ISSN 0956-9979 MARCH 1999

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **CaNT or woNT?** March's comparative review tests the latest products for *NT* workstation. Keep up with the changes, starting on p.12.

• **The main feature:** *Data Fellows'* Katrin Tocheva paints the big picture of the evolution of VBS/VBA viruses in our feature article which starts on p.6.

• **Guest appearance:** *Virus Bulletin's* Technical Editor Jakub Kaminski is not optimistic about the future, including the Millennium Bug in this month's Editorial on p.2.

CONTENTS

GUEST EDITORIAL

Don't Let the Bugs Byte!

Recently, I had the opportunity to witness Wayne Gretzky scoring another record goal at Madison Square Garden (for those very few who don't know what I'm talking about – I mean ice hockey). Just before the game, we went to have a quick bite at the local Club Bar & Grill. The conclusion of the dinner was really unexpected and hysterically funny. The credit card we were trying to pay our bill with was brought back by the manager who asked if we had another card, since that one had just been rejected. The system said that the card's expiry date was invalid.

"Attitudes and approaches to the Millennium Bug tend to vary…"

After a quick glance at the card all of us burst into uncontrollable laughter. The year printed on the card in question read, you guessed it, 00! It looks like the Y2K bug struck ten months earlier than the predicted D-day. In our case, there was no disaster – the old-fashioned, carbon-copy card print was taken and nothing spoiled a great night out.

Significantly, what appeared to us (all involved in software development) to be hilarious, was of serious concern to the businessman running the restaurant. Attitudes and approaches towards the so-called Millennium Bug tend to vary from one professional, social, economic or political group to another, depending on group interests and individual beliefs.

While financial institutions spend big money trying to find and fix all the bugs, a lot of businesses try to 'solve' the problem by shifting the responsibility onto all their partners and suppliers (by collecting year 2000 compliancy forms in case things go wrong). Many ordinary citizens will stock up on food and fuel and move to shelters far away from the big smoke in order to survive the predicted disaster. The FBI has applied for extra funding to deal with groups which expect the end of civilization, and which, after the likely disappointment, will try to dismantle it on their own.

Obviously, the media concentrates on the sensational aspects of the problem. The most 'prophetic' prediction I've heard so far is that 'people will die' – I'd personally like to extend this prediction to the years 2001, 2002 and 2003!

Differences in ways the Y2K bug is tackled seem to have a cultural basis. Americans are amused by stories of Chinese airline executives who have been ordered to spend New Year's day aboard one of their own aircraft. At the same time, the US Senate Commerce Committee discusses legislation encouraging the fixing of Y2K faults by 'granting limited immunity from lawsuits to companies that make good-faith efforts to avert so-called Y2K problems' (they didn't define the term 'good-faith efforts'). The proposed bill 'would cap non-economic damages and bar punitive damages unless there's a showing of extreme negligence' (another undefined term).

A few governments have already decided to print extra cash in order to withstand the huge demand expected at the end of this year. Millions of customers are expected to follow the advice of survival books, brochures and manuals, rushing to banks and ATMs to get at their savings. I don't consider any specific country to be extremely bad or extremely original in the way they address the Y2K issue. On my way home from the airport I noticed a huge new billboard – it depicted a factory worker wriggling in mid-air squashed in the arm of an industrial robot. The accompanying message read 'Don't let Y2K byte your workers' followed by the standard Australian government slogan 'Think it. Talk it. Work it.'. It's a pity that whoever came up with such a silly design didn't do *their* thinking first.

The global problem of the year 2000 can easily be compared to one facing the anti-virus business in particular and the computer security industry in general; the number of educated users constantly grows, but the number of new, more or less uneducated users grows much faster. Which do you think will grow faster in the next ten years, numbers of those having access to computers or numbers of those knowing how they work? The future looks gloomy enough, even without the troublesome start of the next millennium.

Jakub Kaminski

NEWS

Not Mentioning Any Names

Conscience and goodwill make good PR. UK-based *Portcullis Computer Security Ltd* is the latest company to feel compelled to offer a free solution to users currently experiencing 'problems' with rival software.

According to a recent, mysteriously-worded *Portcullis* press release, 'Now that certain suppliers' legacy AV systems are damaging – sometimes destroying – documents found to have viruses, Portcullis is concerned to protect those users for the three months that it expects the problems to continue.' After that? Who knows.

Defuse Enterprise (see *VB*, July 1998, p.18), an heuristic analysis and protection system for *Word* macro malware, is available from http://www.portcullis-security.com/ ▮

Bogus or Blustering?

The Sunday Telegraph (UK) of 7 February carried a feature about Nir Zigdon, a 14 year-old Israeli boy who had, allegedly, written a 'computer virus and sent it in an email' to an Iraqi government Internet site. The site was 'destroyed' when the Iraqis, believing Nir to be an anti-Israeli Palestinian virus writer, opened the message and clicked on the designated batch file. *Virus Bulletin* was intrigued and tried to follow up the story.

It transpired that the report was full of inconsistencies and half-truths. An Israeli source for *Virus Bulletin* tells the story rather differently. For starters, Nir Zigdon's so-called 'virus' was obviously a Trojan Horse. Nir himself admitted that the Trojan was simply a batch file with four lines of code in DOS format.

Less dramatic too, was the reaction of the site manager, who only realized Nir was an 'imposter' after Israeli media coverage had blown the whistle. More importantly, the whole episode begs the question – why was such an 'official' site, designed by an image-conscious regime, not immediately restored from backups?

Most disconcertingly, this boy has appeared on national television and is hailed a hero of his people. While proclaiming Israel second only to California's silicon valley and vaunting its technical sophistication in the field of computing and computer security, this kind of media coverage lionizes a 14 year-old child, who claims to have written his first virus at ten years old, as one of a 'new generation of Israeli computer protégés.'

