the victim.

Part 1: Boot Sector Viruses

Here, there was a pleasant surprise. Heed appears to have been taken of the editorial I wrote in the June issue about the silence *Windows 95* kept concerning the fact that it had been infected with a boot sector virus: now it tells you (see figure 1) as soon as it boots up. Although this is a leap in the right direction, the warning is not 100% accurate.

Figure 1: This dialogue box is displayed when *Windows 95* believes it has detected a virus.

When I infected the system with Form (and nothing else!), however, the box was still displayed. It is well documented that Form modifies only the Partition Boot Sector (the term 'DOS Boot Sector' has some problems when the OS on the partition concerned is not DOS) to insert its code.

In addition, when I cleaned the system, and reinfected with Jumper.B, the message did not appear. Jumper.B is a MBS virus; the one *Windows 95* in a previous incarnation failed to spot. As discussed in the June article, Jumper.B (also known as Viresc) shuns the more popular boot sector virus technique of hooking Int 13h, hooking Int 21h instead.

Disk Performance

The control panel section which describes what is wrong with your system has also changed slightly – instead of the message 'MS-DOS compatibility mode file system: POSSIBLE VIRUS', it now reads 'Master Boot Record modified -- SEE IMPORTANT DETAILS.'

Telling Porkies

Could it be that *Windows 95* is lying to us? It seems unlikely that the OS is genuinely detecting changes to the MBS if it both fails to detect real changes (Jumper.B), and then, on the other hand, detects changes where there are none (Form).

This evidence supports the June article, which concluded that *Windows 95* was using the state of the Int 13h vector to determine that something has changed. This latest build, however, appears to *assume* that the change occurred within the MBS. It then tells the user that this is definitely the case.

It would be churlish to say that this situation is worse than before – it is undoubtedly better. However, there is no disputing the fact that the information displayed is incorrect; technically correct as well as helpful would be nice.

Part 2: File Viruses

There has not yet been time for a fully fledged investigation into the effects of file viruses on *Windows 95*. However, a preliminary study has been undertaken, and reveals a number of things, some of which are discussed below.

Command Prompts: Doing it the Old-fashioned Way

In *Windows 95*, programs operating with an *MS-DOS* box (sometimes called a command prompt window) have their own little environment to play with. The best analogy is perhaps one of plants in a greenhouse – as far as the plants are concerned, the greenhouse is all there is. This only holds true up to a point – if the plants are triffids, they will walk out and discover that there is more. So it is with programs running in command prompts – as long as they are well behaved, they will never find out that the glass in their greenhouse is actually fairly easy to smash.

Bearing all this in mind, it is reasonable to assume that a standard DOS memory-resident virus will function without problems in such a box – the virus will go resident, hook the desired interrupts, see calls to them before the calls leave the greenhouse (sorry, DOS box), and proceed to the *Windows 95* interrupt handling system. With Jerusalem (Jerusalem.1808.Standard, to the *CARO*-esque amongst us), this is the case. The virus goes resident (it is visible with DEBUG, and has correctly hooked the appropriate interrupts), infects files as they are run, and the 'children' are themselves viable.

GUI-ing with the Best of Them

In the wonderful world of *Windows 95* this is not how things are supposed to be done. We want to point and click; to throw away the traditional keyboard in favour of the new, more modern, smaller one – the one that fits neatly under your hand,

moves around the desk, and has fewer buttons than you have fingers.

With *Windows 95*, DOS applications may, of course, be launched from the GUI, as with previous versions of *Windows*. When a sample of Jerusalem is launched like this, *Windows* comes over all clever. It sees the virus going resident, assumes it is some form of pop-up utility, and displays the following message within the DOS box which it has opened for the program:

```
     MICROSOFT WINDOWS POP-UP PROGRAM SUPPORT
Your pop-up program is ready to run. When you
have finished using it, press CTRL-C to close
this window and return to Windows.
```

At this point, as the message suggests, the box can be used for nothing else – when CTRL-C is pressed, the box exits, taking the resident copy of Jerusalem with it (the greenhouse is smashed). Obtaining this effect when a program is run is almost bound to alert the user to the fact that something is amiss. They are likely to complain to the appropriate people, if only because they can no longer run their program.

GUIs from DOS

One of the major failings of previous versions of *Windows* was that its executables could not be run from within a DOS box – you had to run them from Program Manager or File Manager. *Windows 95* corrects this problem – at least, for 32-bit *Windows 95* executables it does; it is still not possible to run a 16-bit *Windows* application from a DOS box. This should cease to be so much of a problem as more and more genuine *Windows 95* software reaches the marketplace.

So, what if you run a *Windows 95* executable from a DOS box in which Jerusalem is resident? It works, and the program is not infected. If, however, you boot DOS 6.20, install Jerusalem, then run the same program, it is infected (and rendered useless to *Windows 95*). To understand this, we must undergo a brief foray into the world of stubs...

Stubbing it Out

A stub is a program within a program; more accurately, a small program stored before the main program. It takes the form of a DOS executable, the sole purpose of which is to print a message informing the user that the program is meant to be run under OS *x*. This is a DOS program: DOS is a sort of lowest common denominator; which *OS/2*, *Windows 95*, and *Windows NT* all understand. Such stubs are typically 400h (1024 decimal) bytes long, and take the form of a standard EXE file – the first two bytes are 4D5Ah ('MZ').

When such a program (for example, PROGMAN.EXE from the *Windows 95* installation) is run from DOS, all DOS sees is the small program: this is where the problem lies. Jerusalem is a careless virus, and does not worry about how big the file really is. It takes a quick look at the size of the base executable, writes its code at the end, and patches the entry point. So, when Jerusalem infects PROGMAN.EXE, it overwrites 1808 bytes of the 32-bit application code with its body, destroying the 32-bit incarnation of PROGMAN.EXE, but leaving the DOS stub fully functional (PROGMAN.EXE now contains an intact and replicable Jerusalem infection).