What kind of message is Israeli youth receiving – for all his moral good intentions, what really separates Nir Zigdon from the US Pentagon, NASA and Navy Research Center hacker, fellow Israeli, Ehud Tenenbaum? ▮

Prevalence Table – January 1999

Virus	Type	Incidents	Reports
ColdApe	Macro	438	21.9%
Class	Macro	318	15.9%
Cap	Macro	226	11.3%
Laroux	Macro	191	9.5%
Temple	Macro	105	5.2%
Form	Boot	69	3.4%
Concept	Macro	58	2.9%
CIH	File	49	2.4%
Npad	Macro	43	2.1%
Appder	Macro	30	1.5%
Ethan	Macro	30	1.5%
Parity_Boot	Boot	29	1.4%
AntiEXE	Boot	25	1.2%
NOP	Macro	25	1.2%
Munch	Macro	21	1.0%
Sampo	Boot	16	0.8%
Stat	Boot	16	0.8%
Eco	Boot	15	0.7%
Groov	Macro	14	0.7%
Showoff	Macro	14	0.7%
Chack	Macro	12	0.6%
Empire.Monkey	Boot	12	0.6%
Jumper	Boot	11	0.5%
Suck	Macro	11	0.5%
Copycap	Macro	10	0.5%
Win32/Ska	File	10	0.5%
AntiCMOS	Boot	9	0.4%
Kenya	Boot	9	0.4%
Wazzu	Macro	9	0.4%
Brenda	Macro	8	0.4%
Nono	Macro	8	0.4%
Hark	Macro	6	0.3%
HLLP.DeTroie	File	6	0.3%
Kompu	Macro	6	0.3%
Mental	Macro	6	0.3%
Extras	Macro	5	0.2%
Junkie	Multi-partite	5	0.2%
Ripper	Boot	5	0.2%
Stoned.Angelina	Boot	5	0.2%
Others [1]		119	7.6%
Total		2004	100%

[1] The Prevalence Table includes a total of 119 reports across 73 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

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 15 February 1999. 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 with 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

Acid.603

CER: An appending, 603-byte virus. Infected files have their time-stamps set to 62 seconds and the word 0ABCDh at offset 0010h (in EXE files).

```
Acid.603        B440 B95B 0299 CD21 B800 422B C9CD 21B4 40B9 1A00 BA67 02CD
```

Adrenalin.552

CR: A 552-byte appender with the encrypted text 'ADRENALIN OVERDOSE error. System dead.'.

```
Adrenalin.552   B928 028D 56FD B440 CD21 8F45 028F 05B8 0157 5A59 80C9 1FCD
```

Atom.351

CN: An appending, 351-byte, direct infector containing the texts '*.COM', '[TAD2A] Created by Memory Lapse of Ontario, Canada', '[TAD2A] The Atomic Dustbin 2A - Just Shake Your Rump!' and 'Fail on INT 24 .. NOT!!'. Infected files have the word 4C4Dh ('ML') at offset 0003h.

```
Atom.351        8D9E 5902 8907 5BB4 40B9 5F01 8D96 0501 CD21 B800 4233 C933;
```

Cheryl.374

EN: An appending, 374-byte virus containing the texts '[Cheryl]', '[Jerk1N/DIFFUSION]' and '*.TXT'. Infected files have the word ('Jc') at offset 0012h.

```
Cheryl.374      CD21 B440 8D96 0001 B976 01CD 21B8 0042 33C9 33D2 CD21 B440
```

Erin.407

CN: An appending, 407-byte, direct infector containing the texts '*.com', ':\windows\command', '[Erin-X] (c) 1998', 'YOUR PC HAS BEEN INFECTED WITH THE ERIN-X VIRUS FOR AWHILE', 'ERIN LEA POPE I LOVE YOU' and 'C:\windows\desktop\Erin'.

```
Erin.407        CD21 B440 B997 018D 9603 01CD 21B4 3ECD 21B4 4FEB 9DB4 098D
```

Erin.883

CN: An 883-byte appender with the texts ':\windows\command', '*.com', '[Ultima Pope Virus]', 'C:\windows\desktop\UPOPE', 'Erin LEA POPE i love you' and 'Written by EV short for emperor virii Erin I love you so i write a bunch of viruses haveing greetz to u in them because i love you erin your my one and only in my heart i love you erin i hope you have feelings for me someday'. The payload, which triggers on the thirtieth of any month, prints the above message and installs the destructive procedure in memory.

```
Erin.883        B440 B989 028D 9603 01CD 21B8 0242 33C9 33D2 CD21 B440 B9EA
```

Fatec.500

CN: An appending, 500-byte direct infector containing the encrypted texts '*.c?m', 'FATEC-SP Brasil 1996' and '[SC.FATEC-SP.#1] by [SC]'. Infected files start with the string 'PKX'.

```
Fatec.500       E886 00B4 40B9 F401 8D96 0C01 2EFE 8E0D 02CD 21E8 7300 2EFE
```

Foggy.382

CN: A 382-byte, appending direct infector containing the text '[HauNTinG]' and the encrypted message 'TaWnyOWLSmUGgleRwHiSpeRsMiDNIgHT*.?oM'. Infected files have their time-stamps set to 62 seconds.

```
Foggy.382       E814 FFBA 5BFD B97E 01B4 40CD 21B8 0042 33C9 99CD 21BA D8FE
```

Gerli.977

CR: An encrypted, appending, 977-byte virus containing the text ' Gerli Virus, Anti-Ren-Del.'. Infected files have the byte 9Eh at offset 0003h.

```
Gerli.977       BE0F 01B9 BD04 81E9 0F01 268A 0232 86BC 0426 8802 46E2 F3C3
```

Guppy.152C

CR: Another minor variant of this simple, appending 152-byte virus infecting files starting with the byte E9h (near jump instruction).

```
Guppy.152C      978B D6B1 9883 EA40 B440 CD21 2BD2 B800 422B C9CD 21B1 032B
```

Hail.327

CN: A 327-byte, appending direct infector containing the texts 'ghost in the shell', 'hail and kill 97' and '*.C?M'.

```
Hail.327        B440 B947 018D 9600 00CD 212E FE86 7100 B43E CD21 B44F CD21
```

JDC.1165

CN: An appending, encrypted, 1165-byte virus containing the texts 'Hello from John Darland!!! JV-102e - Nothing-to-do', '*.COM', 'JDC Production' and 'John Darland Computing'. Infected files end with the byte 0ACh.

```
JDC.1165        B9A4 03BB 2701 8A26 EA04 8A07 32C4 CDDE 8807 4349 83F9 0075
```

Mdrg.544

CR: A stealth, 544-byte appender with the texts 'Mandragore', '[Mdrg v3.8]' and 'Mandragore'z sPirIt haunts ur computah !'. Infected files have their time-stamps set to two seconds.

```
Mdrg.544        B43F B91D 0233 D2FE C4CD 21B8 0042 33C9 33D2 CD21 B43F B903
```

Neurotic.568

C: A 568-byte virus containing the texts 'Tranquilo chico que si no es en septiembre será en Junio :-)', 'Que los 12 créditos mínimos te acompañen' and 'by nEUrOtIc cPu cOrpOrAtIOn S.A.'. Infected files have the byte 40h ('@') at offset 0003h.

```
Neurotic.568    F3A4 B8BA BACD 213D CACA 744C B821 35CD 212E 899E 8F02 2E8C
```

Nucleii.1203

CN: An overwriting, 1203-byte, direct infector containing the texts 'nUcLeii~ *v. i. a*', '*.*', and '1200..n0name'.

```
Nucleii.1203    E80F 00BA 0001 8B1E BC03 B9B3 04B4 40CD 21C3 BA5C 03B4 1ACD
```

Orce.60

CER: A 60-byte overwriter. Infected EXE programs become COMs (ie the MZ-header is overwritten).

```
Orce.60         218B D81E 0E1F B440 BA00 01B9 3C00 CD21 B43E CD21 1F61 9DEA
```

Quell.511

CR: A 511-byte appender with the text 'COMcom'. Infected files have the byte 4Bh at offset 0003h.

```
Quell.511       BA00 00B9 F501 B440 CD21 B800 572E 8B0E 1201 2E8B 1614 01CD
```

Quevedo.442

CEN: An overwriting, 442-byte, direct infector containing the texts 'Virus QUEVEDO! by Xavirus Hacker', 'Dedicado a Francisco de Quevedo y Villegas, el mejor escritor conceptista que ha pisado nuestro suelo. Quevedo: ¡aun vives en nuestros cerebros!*.com', '*.exe', 'c:\windows\win.com' and 'WINDOZE SUXX!!!!! Exiting...'.

```
Quevedo.442     B9BA 01BA 0001 B440 CD21 FBB8 0157 5A59 CD21 B43E CD21 B44F
```

SillyC.128

CN: A 128-byte, direct-infecting appender with the text '*.com'. It re-infects already infected files.

```
SillyC.128      B440 B980 008D 9668 01CD 21B8 0042 33C9 99CD 21B4 40B9 0300
```

SillyC.143

CN: An appending, 143-byte, direct infector containing the text '*.COM'. Infected files start with the byte 4Dh ('M').

```
SillyC.143      B440 B98F 008D 9604 01CD 21B8 0042 33C9 99CD 21B4 40B9 0400
```

SillyC.168

CN: A 168-byte direct-infecting overwriter with the text '*.com'. Infected files start with 36FFh.

```
SillyC.168      B440 B9A8 0090 BA00 FDCD 21B8 0042 33C9 33D2 CD21 B440 B9A8
```

SillyC.214

CN: An appending, 214-byte, direct infector, containing the text '*.com'.

```
SillyC.214      B440 8D96 0301 B9D6 00CD 21B8 0157 8B8E EF01 8B96 F101 CD21
```

SillyC.329

CN: A prepending, 329-byte virus containing the text '..*.com'. Infected files have their date- and time-stamps set to 30/06/2076 and 3:11:44 respectively.

```
SillyC.329      B949 01B4 40CD 2172 262E A13E 022E 8B1E 9F01 8ED8 33D2 2E8B
```

SillyC.330/359

CN: Two encrypted, appending, direct infectors containing the text '*.CoM'. Infected files have their time-stamps set to six seconds.

```
SillyC.330      E800 005D 8D76 1556 8B96 4501 B998 008B FEAD 33C2 ABE2 FAC3
SillyC.359      8D76 19E8 0200 EB10 8A96 6101 B948 018B FEAC 32C2 AAE2 FAC3
```

SillyE.654

EN: An appending, 654-byte, direct infector containing the text '*.E?E'. Infected files have the word 6E6Ah ('jn') at offset 0012h.

```
SillyE.654      B440 B98E 028D 9600 01CD 21B8 0042 33C9 8BD1 CD21 B440 B91A
```

Sisters.2181

CER: An encrypted, appending, 2181-byte virus containing the plain-text string 'TEMPLE OF LOVE V1.0 MS 95' and the encrypted texts 'FoUnD VIRUS SYSTERS OF MERCY iN yOuR sYsTeM !!!', 'CHKLIST.MS', 'CHKLIST.CPS', 'KRNLC:\COMMAND.COM', 'C:\DOS\COMMAND.COM', and 'SyStEm is now halted.'. The virus contains the payload erasing the CMOS data.

```
Sisters.2181    0600 E8CE FEE9 FFFE 5051 9CB9 3408 2EF6 1446 E2FA 9D59 58C3
```

Sperm.718

ER: An encrypted, appending, 718-byte virus containing the texts '<Dr.Agon przedstawia SPERM-a 2.0>' and '*.exe'. Infected files have the word 4453h ('SD') at offset 0012h.

```
Sperm.718       5053 5152 5657 B8B9 8EBB E660 B9B0 F550 5351 8926 0400 CD01
```

Sperm.756

ER: An encrypted, appending, 756-byte virus containing the texts '<Dr.Agon przedstawia SPERM-a 2.0>' and '*.exe'. Infected files have the word 0404h at offset 0012h.

```
Sperm.756       5053 5152 5657 B8B9 B4BB E660 B9B0 F550 5351 8926 0400 CD01
```

Sterculius.432

CER: An appending, 432-byte virus containing the texts 'STERCULIUS]['. Infected files have the byte 53h ('S') at offset 0003h (COM) and the word 7777h ('ww') at offset 0012h (EXE).

```
Sterculius.432  B440 B9B0 01BA E001 E813 FFB8 0042 33C9 33D2 E809 FF83 FF01
```

WoodGoblin.2423

ER: A polymorphic, appending, 2423-byte virus containing the texts 'AIADWEVDVSMSHIDR' and 'WG03m Copyright (C) 1995-1996 by WoodGoblin'.

```
WoodGoblin.2423 B977 09F3 2EA4 8ED9 BE84 0056 66A5 5F06 1E07 B83D 09AB 58AB
```

Xor.289

CN: An appending, 289-byte, direct infector containing the texts '[XOR]' and '*.COM'.

```
Xor.289         8896 9102 8D96 0301 B921 01B4 40CD 21B8 0042 33D2 33C9 CD21
```

Xute.1056

ER: An encrypted, appending, 1056-byte virus containing the text 'By XUTE!!'.

```
Xute.1056       B920 04BE 3C00 2E8A 160A 00F6 D28A 048A D822 C2F6 D32E 221E
```

FEATURE

From VBS to VBA

Katrin Tocheva
Data Fellows

This article describes the first Visual Basic Script (VBS) viruses, how their development has evolved and how they have become increasingly sophisticated. Furthermore, it will show how these viruses conflict with existing macro viruses and examines the connection between the two.

At the end of October 1998 the first VBS virus, VBS/First, appeared. VBS viruses replicate only if they are run on a computer where *Windows Scripting Host* (*WSH*) is installed. Thus, those most vulnerable to this kind of virus are *Windows 98* and *Windows 2000* (*NT 5.0*) users where *WSH* is installed by default.

However, *WSH* is available from *Microsoft's* Web site as a free, standalone product and can be installed on *Windows 95* and *NT 4.0* machines too. When *Windows 98* and *Windows NT 5.0* become widely used, the population of vulnerable users will increase further – possibly making VBS viruses more common.