When a clean PROGMAN.EXE is run from a *Windows 95* DOS session where Jerusalem is resident, it is not infected. Why? The reason is that *Windows 95* is ignoring the DOS stub. (Why should it pay attention to it? This is *Windows 95*; it knows what to do with the body of the program.) When you type 'PROGMAN' at the command prompt, *Windows 95* looks at the program, sees that it is a 32-bit executable, and runs it itself. Jerusalem never gets a chance to infect.

Conclusions

The handling of boot sector viruses by *Windows 95* is much improved from the pre-release versions previously tested; the user warning is definitely a plus. However, it would be nice if it were accurate.

File viruses are still very much under investigation – tests with a few simple non-resident file infectors produced unpredictable results. The resident file infectors tested appear to behave much as expected in the new environment, that is, pretty well, as long as the user is working within a subset of the operating system which resembles the operating system for which the virus is designed.

How well such viruses spread in the real world under *Windows 95*, where most users would prefer not to work in DOS boxes, remains to be seen. However, I remind people once again: *Windows 95* is not as far removed from DOS as *Microsoft* would have us believe. Devising viral attacks specific to the system is far from impossible.

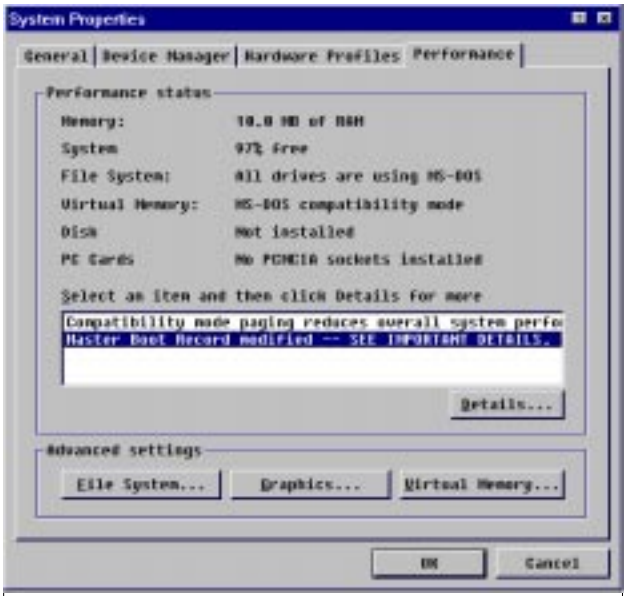


Figure 2: The messages in the system properties dialog in *Windows 95* have been changed from previous versions.

FEATURE

Computer Viruses: Naming and Classification, Part II

David B Hull, PhD
National University, California

In order to explore the classification of computer viruses [*see also part one of this article, VB September 1995 p.15*] using techniques from numerical taxonomy as applied in zoology, a collection of boot sector viruses on CD-ROM, containing a large set of viruses of all types [*Ludwig, 1994*] was obtained by the author.

As there are many duplicates of each virus on the CD-ROM, only the first of each series (*\001.BOO) was chosen. All samples which contained the name 'Stoned' were used: these were to represent the boot sector portion of the viral code.

The names supplied by *The Collection* have been retained, to enable traceability to this CD-ROM. The names were derived by running the virus scanner *F-Prot v2.08* over the samples [*Ludwig, 1995*]. In most cases, these names are very similar to the *CARO* usage. In addition to these samples of the Stoned virus, the boot sector for DOS 6.22 was added as a reference standard.

These 34 samples were converted to the hexadecimal representation of the machine code of their boot sectors, which yielded an ASCII mapping of 1024 characters. The ASCII mappings were remapped with 0 => G, 1 => H, ... 9 => P; to create a 'pseudo amino acid sequence' which would be suitable for analysis.

Methods

Analytical methods were derived from mathematical techniques used to compare protein and nucleic acid sequences [*Doolittle, 1990*]. No attempt was made to align the sequences, nor were any of several possible 'corrections' for multiple mutations or insertions and deletions applied.

A software package for the construction of evolutionary trees from nucleic acid and amino acid sequences greatly speeded up this analysis [*Peer, 1994*]. A dissimilarity matrix was calculated where S (dissimilarity) was the fraction of different pseudo amino acids between two samples. The distance matrix was then 'transformed' or standardized using the DOS boot sector as the 'reference organism' [*Peer, 1994*].

This dissimilarity matrix was then used to create an 'evolutionary' tree with cluster analysis. The weighted pair group method using arithmetic averages (WPGMA) was used, since it is relatively fast and not noticeably prone to chaining [*Jardine & Sibson, 1971*].

Results

Despite the obvious crude simplifications described above, the resulting tree is remarkably close to the *CARO* presentation and to the naming groups used in the source CD (fig. 1).

Each sample of Stoned.Standard, Stoned.June_4th, and Stoned.Michelangelo are clustered tightly together. The virus samples of the only other subgroup, Stoned.Empire, do not group strongly with each other. Fridrik Skulason notes that some viruses in the Stoned.Empire family are encrypted, which would explain this lack of linkage [*Frisk, 1995*]. All Stoned samples are clearly demarcated from the DOS boot sector sample.

Discussion

The lack of standardized names for PC viruses, and the lack of traceability to *CARO* references, means that the exact nature of the samples is somewhat murky. Indeed, the CD-ROM contains 28 samples of the virus Stoned.Michelangelo.A, all of which are classified as Michelangelo.A by *F-Prot*, but which differ in the exact hexadecimal sequence of the boot sector.