On the one hand, VBS viruses are not a big problem at the moment because VBS files are not exchanged as often as, for example, *Word* documents or executables. On the other hand, the VBS language is both very powerful and very easy to program in. Virus writers no longer have to study hard in order to write a virus in assembler. The last three months are proof of this. Never before have we seen so many different viruses of one relatively novel kind appear in such a short period of time.

Early, Simple Efforts

The three variants in the VBS/First family are simple, overwriting viruses which do not contain anything interesting in their code. However, they were the first of their kind.

All these variants are only able to infect in the current directory. The second variant, VBS/First.B, uses the vbHide keyword to hide the fact that the script shells to DOS. The third and last variant, VBS/First.C, uses the FileSystem command to infect both VBS and Java Script (JS) files. Part of the virus code contains a payload which drops a URL file and tries to run it in order to connect to a virus exchange site on the fifteenth of any month.

The First Destructive Code

VBS/Seven.A was the second VBS virus to appear that used a different method of infection. This virus is a non-overwriting, prepender which infects using the Write command. VBS/Seven targets all VBS files in the following directories – the current directory, the *Windows* directory, the C:\Desktop, C:\Profiles\Administrator\Desktop, and C:\Profiles\AllUsers\Desktop. This virus has a nasty payload that overwrites all DOC and TXT files on the C: and D: drives with a picture on the second of each month between 9 and 10am.

HTML Files

Early November 1998 saw the first infected HTM and HTML files. Note, however, that they were not infected with HTML viruses. The viruses themselves were written in VBS. All these viruses are VBS viruses embedded in HTML files. They can be executed only from within *Internet Explorer* (*IE*) *v4.0* and above, because this is the only browser that supports VBS (*Netscape* and others do not). Therefore, only users of *IE v4.0* and later are in danger of infection by this kind of virus. All these viruses are able to infect only if an infected file is executed on a local machine. It is not supposed to be possible to get infected by browsing an infected web page.

When a user tries to open an HTML (or HTM) file infected with a VBS virus, the browser shows a warning. The experience with macro viruses and the so-called built-in macro virus protection in the *Office 97* applications shows that this kind of 'protection' cannot stop viruses from spreading. This is because most users choose to let the macros run by answering 'Yes' to the warning's question.

Similar results seem likely should VBS viruses become at all widespread. Also, the warning mentioned above will not appear if the user has previously lowered the browser's security settings. If such a modification is made, the VBS program will execute without warning.

The first viruses capable of infecting HTML files were the five known variants of VBS/Internal. This virus ensures that its code is executed when infected files are loaded into the browser by inserting the tag

```
onload="<VBS subroutine to be executed>()";
```

into the header of infected files. If an infected file is opened with *IE 4.x*, the message warning of the presence of ActiveX objects will appear. If the user chooses to let the ActiveX object (VBS program, in this case) run, the virus will replicate. The last two variants, VBS/Internal.D and VBS/Internal.E, try to obscure the message box of the warning with their own message box, but this trick works only in the VBS/Internal.D variant and only if the screen resolution is 800 x 600.

All variants of the VBS/Internal family infect HTM and HTML files in both the current and the parent directories. VBS/Internal.D and VBS/Internal.E also infect HTT files which are the HTML templates used by *IE*. The A, D and E

variants use the VBScript operator WriteLine to infect. The B and C variants use the CopyFile operator instead. VBS/Internal.B, is the first companion VBS virus. It creates a companion by copying the original HTM as HTML and infects HTM by copying its body into it. To execute the original file the virus replaces the name of the active file. VBS/Internal.C (also Offline), is an overwriting virus.

Early Encryption Efforts

VBS/Luser.A, also known as Zulu, is the first encrypted VBS virus. The virus contains two simple functions to decrypt its body. Thus, it performs the decryption in several steps, decrypting different parts of itself. When an infected file is executed, the virus drops the file WINSTART.VBS in the *Windows* system directory. This is created using a function named A to decrypt strings in its body.

After that, it modifies the Registry to run WINSTART.VBS automatically each time the computer is restarted. When this file is executed, it uses a function named B to decrypt the other strings it uses. One of these strings is the message which the virus displays on the first of every month. Each time it executes, the virus chooses a directory at random and searches it and its subdirectories for HTM and HTML files. It infects these using the WriteLine operator. The chosen path depends on the environment. When the virus infects, it encrypts itself back using a function named W and appends itself at the end of the infected files.

Early VBS Efforts

In December 1998 we saw the first script virus which infects over the Internet. The first variant, JS/Charlene.A, uses JavaScript. The second variant, which appeared a few days later, VBS/Charlene.B, is written in VBS. On the local machine both Charlene variants infect all HTM, HTML and HTT files in C:\Inetpub\wwwroot, the Web subdirectory in the *Windows* directory and C:\My Documents.

These viruses use the WriteLine operator and prepend their code to the infected files. The second variant also infects HTA files. If the machine is working as a web server Charlene can infect all pages by simply browsing the web. JS/Charlene.A uses a security hole in *IE 4's* interpretation of the 'about' tag. The infected page looks to the browser as if it is loaded from the local zone. VBS/Charlene.B modifies the Registry to lower *IE's* security settings. On the fifteenth of any month the A variant decides randomly whether to connect to the virus writer's web page. The B variant tries to connect to www.avp.ch/avpve (the *AVP* virus encyclopaedia site). There is no report of Charlene in the wild but a major web site infection would be chaos.

Dropping from an Infected Word Document

The first connection between VBS viruses and Visual Basic for Applications (VBA) macro viruses was realized in W97M/ColdApe.A. ColdApe is one of the first macro viruses to use the AddFromString operator to infect documents. It drops the VBS virus VBS/Happy which can infect all VBS files in the directories C:\, C:\Windows, C:\Windows\Desktop, C:\My Documents and C:\Startup.

It uses the Write command to infect other VBS files. The macro virus also drops another VBS program which, however, is not a virus. It uses Outlook, if it is present, and tries to send an email message from the infected user (ApplicationUserName) to Nick FitzGerald, the former Editor of *Virus Bulletin*.

Inserting a VBA Macro Virus

Next, we saw the VBS/Loud.A dropper. It was distributed in both VBS and HTML versions. Each of them contains a VBS program which then inserts a VBA macro virus (W97M/Loud.A) in the global template of *Word 97*. To infect the global template, the VBS/Loud.A dropper uses the InsertLines command. Once inserted, the VBA macro virus will infect all *Word* documents when they are closed (it uses a Document_Close event handler). The InsertLines command prepends the virus code to any code already existing in the ThisDocument module.

In this way, every time the dropper script is executed, it will add another copy of the virus code to global template. This will increase the size of the global template and will cause a Visual Basic error message (because two or more subroutines named Document_Close will be present). The Visual Basic Editor will display the virus code and this makes the infection very obvious.

VBS/VBA Infection

The first VBS virus which could infect both VBS files and Word documents was VBS/Break.A. In this particular virus the VBS code forms part of the virus itself, not a dropper as it is in VBS/Loud.A. To infect *Word 97* documents, the virus uses the commands CreateObject('WordApplication') and 'AddFromFile'. Similar to the Class viruses, it inserts its code in the ThisDocument module. The virus uses the ReplaceLine command to comment out the VBS-specific part of its code.

The next time *Word 97* is executed, the virus will infect all documents when they are closed (again, by using the Document_Close event handler). In the other direction, from *Word 97* documents to VBS files, the virus replicates by infecting all VBS files on the C: drive, overwriting them with its code by using a simple Write command – in this case, replicating only on the fifteenth of each month.

VBS/DOC/HTM Infectors

The VBS/Hopper family infects as VBS files and *Word* documents, and as HTML files. Hopper infects other VBS files by prepending its code to the original file using the WriteLine command. Some variants also infect HTA and HTT files. Some of them lower *Internet Explorer's* zones' security settings.

To ensure that the virus works on all platforms, when infecting VBS and DOC files, it comments out the HTML-specific part of itself. To infect the global template, it uses the AddFromFile command. Like Class, it infects *Word 97's* global template, and infects documents by using the InsertLines command using a Document_Close event handler. AddFromFile and InsertLines commands can often result in multiple infections or 'sandwiches'.

Fancy a Sandwich?

Multiple infections are caused when more than one virus adds its code to the same module (usually ThisDocument). This is sometimes called 'pseudo-parasitic infection'. Such 'sandwiches' can occur when a VBS virus infects a system already infected with an extremely widespread virus like W97M/Class.D.

The viruses W97M/Brenda.A and W97M/Class.D were involved in the first pseudo-parasitic infection. The system was previously infected with Class.D and then the user opened a document infected with Brenda. The command AddFromString, used by Brenda when infecting the global template, caused this virus to insert its body just before that of Class.D in the ThisDocument module. In this particular case, the CLASS.SYS file that Class.D uses was previously deleted. Thus, Brenda replicated into documents by using the AddFromString command to insert all the lines from the NORMAL.DOT ThisDocument module into the document.

In other words, Brenda added its body to Class.D's body (which was initially present) to all infected documents. That is why, in this particular case, the virus was not polymorphic. If the CLASS.SYS file is present, however, all the infected documents will be polymorphic. The reason for this is that the AutoClose subroutine from Class.D executes before the Document_Close event handler used by the Brenda part of the sandwich.

Hence, all documents will be infected first by Class.D and then by Brenda by copying all lines of code from its module, thus carrying the body of Class.D with itself. The resulting infection contains the non-polymorphic part of Brenda/Class.D from the global template and the polymorphic part for the Class.D infection. All the infected documents will contain Brenda once and Class.D code twice (one constant, non-polymorphic Class.D part from the global template and one polymorphic part due to the Class.D infection).

The same sort of double infection happens on a subroutine level when VBS/Break.A infects the global template which has been infected previously with Class.D. Break replicates using the AddFromFile command when trying to infect the global template from VBS. For exactly the same reason as in the Brenda/Class.D multiple infection, the Break virus adds its code, when infecting the global template, at the beginning of the ThisDocument module. This time, the global template will contain Break's code at the start and Class.D code after that.

Again, the Class.D virus will infect documents first due to its AutoClose subroutine executing before Break's Document_Close event handler. Thus, all infected documents will contain polymorphic forms of Class.D. After that, Break infects the same document by inserting only the lines from its body in the beginning of the ThisDocument module (already infected with Class.D). This is because Break uses the InsertLines command. Here, Break inserts only its code when infecting documents from the infected global template. Therefore, all the infected documents will contain Break code once and Class.D code once.

If a global template, previously infected with the Class.D virus, gets infected with a virus which uses the InsertLines command and inserts all its lines in the documents it infects (W97M/Loud.A) then the resulting Loud/Class 'multiple infection' looks similar to that of Break/Class. The global template consists of the Loud virus code at the beginning of the ThisDocument module and Class.D code after that. Loud inserts its code before that of Class because it uses the InsertLines command.

This 'sandwich' will continue to infect documents and will be polymorphic, like Class.D, because the Class part of the virus is the code which infects first – it uses the AutoClose subroutine that executes before the Document_Close event handler from the Loud part of the 'sandwich'.

Another example is a virus that uses the AddFromString command and AutoClose/AutoOpen subroutine (for example W97M/ColdApe.A). If such a virus infects a global template already infected with Class.D, it will insert the virus body at the beginning of the ThisDocument module because it uses the AddFromString command.

The resulting multiple infection in the ThisDocument module will contain first the code of ColdApe and after that Class.D code. This cannot replicate and is not a virus because both virus codes present in the ThisDocument module of the global template use the AutoClose subroutine when infecting documents. This causes a VBA compilation error and stops the virus replication.

Most multiple infections can infect documents but they cannot reinfect global templates any more because of a compilation error caused by double infection at the subroutine level. All these VBS and VBA infection methods will result in many non-working 'sandwiches'. It is possible that many of these viruses will kill each other off by using their own infecting methods.

The Situation Now

In only three months, the number of VBS viruses has increased noticeably. Now there are several infection methods and more sophisticated VBS viruses appear quite often. We have already received a sample of an HTML file infected with an encrypted VBS virus (VBS/Luser). When *Windows 98* and *Windows* (*2000 NT 5.0*) become widely used, we expect the VBS virus situation to worsen.

VIRUS ANALYSIS 1

A Hill of Beans

Costin Raiu
GeCAD, Romania

The first known Java virus, StrangeBrew, (see *VB*, September 1998, p.11) formed an interesting addition to the large spectrum of viruses detected by today's anti-virus products. While buggy, this particular virus was able not only to infect many Java class files correctly but to propagate its code further from there.

It also had to load that code from a single infected file, since Java viruses cannot access their own bytecode in memory. The recent BeanHive virus goes one step further down this evolutionary road by loading its own replication code from the Internet.

The BeanHive Virus

The author of StrangeBrew posted a message announcing his new creation – BeanHive – to the alt.comp.virus newsgroup in mid-January 1999. A URL to his homepage on the web was provided for anyone interested (or not) in evaluating this new virus.

The relevant WWW page hosts a Java applet which is designed to allow the direct infection of Java class files on your local machine. Interestingly enough, the Java applet is signed, so the browser will allow it to run with extra privileges, should you authorize its signer. As one would expect, this is only allowed after the user has been warned about a possible security hazard, and only after asking for confirmation to run the untrusted code.