It is not clear whether these minor differences are significant or

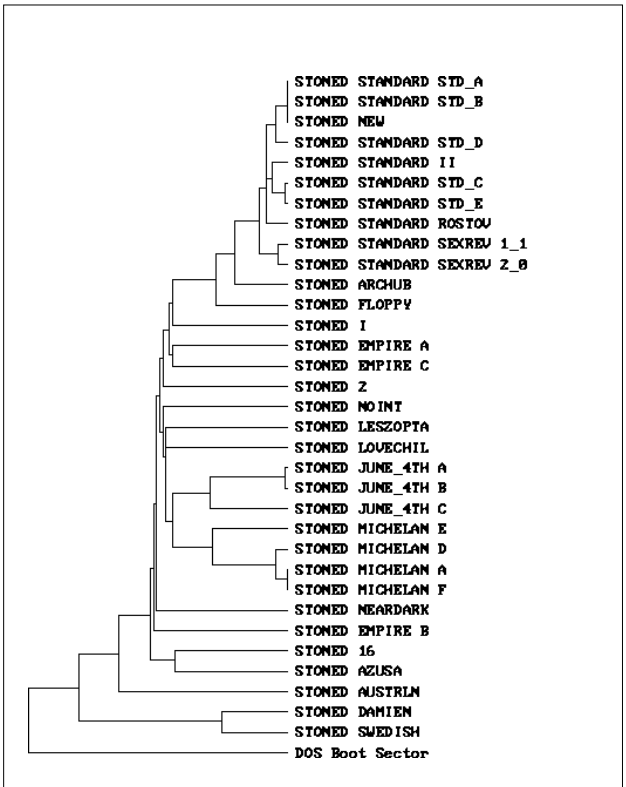


Figure 1: This represents the 'evolutionary tree' of the Stoned virus, as classified by sequence analysis.

not without disassembling the entire code to examine the functionality of the differences. It is also not clear which, if any, of these samples is the exact equivalent of the sample used by *CARO* to define the name! It seems that the gloomy prediction of an editorial in *Virus Bulletin* may be true – the CD-ROM used for this study 'could become the *de facto* test-set…' [*Ford, 1994*].

The classification presented by 'evolutionary tree' suggests interesting relationships beyond the obvious broad groupings. It would seem that there are computer virus equivalents to the biological 'quasispecies'. Stoned.Standard, Stoned.June_4th and Stoned.Michelangelo all represent the computer virus equivalent of the biological phenomenon.

The 28 samples of Stoned.Michelangelo mentioned above can be used to develop a consensus sequence for its 'wild type'. Note also that his tree represents a more sophisticated approach to representing degrees of dissimilarity among computer viruses than the simple grouping into four categories used by *CARO*.

"issues concerning … consistent naming of computer viruses still need to be resolved by the anti-virus community"

It would be interesting to develop an exact idea of the degree of difference. Two samples both classified as Stoned.Michelangelo.A differ by 1.39% in a pairwise comparison of their pseudo amino acid sequences, while the Michelangelo.A sample used in the classification study differs from the DOS boot sector by 32.1%.

It would also be interesting to know the exact amount of difference intended to be represented by the various classes used by *CARO*.

The simplistic nature of the data preparation can be much improved. For example, the first part of most boot sector viruses emulates the standard DOS boot sector with a jump over the BIOS Parameter Block. This jump instruction could be used as an offset to align the actual viral code.

There are also a number of sophisticated amino acid sequence alignment packages in use for molecular biology which could be applied [*Doolittle, 1990*]. Furthermore, many of the 512 bytes in these boot sector viruses are either empty space or non-functional. Isolating and comparing active virus code should certainly result in improving the classification.

Conclusions

Classification techniques from molecular biology appear to provide a powerful tool for classifying computer viruses. Simple conversion of the machine code into pseudo amino acid sequences and analysis by pairwise comparisons produce useful dissimilarity matrices which can be clustered using standard techniques.

The resulting 'evolutionary trees' provide meaningful insight into the types and degree of relationship among the computer viruses which were sampled. Considerable development of these techniques is available in the literature of molecular biology.

Basic issues concerning traceable and consistent naming of computer viruses still need to be resolved by the anti-virus community; in particular, those of valid names, and linking valid names to type specimens.

Essential issues on the basic units to be classified, and problems about the interpretation of 'parallel' code structures and encryption, still need to be examined. However, it would seem that there are 'quasispecies' of computer viruses. The artificial life community might want to explore this analogy further.

The *CARO* classification scheme certainly appears to be 'natural' or 'information rich', which is what one would expect from a system truly reflecting reality [*Jardine & Sibson, 1971*]. This should lend support for its use as a standard for naming computer viruses.

Appendix

CARO Classification of a Computer Virus [*CARO, 1991*]

'The full name of a virus consists of up to four parts, desimited (sic) by points ('.').'

The general format is:
Family_name.Group_name.Major_Variant.Minor_Variant

1. Family Names.
   The Family_Name represents the family to which the virus belongs. Every attempt is made to group the existing viruses into families, depending on the structural similarities of the virus…
2. Group Names.
   The Group_Name represents a major group of similar viruses in a virus family, something like a sub-family … (or) distinguished clone…
3. Major Variant Name.
   The Major_Variant name is used to group viruses in a Group_Name, which are very similar, and usually have one and the same infective length…
4. Minor Variant Name.
   Minor_Variants are viruses with the same infective length, with similar structure and behaviour, but slightly different. Usually, the minor variants are different patches of one and the same virus.

[*The first instalment of this article, along with a bibliography, can be found in VB, September 1995, p.15. David Hull can be contacted through email at: dhull@nunic.nu.edu.*]

PRODUCT REVIEW 1

IBM AntiVirus for NetWare

Jonathan Burchell

IBM AntiVirus for Netware v2.2 (*IBMAVN*) is the file server side of *IBM*'s anti-virus solution. We first reviewed it in October 1994: since then, many features have been added, and the signature database has been improved.

In choosing any anti-virus solution, both server and workstation components must be considered. *IBM* has done much to enhance and improve the workstation protection of their product, including adding a new DOSShield component. As these were reviewed only last month [*see VB, August 1995 p.21*], it seems appropriate to restrict this review to concentrating on the *Novell NetWare* side of the product.