The signed applet loads the virus infection module from the author's web site, and calls it to process the selected file. What is interesting about the BeanHive virus is that the code attached to the target file is not the real virus body, but a short loader for it. This part of the virus should be able to defeat almost any anti-virus heuristic analysis, unless the analyser was written to download the data for itself.

While inserting the virus loader into the target file, the class is patched so that the object becomes a descendant of the ClassLoader superclass. This allows the loader to build a new class directly from the Internet-downloaded code and then run it.

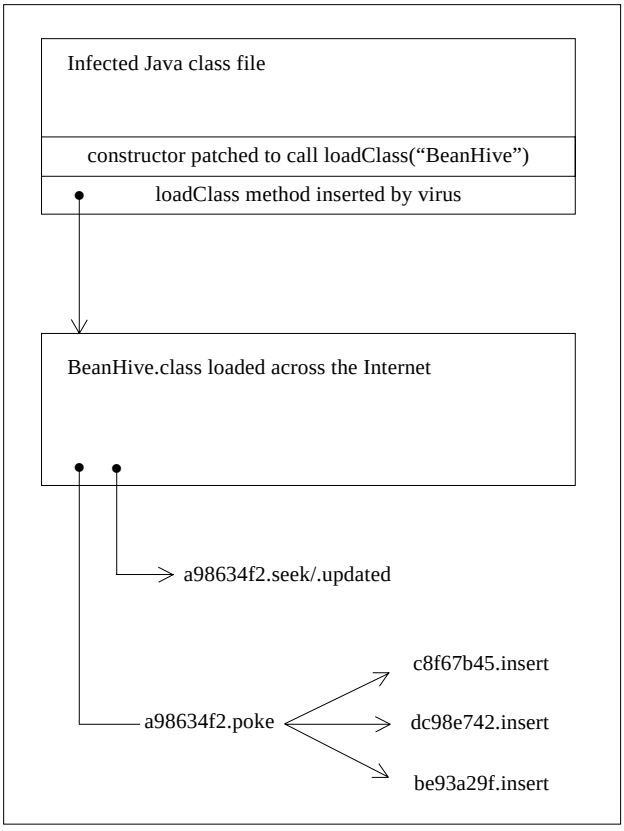
BeanHive.class

BeanHive.class is the Java application downloaded from the virus author's website. After the bytecode for this class is initialized in memory, the virus loader runs it. The BeanHive class is the main virus replication module. Unlike StrangeBrew, BeanHive is split into a couple of class files each designed to perform highly specific tasks. The BeanHive class has one main purpose: to scan the current directory for .class files with sizes smaller than 65536 bytes which can also be read from and written to.

If such a file is found, the 'poke' member of the class named e89a763c is called. This member function is designed to implant the virus body into the new host file. Once again, BeanHive shows some 'improvements' over StrangeBrew when it comes to finding targets for infection: for each subdirectory in the current folder this 'find & touch' code is called recursively. The files in the current directory will be infected in addition to files in subdirectories. Up to three files will be infected by the virus each time its code is run.

The e89a763c class contains most of the Java file format parsing routines. It also checks for the 0xCAFEBABE signature in the header if the constant pool count is higher than 160. It makes many other checks to ensure that the candidate file is suitable for infection.

If all the checks are passed, the actual virus insertion code is called, in the form of a class called c8f67b45. The 'insert' member of this class prepares the victim class for infection. If the preparation is successful, a further class is used to continue the infection process.



The dc98e742 and be93a29f classes effectively insert the loader code in the victim file, patching the constructor of the target class file in order to allow for the virus loader method to take control. They also process the new constant pool and relocate the loader bytecode.

In addition, a small class called a98b34f2 is used by the virus during infection as a wrapper for file 'seek' methods. As with StrangeBrew, BeanHive's infection methods are rather brutal and still full of bugs. Some files are damaged during infection, and the infection routine will throw an exception while parsing some target files.

Like Bees Round a Honey Pot

It should be noted that since BeanHive loads its code from the Internet each time a new infection occurs, the relevant portion of code can, theoretically, be updated by the author to new versions. A more worrying prospect is that it could even be replaced with a Trojan designed to plant a 'back-door' in your computer. As the author may well have access to the logs on the web server which holds the infective body, this should be concern enough to ensure any attempted infection by BeanHive is detected and blocked.

It is unlikely that BeanHive, or StrangeBrew, will become wild, so all the hits on the author's web site will probably be from anti-virus people trying to replicate his virus.

Conclusion

The BeanHive virus uses some new concepts but it is neither revolutionary nor totally unexpected. The multi-object model employed by the virus shows an interesting path for the possible development of future Java viruses. It also suggests increased care about the code's efficiency and reliability from the author.

Overall, the only remarkable thing about BeanHive is that it is the first virus which does not store itself on the host computer but loads its infective body from the Internet each time it infects. This can also be seen as a limitation, as the infection will not work on computers without Internet access. Anyway, this idea did not go unnoticed in the VX world because only weeks after BeanHive was released, a macro virus which works in a similar way was reported.

Java/BeanHive	
Aliases:	None known.
Type:	Non-resident, direct Java class file infector.
Payload:	None.
Detection:	No reliable string can be extracted.
Disinfection:	Delete infected class files and replace from clean originals.

VIRUS ANALYSIS 2

ViruSoft Office

Andy Nikishin
Kaspersky Lab

Until recently, macro viruses infected only *Word* or *Excel*. Then, about a year ago, the first multi-platform macro virus appeared. It was Cross (see *VB*, June 1998, p.11), and it was huge and buggy. As time went by, virus writers honed their skills and O97M/Shiver (October 1998, p.9) made use of DDE to cross-infect *Word* and *Excel*.

At the end of 1998, the first *PowerPoint* virus appeared. A month or so on there is a new multi-platform virus which can spread amongst the three main *MS Office* applications. It is known as Triplicate and takes advantage of the object model now employed in *Microsoft* products: ActiveX.

ActiveX is just a new name for what used to be called OLE Control. It is based on Component Object Model (COM) and Automation (previously OLE Automation) technologies. This technology was created as a useful tool for writing software controls. It seems likely that it will become more popular, as *Microsoft* uses it extensively in the operating system and application programs.

Looking Closer

The main feature of this technology is its ability to access objects in one application from another using a standard interface. Thus, a simple procedure can easily get the data from a given cell in an *Excel* spreadsheet or create a report in *Word*.

Despite the advantages of this technology, there is one serious drawback from the virus protection point of view. It allows a virus to spread easily from one application to another. Triplicate is an example of this.

O97M/Triplicate is a multi-platform macro virus which infects *Office 97* components – *Word* documents, *Excel* sheets and *PowerPoint* presentations. The virus does not manifest itself in any way, and does not deliberately destroy any data on the computer. This is the first known virus to infect three of the *Office 97* components.