Presentation and Installation

The *NetWare* component of *IBMAVN* ships in two forms: a ZIP file on a single 3.5-inch high-density diskette, or already-expanded files on two diskettes. The product fully supports *NetWare* v3.1x and v4.x. It also supports the use of *OS/2* and *Macintosh* name spaces on the file server. The documentation consists of an extremely comprehensive administrator's manual, bound in an A4-size ring binder.

No installation software is provided; the files must be manually copied to a suitable directory on the server (*IBM* suggests SYS:\SYSTEM\IBMAVN, but in fact any directory can be used). The product actually contains three separate versions of the NLM, due to the current confusing situation surrounding the capabilities of different *NetWare* versions.

Depending on which versions of CLIB (and a few related NLMs) are present, your server may or may not be capable of supporting *IBM*'s anti-virus sophisticated file access and scanning features.

However, all is not lost if your server is out of date – *IBM* ships an executable from *Novell*, capable of updating a 3.x or 4.x server. This update procedure is largely automatic and self-determining. *IBM* is to be congratulated for shipping the update package rather than taking the usual stance of 'Contact *Novell* for the latest updates'. In early versions of the product, it was necessary to determine which version of the NLM to load, based on current server NLM revision level. I am glad to report that *IBMAVN* now detects automatically the correct version of the NLM to run at load time.

Product Highlights

IBMAVN supports real-time, scheduled, and on-demand scanning of *NetWare* servers. Real-time checking includes the ability to check files as they are written to the server, read from the server, renamed, or erased.

The ability to check files on rename means that you might decide to remove checking of files on being read (which in any server product always has some impact on performance) and instead rely on checking when files are written to the server and/or renamed. Without rename support, this is a totally different policy, as an infected file could be written to the server as VIRUS.TXT, then renamed to VIRUS.EXE.

One of the more sophisticated features of *IBMAVN* is the way in which it provides access to files at all times. It is a problem with all background-scanning products that they may be scanning a file at the same moment a user requests exclusive access to the file. With many products, the user would be denied access and receive an error message from the server; however, *AVN* can spot that this situation is about to occur and relinquish file control to the user. In this case, *AVN* simply checks the file when the user releases it.

'Full-time access', as *IBM* terms this system, is used for both real-time and scheduled scanning. In fact, real-time scanning works a little differently under *IBMAVN* from most other products, in that files are immediately allowed on or off the server, but the file name is placed in a queue for 'as soon as possible' background scanning.

This ensures minimal real-time overhead, but does introduce conceptual problems: for instance, is it possible to copy a file onto and off a heavily-loaded server without it being checked at all? Such an issue is difficult to check in reality; however, *IBM* assures us that scanning priorities can be adjusted to keep the number of files in the queue small.

Virus alerts can be delivered by any combination of three messaging paths: standard *NetWare* broadcasts, email (via a *Novell* MHS-compatible system), and *IBM*'s own *NetWare*-compatible messaging systems.

These last are delivered over the network direct to copies of the Alert program running on *OS/2* or *Windows* workstations. They have the advantage, in comparison with standard *NetWare* broadcast messages, of being persistent – that is, if the designated recipient is not logged in, the software will continue trying to contact them until it succeeds.

Configuration and Administration

IBMAVN is administered via the console interface (or via RCONSOLE): no tools are provided for remote administration. The alert program can display a summary list of servers, and current scanning/protection configuration and version data – this, however, hardly qualifies as a remote administration tool.

The product's user interface is, as I commented in a previous *Virus Bulletin* review [*see VB October 1994, p.17*], distinctly 'big blue' in flavour. It does not follow the usual *NetWare* character-based menuing interface (based, ironically, on an *IBM*

standard); instead, the screen is divided into two: a status display of the current configuration and a prompt line at which product configuration is changed.

Parameters are altered by entering keywords and values. There is a limited on-line help system, but for the more complex options, consulting the printed manual is the best option. Everything entered at this pseudo-prompt can be placed on the command-line when starting the NLM, or in a file called IBMAVN.PRF for preconfigured operation.

An idea of the idiosyncratic feel of this user interface can be gained from examining the way we would set up scanning all incoming files with extensions EXE or COM. With most products, we would navigate through a few menus and dialog boxes, tick some options, and that would be that. With *IBM AVN* we enter the following command, which many users might feel is perhaps a little less than obvious:

```
-Check incoming - Match *.exe *.com,
```

Scheduled Scans

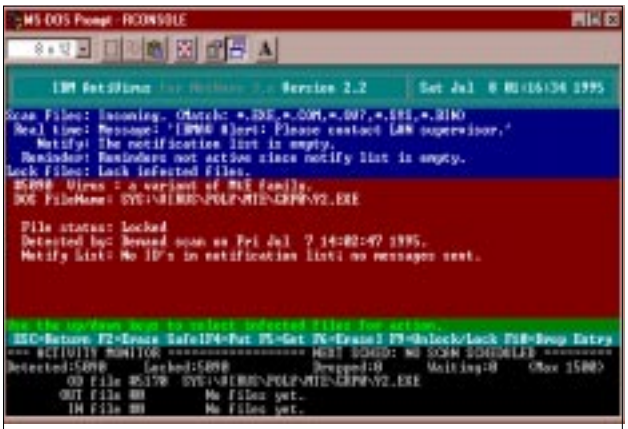
The general format of a scheduled scan command is:

```
-SCHED <Number of Scan> <period> <date>
<time> FOR <time> <paths> <files>
```

Five separate scans may be specified, and each scan job may be set to occur at preset times. The options are: once, every five minutes, daily, weekly, Saturdays, Sundays, weekdays, or weekends.

The date and time options indicate date/time to start the scan. The time after the 'FOR' option gives the maximum permissible scan time: scans which take longer than expected can thus be prematurely and automatically terminated before the next period of heavy server usage begins. The directories and files to be scanned may be specified, and wildcards may also be used. Sub-directories are automatically included. Overlapping scheduled scans are not allowed: the product automatically checks for such clashes when a job is defined.