Triplicate contains three VBA5 procedures in *Excel*, *PowerPoint* and *Word* files – Document_Close in *Word*, Workbook_Deactivate in *Excel* and actionhook in *PowerPoint*. Its infection routines are separated into three subsets. The appropriate subset is activated for the *Office* component under which the current instance of the virus runs.

There are at least four known versions of this virus at the time of writing. Two of them are just 'bug-fix' versions of the first one, but the third contains some significant new

features. This version uses the Word 97 Template Vulnerability (see *VB*, February 1999, p.4) to enter a computer, unheralded, from the Internet.

Infection via Documents

When Triplicate is activated from an infected *Word* document, it disables *Word's* virus protection and checks NORMAL.DOT for its presence. If that file is not infected, the virus starts to get into other *Office* components. These operations consist of three steps: *Word* infection, *Excel* infection and *PowerPoint* infection.

The first is Triplicate's simplest operation. It just copies its code from the current document to NORMAL.DOT. The second step is more complex.

Initially, the virus creates an instance of *Excel* using the CreateObject('Excel.Application') function. It then checks for the BOOK1 file in the startup folder. If the file is not there, the virus infects *Excel* by disabling its virus protection in the system registry and creating a new workbook. It then copies its code into that workbook, saving it as BOOK1 in the startup folder. Every spreadsheet from this folder is automatically loaded when *Excel* starts, and, as a result, *Excel* is infected on the next restart.

PowerPoint infection is similar. Triplicate checks in the template folder for Blank Presentation.pot and tries to locate a module in it called 'Triplicate'. If this module is not there, the virus infects *PowerPoint* by disabling its virus protection in the system registry and creating a new module Triplicate in Blank Presentation.pot. It copies code into it and then adds a new 'shape' into the presentation, with the same width and height as those of slides. An activate procedure for this shape is set to actionhook (this procedure will activate when a user clicks on this shape).

Finally, the virus checks for the infection of the current *Word* document and infects it if it is clean. These routines are only executed when the virus is loaded from an infected template, and a new, clean document is closed.

Infection via Spreadsheets and Presentations

Excel and *PowerPoint* procedures are quite similar. The BOOK1 file in the *Excel* startup folder is used by Triplicate as an indicator that *Office* is infected. Firstly, the virus looks for this file and if it does not exist, infects *Office* applications. After that, it tries to infect *Word*.

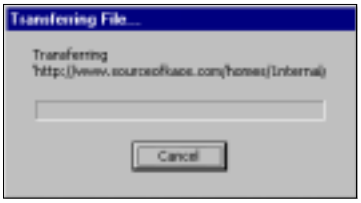
Instead of using the CreateObjects function, Triplicate uses GetObject to get objects from the currently active application. It needs that to infect NORMAL.DOT, which cannot be accessed for writing if it is already open in *Word*. If *Word* is not currently active, the virus accesses it and starts its spreading routine. It deletes all code in the normal template, creates the DisableAV procedure, copies a block of code there, executes and then deletes it. That eight line procedure disables *Excel* and *PowerPoint* virus protection. Triplicate then copies its code from the infected file to the normal template and *Word* infection is complete. *Excel* and *PowerPoint* are infected in the next step when Triplicate infects the *Excel* startup folder from the *PowerPoint* presentation, or inserts its code into the *PowerPoint* template (as above) spreading from an infected *Word* document. Then the virus checks *Excel* for current sheet infection and if it is not infected yet, infects it. The *PowerPoint* activation procedure has an additional detail; the virus activates its infection code in one case in seven, depending on a randomly-generated number.

Infection from Remote Word 97 Template

Office applications have vulnerabilities. Some are already fixed, some are not. One variant of Triplicate takes advantage of the Word 97 Template Vulnerability, which allows macros to run without warning the user as the document is opened. This template may be on the same workstation, a share on the LAN, a web server on the corporate Intranet or even on the Internet.

This security hole was used to distribute the virus. A specially prepared *Word* document was placed on a Web site and when downloaded and opened in *Word*, a reference to a linked template caused *Word* to download and open an infected template from the virus author's Web site.

The trick is that *Word's* macro warning is not activated in this case – macros in the infected template go undetected, and the virus macro runs and infects the system. A user can tell the moment such a download starts – while transferring the file *Word* shows a progress dialog box complete with the path (or URL) to the template.



Conclusion

It seems that Triplicate may 'father' a whole series of multi-platform viruses. Using these techniques, viruses can spread not only to *Office* applications but to Visual Basic Scripts (including those embedded in HTML files and the like), and other ActiveX-oriented applications.

O97M/Triplicate

Aliases:	OM97/Crown.
Type:	Native *Word 97*, *Excel 97* and *Powerpoint 97* macro virus that can cross-infect either/both other platforms.
Payload:	None.
Detection:	Ensure macro virus protection options are enabled in Office applications and be increasingly vigilant.

COMPARATIVE REVIEW

NTirely Up to You

Nick FitzGerald

The first *Virus Bulletin* comparative review I oversaw was on the *NT* platform, so perhaps it is fitting that it is the platform for this, likely my last. Eighteen months ago, eighteen products lined up. Now, with several 'new' products relative to that review, we again have eighteen products to test due to various acquisitions and mergers.

Twelve products in the September 1997 review sported on-access scanners. Thirteen of the reviewed products here have full-featured on-access scanners – meagre progress – but one consistently crashed when this option was enabled. The more things change, the more they stay the same…

Test-sets and Procedures

All of the detection tests were run on three essentially identical machines under *NT 4.0* with Service Pack 4 applied. To remove any possible variation due to inconspicuous hardware differences, a single machine was used for all speed and overhead tests.

The *VB* test-sets were updated, and most importantly the In the Wild File and Boot test-sets were aligned to the December 1998 WildList. As that WildList was posted a little later in the month than is usual, the product developers were given an extended submission deadline of 6 January 1999. Of some personal interest to the reviewer was the performance of the products against W97M/ColdApe – the A variant of which was new to the December WildList, but both were clearly 'doing the rounds' at the time.

Also newly added to the In the Wild test-set were several Laroux variants. As a few products have shown something of a weakness on *Excel* macro viruses in the past, the impact of this development, if any, on the the In the Wild File results should be noted.

Whenever possible, the tests were run against a copy of the test-sets stored on a read-only share on a server. Various, but fortunately few, problems were encountered with this setup and they were resolved by copying the test-set from CD to a local drive for the duration of each test that required this. One or two test cases were run directly against the test-sets on CD, removing the need to copy the virus samples to hard disk, though this was prone to triggering 'inpage operation' faults from *NT*, and on occasion Blue Screens of Death (BSOD).