A scan which is terminated because it runs out of time is noted in the log file as having been prematurely terminated. The next time this scheduled job starts, the scan will start from the point at which it had terminated, not at the beginning. A slightly modified version of the SCHED command allows for a single scan to be specified that is automatically started whenever the server is rebooted or restarted.



IBM AntiVirus is able to lock infected files to prevent them being accessed.

On-demand scanning is started directly from the keyboard (via the F7 key). A prompt allows specification of which volumes, paths and file types are to be scanned.

Other Features

The scanner will automatically decompress PKZIP, self-extracting PKZIP (SFX) and LZEXE files. These files are decompressed to a specified directory before checking: it is possible to specify both which types of scan compressed files should be decompressed (ON-DEMAND, SCHEDULED, INcoming, OUTgoing) and what types of file extension should be checked for possible 'compressedness'.

Real-time scanning requires that your file server be equipped with CLIB version 3.12f or later. Files may be checked as they are written to the server and/or read from it. Files with the extensions *.EXE, *.COM, *.OV? and *.SYS are scanned by default. This list may be modified, and/or extended, and specific volumes and or paths may be specified.

The software is highly configurable – there are options and combinations too numerous to cover here. The big advantage of not using a menuing or GUI interface is that most of the options can be combined as desired to create customized requests in ways that may never have been thought of had the user been restricted to a 'canned interface'.

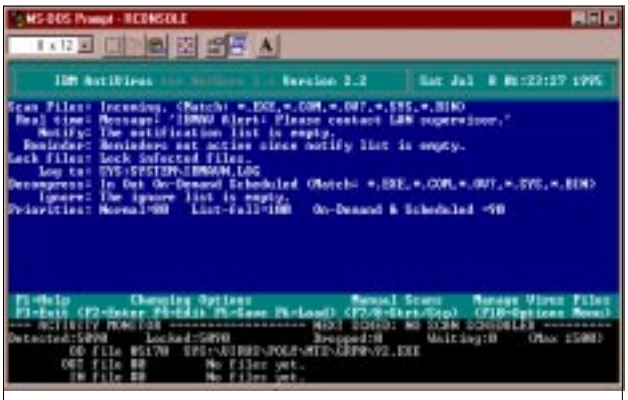
The disadvantage is that, as the complexity of the product grows, and the number of its features increases, achieving a result which is different from the standard defaults requires a serious investment in understanding the product and its configuration.

All in all, I suspect that this outweighs the marginal advantages of complete configurability, and that many potential users will be put off by not seeing a friendly interface – that at present, in this product, is light years away from current trend towards answer wizards and on-line help systems.

Virus Detection

On detecting a virus, a number of things happen:

- messages are sent via a specified combination of *Novell* broadcasts, email and *IBM AV*-specific *NetWare* messages to the nominated list of users
- the file is locked to prevent further access by any user or by an NLM
- the file is added to a list of infected files to await further processing



IBM AntiVirus' user interface is unconventional, but allows the product to be customised to a greater extent than most.

No options exist to delete the file automatically, rename it, or move it to a quarantine directory. The list of infected files must be further processed by an operator via the console interface. The incident screen for each infection is viewed, and a number of actions can be taken, including unlocking the file, deleting the file, and making a copy of the file (so as to try file disinfectors – such as that built into *IBM's* workstation product – on it).

Each incident in the log must be individually processed. Apart from an option to erase all log file entries, but not their associated files, automatically, no global processing options exist. *IBM* argues that, in the real world, infections will occur in only a few files, and should be dealt with individually.

The Log File

The software maintains a log file of major events and detections. Options are available to keep the log file pruned to a sensible size. No facilities exist to filter or scan the file: this must be done using external editors or viewers.

The format of a log file is well documented in the manual. It would also be feasible to build a file parser to extract important information directly into a database from the supplied log file.

Results

As can be seen from the results table, detection rates are extremely good. The results obtained place it firmly in the top three or four products in terms of detection capability. Missing five of the 'In the Wild' samples is unacceptable, however, and should be corrected immediately.

The polymorphic detectors are very good, although they inexplicably missed about 0.1 percent of the Girafe and Pathogen infections. Detection of the Cruncher samples is completely absent; I suspect because *IBMAVN* does not try to decompress the DIET-compressed files which Cruncher produces – I would have thought it a relatively simple matter to add this feature. [*IBM states that the current shipping version of the product (2.3) does handle this virus. Ed.*]

One advantage of the way the product queues files accessed in real-time for later scanning is that exactly the same scanner is used for both on-access and background processes. Thus, results for real-time detection are identical to those for the background scanner.

IBM Anti-Virus for NetWare is a product that is rapidly evolving: each of the four versions we have seen in the past 18 months has had greatly improved performance and features. This suggests that *IBM* is very serious about the virus detection game, and is investing resources into the product to produce a competitive solution.

IBMAVN lacks such features as the ability to administer groups of servers as a single domain – this may limit its suitability for large multi-server sites. Additionally, it has a 'unique' user interface.

On the other hand, what it does, it does extremely well. Given the combination of the excellence of the DOS scanner detection [*see VB September 1995, p.21*] and the current results, *IBMAVN* must rate highly as a potential choice (together with only one or two other products) for protection of your network.

IBM Anti-Virus for NetWare

Detection Results

Main Scanner:

Standard Test-Set[1]	229/230	99.6%
In the Wild Test-Set[1]	121/126	96.0%
Polymorphic Test-Set[1]	4741/4796	98.9%

NB: For detection results on the DOS scanner, see the in-depth review of that product on p.21 of the September 1995 edition of *Virus Bulletin*.

Technical Details

Product: *IBM AntiVirus v2.2.*

Developer/Vendor (UK): *IBM UK* , Normandy House, Alencon Link, Basingstoke, Hants, RG21 1EJ. Tel 01256 314558, fax 01256 332319.

Developer/Vendor (USA): *IBM Corporation*, Long Meadow Road, Sterling Forest, NY 10979-0700. Tel +1 914 759 2901, fax +1 914 784 6054. Note also that *IBM* provides support for its *AntiVirus* program through its usual outlets in almost every country in the world. The documentation contains a voluminous list of contact addresses and telephone numbers.

Price: 1-250 users, £1000; 251-500, £2000; 501-1000, £4000; 1001-2000, £6500; 2001-3000, £9500; 3001-5000, £12,500; 5000+ on application only. Includes quarterly updates.

Hardware used: Client machine – 33 MHz 486, 200 Mbyte IDE drive, 16 Mbytes RAM. File server – 33 MHz 486, EISA bus, 32-bit caching disk controller, *NetWare 3.11*, 16 Mbytes RAM.

[1] **Test-sets:** Each test-set contains genuine infections (in both COM and EXE format where appropriate). For details of the Standard test-set, see *VB*, January 1994, p.19 (file infectors only). For details of In the Wild and Polymorphic test-sets, see *VB*, August 1995 p.19.

PRODUCT REVIEW 2

Norton Utilities

Dr Keith Jackson

The *Norton Utilities* is not a product that can be reviewed by *Virus Bulletin* in its usual manner. The package offers no explicit anti-virus features, but does provide a wealth of features which are very useful, nay essential, when a virus infection is found.

The package has been around for a long time; indeed, I cannot remember ever having used a PC without a copy of *Norton* close at hand. This month's review looks at the latest release of the product for *Windows* (version 8), specifically with regard to anti-virus features and, in general, at its various computer security features.

If you have ever had a need to reclaim an erased file, dig around inside *MS-DOS* or *Windows* setup files, completely erase the contents of a disk or a file, hide files, inspect hidden files, or generally do things for which the programs supplied as standard with *MS-DOS* and *Windows* are close on useless, then one of the best starting-points is the *Norton Utilities* – a collection of integrated programs which provides just such features.

Documentation

The documentation comprises a voluminous A5 book, which describes the available features in a reasonable level of detail. It is very readable, and thoroughly indexed – all in all, quite a reasonable effort.

Previous versions of the *Norton Utilities* that I have seen came with various volumes of documentation: it is definitely a great improvement to see that everything has been combined into a single volume.

Installation

Version 8 of *Norton Utilities* was provided on four 3.5-inch, high-density (1.44 MByte) floppy disks. The user is warned not to proceed with installation to a hard disk if there are files on the disk which need to be unerased. After choosing between full and custom installation, installation is simply a matter of inserting the floppy disks in sequence as requested. Full installation requires 9 MB hard disk space.

My only problem occurred when the installation program warned that it could not add the location of the *Norton Utilities* subdirectory to my *MS-DOS* PATH. It produced the onscreen error message 'DOS PATH is too long'.

The installation program offers to alter various system start-up files immediately, or to save the changes on disk for later manual alteration. A Rescue Disk may also be created: this will contain the 'get-out-of-a-hole' portions of *Norton Utilities*, and could well be useful at some point later if things go awry.

Testing and Recovery Features

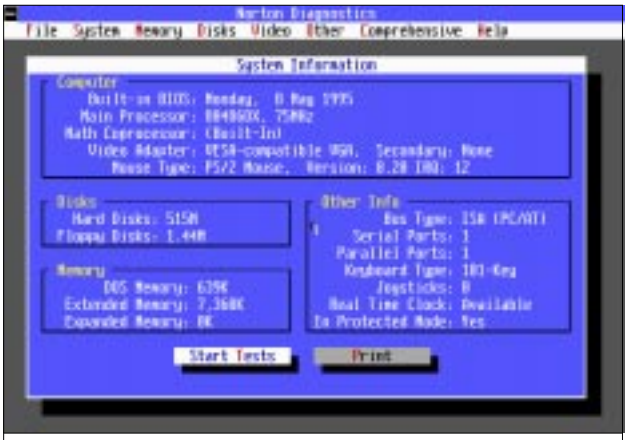
Norton Diagnostics (NDIAGS) tests the hardware components of a PC. The specific tests offered by NDIAGS can be used to test the CPU, various hardware controller chips, the real-time clock, memory (low, expanded and extended), the serial port(s), the parallel port(s), the CMOS memory, interrupt configuration, floppy and/or hard disks, the video memory, the keyboard…

Everything. Even the PC speaker is included; in fact, a few advertising phrases are reproduced through this! Individual tests can be executed in sequence or repetitively.

Norton Disk Doctor (NDD) can provide thorough tests on all aspects of a disk drive. It is possible to test the integrity of a disk at both the logical and the physical level. Tests can be set up which will check the disk surface repeatedly in the hope of identifying intermittent problems.

Probably the most useful features of the Disk Doctor are those that can make a disk bootable (even where DOS reports that there is no room for the system files), and those that can revive a faulty disk, making sure that the data originally on the disk is retained even though a new format pattern has been inserted.

It is always worth trying to correct a disk problem using NDD before calling in a highly-priced consultant to fix the problem. This of course assumes that you are confident that you know enough to avoid compounding the problem by taking such a course of action. I have used NDD in anger in the past, and can vouch for its capabilities.



Norton Diagnostics can be used to check every part of the computer's hardware, offering specific tests for each component, right down to the speakers.

Another plus of the NDD is that it will always ask for permission before it makes any alteration to a disk. I found NDD very hard to test, as, if a floppy disk ever shows an error, I immediately retrieve the files and throw the disk away. Likewise, my stock of damaged hard disks is somewhat small. Damaging one's own hard disk is a tad beyond the normal call of duty for a *VB* reviewer.

DISKEDIT can be used to view/edit the entire contents of any type of disk (floppy or hard), including disks DOS cannot recognise or access. Access at any level down to individual sectors is available, and DISKEDIT can attempt to repair/rescue any type of damaged file or absolute sector.

By default, DISKEDIT starts execution in read-only mode, making it impossible to blunder straight in and make disastrous low-level changes. The default can be altered when alterations are really necessary. Disk information can be viewed in a variety of ways, depending on what the information represents (file content, FAT, Partition Table, boot record etc) – this varies from individual sectors up to a map of an entire drive.

If you have a disk drive which NDD cannot fix automatically, perhaps one where the content of the disk has been ravaged by one of the more destructive viruses, then DISKEDIT is the means by which manual alterations can be made. Having such power available necessarily involves using it responsibly.

The *Norton Utilities* even provides specific utilities called 'Disk Tools' and 'File Fix' which can perform such tasks as making a disk bootable, repairing a particular type of disk file (e.g. a *Lotus 1-2-3* spreadsheet file), and marking a particular cluster on a disk. Such actions could be carried out manually using DISKEDIT, but they are easily automated, and then less prone to error.

Rescue utilities are provided which ensure that accidentally formatted disks can be recovered, which keep track of all of the current *Windows* INI files, and which can create a rescue disk for use if the hard disk dies at some future date.

UNERASE and UNFORMAT are self-explanatory components of the *Norton Utilities*, both of which I have used at various times in the past. UNERASE relies on the fact that *MS-DOS* does not actually remove the content of an erased file; it merely marks the FAT entry as de-allocated.

UNFORMAT relies on the *Norton* SAFE FORMAT utility being used to format the disk in the first place such that although the disk appears empty, the actual content of each track is still present and, if the file directory structure is reimposed, files can magically reappear.

Security Features

DISKMON is a utility that can prevent data from being written to disk. It operates in the background, and can be set up to protect system areas, files, or the entire disk. This is a most useful utility as far as viral activity is concerned, as it is a memory-resident program which prevents unwanted disk access by *all* programs, including any virus that may also be memory-resident (and active).

DISKMON can show which disk is being accessed by a flashing letter corresponding to the disk drive in the top right-hand corner of the screen. This may be too irritating to be used on a routine basis, but could well prove useful under certain circumstances (such as when a virus is active).

The documentation claims that DISKMON uses about 8 KB of memory, but I measured this as 9.4KB (though all of this occupied expanded memory). Isn't it strange how developers of memory-resident programs *always* minimise the amount of memory such programs use? Users deserve better. DISKMON captures interrupts 13h, 21h, 25h, 26h, and 2Fh. This is such an extensive list that it may well cause problems with other memory-resident programs which try to capture the same interrupts.

> *"tests can be set up which will check the disk surface repeatedly in the hope of identifying intermittent problems"*

Another component, DISKREET, can encrypt and password-protect files. It can also be used to create password-protected logical disk drives on which all files are encrypted. This may be a useful security feature under certain circumstances, where it is necessary to keep tight control over a specific set of files.

Files can be encrypted/decrypted singly in any desired manner: two different methods of encryption algorithms are discussed in the *Norton Utilities* manual – one is proprietary; the other is the Data Encryption Standard (DES). The DES option was not available in the copy provided for review, presumably to avoid contravening US export regulations. No details are provided about the proprietary encryption algorithm, therefore I cannot comment on its strength (or lack thereof).

On my test PC, DISKREET was capable of encrypting (and decrypting) a 330 KByte text file in 15.1 seconds. For reasons which are beyond me, encrypted files were always 6 KB larger than their plaintext equivalent.

A password is required for each encryption and/or decryption (passwords can be specified as a command-line parameter), and, after a file has been encrypted, the user is asked whether the original plaintext file should be deleted. Files can be repetitively encrypted: *Norton Utilities* does not seem to mind that a file is already encrypted; however, it will not encrypt a file into a file of the same name.

The final security utility is called WIPEINFO. This program can be used to overwrite single files, entire disks, or erased data, so that they cannot be recovered by any means – not even by the

Version 8 of the *Norton Utilities* contains many separate components, any one of which could prove to be invaluable in times of crisis.

Norton Utilities! This single utility combines the features of two separate programs (WIPEDISK and WIPEFILE) provided with previous versions of the product.

Speed Components

Although the *Norton Utilities* provides features which can speed up use of a PC, such programs should be used with great care. Even though they are well-proven, and seem to be thoroughly tested, a complete backup is well advised before any experimentation is performed. The manual advises users to ensure that they have a backup, and an onscreen message reinforces this point before execution is allowed to commence. You have been warned!

The *Norton Utilities* can tune the way in which sectors on a hard disk are interleaved, so that access to disk data takes place in the most efficient manner. The program that does this (CALIBRAT) alters the sector interleave, and provides clear graphs (in the form of bar charts) to show what the effect of its actions will be.

In similar fashion, a defragmenter program (SPEEDISK) is provided, which can reorganise the location of individual sectors of a hard disk so that all sectors of each file are stored consecutively on the disk, and all directory elements are gathered together. I have used this program for many years on a regular basis without any ill-effects, and with a beneficial effect on data access.

SPEEDISK now has knowledge of compressed volumes (such as those controlled by *Stacker*). When such disk drives are processed, the underlying hard disk is first defragmented, then the *Stacker* disk itself is defragmented.

The Rest

Version 8 of the *Norton Utilities* includes several *Windows*-specific utilities. I especially like the *Windows* version of *Norton* Disk Doctor (NDD); it's the first program I have reviewed for *VB* which made me laugh. When a disk is examined by NDD, a small picture of a disk revolves, and a man in a white coat examines his papers whilst the file directory structure is being examined, and runs a magnifying glass over the disk surface whilst the disk itself is searched for problems. Brilliant!

Four separate *Windows* programs are provided which can manipulate/monitor/tune the various *Windows* INI files. A *Windows* version of SPEEDISK is included (offering the same features as the DOS version – but prettier!). The final *Windows* utility (System Watch) allows a user to monitor Windows memory usage, CPU utilisation, open files etc. In fact, all the resources over which *Microsoft Windows* should offer fine control. But doesn't.

I have run out of space to describe, let alone examine, the myriad components of the *Norton Utilities*. Still to be mentioned are a disk cache which claims to be better than most, a program to extend *MS-DOS* batch file language, many utilities that examine, manipulate and/or locate files, a floppy disk formatting program more flexible (and faster) than *MS-DOS* FORMAT. I could go on, but I've probably done enough to give some idea of the range of facilities provided by the latest version of the *Norton Utilities*.

Conclusions

As the above text makes amply clear, I am a confirmed user of the *Norton Utilities*, so any claims to showing no bias in this review are fatuous. I have used the *Norton Utilities* since 1987: it is a mature stable product, and I have no real complaints about the package.

I've tried most of the competition, reviewed the products, tried them out as shareware, I've even paid money for some of them! Yet I keep coming back to the *Norton Utilities*: it offers a suite of programs which are very easy to use, and several features which could one day prove indispensable.

Version 8 is an advance on the previous version by offering *Windows*-specific features, but I do grow somewhat weary of the amount of disk space which is required – 9 MBytes, for God's sake!

Technical Details

Product: *Norton Utilities version 8* (No serial number visible).

Developer/Vendor (US): *Symantec Corporation*, 10201 Torre Avenue, Cupertino, CA 95014-2132, USA, Tel +1 310 449 4900, fax +1 310 829 0247.

Support outside US:
UK – Tel +44 1628 592222, fax +44 1628 592393.
Australia – Tel +61 2 879 6577, fax +61 2 879 6805.

Compatibility: 286 or higher PC, PS/2 or compatible, DOS 3.3 or later, *Windows 3.1*, 1 MB RAM, and VGA card (or better). Use of mouse recommended. Network options are also available, allowing network diagnostics to be carried out, evaluation of nodes attached to the LAN, and dissemination of the product throughout the network.

Price: £129.00 (RRP) for full version; £49.00 for upgrade.

Hardware used: A *Toshiba 4400C* laptop computer containing a 25 MHz 486, one 3.5-inch (1.4 MByte) floppy disk drive, a 120 MB hard disk and 12 MBytes RAM, using *MS-DOS v5.00*, *Windows v3.1* and *Stacker v2*.

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 editorial@virusbtn.com
CompuServe address: 100070,1340

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 ***22nd Annual Computer Security Conference and Exhibition*** will be held in Washington, DC from 6-8 November 1995. The conference will feature over 120 sessions on various topics. Further information is available from the *Computer Security Institute* on Tel +1 415 905 2626, fax +1 415 905 2626.

Infosec, the UK's first dedicated information security show, will be held at the *London Olympia* (London, UK) from 30 April–2 May 1996. It is planned that **the programme will include conferences and seminars on topical security issues**. Information on attending or exhibiting is available from *Infosec* on Tel +44 181 910 7821.

The next round of **anti-virus workshops presented by *Sophos plc*** will be held at their training suite in Abingdon, UK, on 22/23 November 1995. Cost for the two-day seminar is £595 + VAT. Any one day (day one: Introduction to Computer Viruses; day two: Advanced Computer Viruses) can be attended at a cost of £325 + VAT. Contact Julia Line on Tel +44 1235 544028, fax +44 1235 559935, for details.

A three-day security and audit practitioner's guide, *Cruising the Internet Securely*, will be held from 6–8 November 1995 at Grosvenor House in London, UK. This will be followed, at the same venue, by an **intensive two-day workshop on *Firewalls and Internet Security***. The courses are to be presented by the *MIS Training Institute*, in association with *Euromoney Publications PLC*. Contact Louise Thomson on Tel +44 171 779 8795, fax +44 171 779 8944 to book.

MIMEsweeper*, anti-virus software for *cc:Mail, was launched by *Integralis* at *NetWorld* (Paris, France and Atlanta, GA, USA) in September. The product claims to give email administrators a way to examine incoming and outgoing messages for viruses automatically. Further information on the product is available from David Guyatt at *Integralis*; Tel +44 1734 306060, email d_guyatt@integralis.co.uk.

***Compsec 95* will take place in London**, UK, from 25-27 October 1995. For details on the conference, contact Sharron Emsley at *Elsevier Advanced Technology* on Tel +44 1865 843721, fax +44 1865 843958, email s.emsley@elsevier.co.uk.

***S&S International* has announced the appointment of two more virus researchers** to its team: Igor Muttik has joined the corporation's UK staff from the Physics Department of *Moscow State University*, and Glenn Jordan has taken up the post of Senior Technology Consultant at the US offices.

The *British Standards Institution* has produced **an electronic version of BS 7799**, the code of practice for information security management, on floppy disk. The package includes *BSI Electronic Products Helpdesk* support, and is available at £95 to *BSI* members. To order, Tel +44 181 996 7333, fax +44 181 996 7047.

Proceedings of the *Fifth Annual Virus Bulletin Conference*, *VB 95*, are now available from VB offices. The price is £50 + airmail p&p (England £7, Europe £15, elsewhere £25). To order, contact Petra Duffield or Dale Tabrum at the *VB* Conference Department; Tel +44 1235 555139, fax +44 1235 531889.

Reflex Magnetics is holding a **three-day course on hacking methods and techniques**; offering hands-on experience of hacking tools and access to hacker bulletin boards. Cost for the seminar is £945 + VAT, and further information can be obtained from Rae Sutton at *Reflex*; Tel +44 171 372 6666, fax +44 171 372 2507.

On 9/10 October and 13/14 November 1995, *S&S International* is presenting further **Live Virus Workshops** in Buckinghamshire, UK. The two-day course costs £680 + VAT, and offers the opportunity to gain experience with viruses within a secure environment. Contact the company for details: Tel. +44 1296 318700, fax +44 1296 318777.