In all cases, the software under test was installed and configured in its default form, unless the requirements of a given test condition dictated otherwise. For example, on-access components were completely disabled while running on-demand tests and report files were always generated for the main detection tests, regardless of the default setting for that option but left at the default setting for speed tests. All tests were run from the local Administrator usercode on the workstation and as a very low-privileged usercode on the server, having only read access to the test-set directory tree.

The products were, of course, subjected to *VB's* typical speed and overhead tests. The hard disk scanning test, combining speed and false positive testing on the 5500 executables of the *VB* Clean test-set, should produce results directly comparable with recent *NT* comparatives.

The overhead introduced by the on-access scanner was tested using XCOPY to move large numbers of executables, the results being compared against a baseline and normalized across the products for subsequent presentation. Floppy disk speed tests were performed upon two almost identical disks, differing only in that the files on one were universally infected with Natas.4744.

As usual, developer requests to run in 'all files' mode or with special commandline options were ignored. Whilst it is undoubtedly true that many 'typical users' of these products run them with other than the 'out of the box' settings, this observation provides little indication of what might represent 'typical usage'. Much of the general use of these products will simply be with the 'factory settings', and that condition is easily configured by others wishing to reproduce the test conditions.

It should also be noted that the same vendors who ask for 'full-paranoia' modes (all files, high heuristics), often equally strongly advocate 'standard settings' when speed and false positive testing is under discussion. You can't have your cake and eat it too…

In fact, this issue accounts for the differences often seen between *VB* test results and those of various certification agencies. A product *VB* claims fails to obtain 100% against the touchstone In the Wild test-set, may well do so if run in full-paranoia mode. Unless false positive and speed tests are run with the same settings, however, the meaning of the results as a whole is an open question.

The complete detection tests are reported in the main tables. The results reported in the summaries are only the on-demand ones, plus the on-access result for the combined In the Wild test-sets, where applicable.

Aladdin eSafe Protect v2.0

ItW Overall	99.3%	Macro	91.3%
ItW Overall (o/a)	99.2%	Polymorphic	91.8%
ItW Boot	98.8%	Standard	97.7%

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Protect	83	98.8%	840	99.4%	99.3%	2426	91.3%	13637	91.8%	1010	97.7%
Alwil Avast32	84	100.0%	856	99.9%	99.9%	2578	96.7%	14435	98.7%	1046	99.7%
CA InoculateIT	83	98.8%	853	99.5%	99.4%	2608	97.9%	14433	99.1%	1046	99.7%
Command AntiVirus	84	100.0%	844	99.6%	99.5%	2647	99.4%	14198	97.4%	1036	99.2%
Cybec Vet Anti-Virus	84	100.0%	842	99.5%	99.5%	2555	96.1%	14185	97.3%	1043	99.5%
Data Fellows FSAV	84	100.0%	856	99.9%	99.9%	2665	99.8%	14444	100.0%	1037	99.5%
DialogueScience Dr Web32	75	89.3%	857	100.0%	99.0%	2511	94.2%	14444	100.0%	1051	99.7%
ESET NOD32	84	100.0%	857	100.0%	100.0%	2657	99.5%	14444	100.0%	1046	99.7%
GeCAD RAV	83	98.8%	843	99.6%	99.4%	2631	98.6%	13668	94.5%	1001	96.1%
Grisoft AVG	76	90.5%	856	99.9%	99.1%	2071	77.4%	13496	93.3%	913	87.9%
iRiS AntiVirus	84	100.0%	857	100.0%	100.0%	2652	99.4%	14433	99.1%	1046	99.7%
Kaspersky Lab AVP	84	100.0%	857	100.0%	100.0%	2626	98.3%	14444	100.0%	1046	99.7%
NAI NetShield NT	84	100.0%	857	100.0%	100.0%	2653	99.5%	14091	96.7%	1046	99.7%
Norman Virus Control	84	100.0%	857	100.0%	100.0%	2612	98.1%	14444	100.0%	1046	99.7%
Proland Protector Plus	48	57.1%	470	58.8%	58.6%	1219	46.3%	1735	10.7%	494	54.1%
Sophos Anti-Virus	84	100.0%	857	100.0%	100.0%	2614	98.6%	14444	100.0%	1035	99.2%
Symantec Norton AntiVirus	83	98.8%	856	99.9%	99.8%	2644	99.1%	14443	98.7%	1037	99.5%
Trend OfficeScan NT	82	97.6%	856	99.9%	99.7%	2496	93.8%	14319	96.8%	1026	98.7%

Recently purchased by *Aladdin Knowledge Systems* (*AKS*), the former *eSafe* product shows little sign of change yet, if in fact, any is likely. Virus scanning is one part of the complex of functionalities that *eSafe Protect* provides and finding the desired configuration settings amongst its plethora of options could be daunting to the less-experienced user. This is not necessarily a bad thing!

Hare.7610 on a 1.44 MB diskette is still *eSafe's* bugbear in the ItW Boot test-set, but was not solely responsible for the product's failure to reach VB 100% performance. The Win95/Fono VxD, Win95/Marburg-infected screen savers and all *Windows* (NE) EXE samples of TPVO.3783.A were also missed.

Detection percentages in the low nineties on the Macro and Polymorphic test-sets are not encouraging compared to most other products in the review. *eSafe Protect* has something of a penchant for missing the template sample forms of *Word* macro viruses (those samples usually being derived from the NORMAL.DOT off the replication machine). Given that whilst not necessary, most successful macro viruses do infect the default global template, the persistence of this effect in *eSafe's* results (and in those of its forerunner, *ViruSafe*) is of concern.

Initially, on-access tests proved problematic, with Dr Watson intervening part-way through the tests and closing what it considered was an errant process – namely the *eSafe Protect* scanning service. *AKS* staff confirmed a problem and were working on a fix as this copy went to proof. After reporting this to *AKS*, however, another fresh install was tried and this time the on-access tests ran to completion.

AKS claimed that the on-access scanner should detect exactly the same viruses as the on-demand one, and the (mainly) small difference between the results of the two scanning tests may be due to lingering issues with a not fully functional service. But then, I have been told innumerable times by many vendors that both test modes should return the same results, and experience tells me this is the exception rather than the rule. That said, *eSafe Protect's*

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Protect	82	97.6%	840	99.4%	99.2%	2420	91.1%	12617	84.7%	1010	97.7%
Alwil Avast32	84	100.0%		n/t	n/a		n/t		n/t		n/t
CA InoculateIT	73	86.9%	841	99.1%	97.9%	2595	97.4%	14187	96.5%	1046	99.7%
Command AntiVirus	73	86.9%	844	99.6%	98.4%	2647	99.4%	14198	97.4%	1036	99.2%
Cybec Vet Anti-Virus	84	100.0%	796	93.0%	93.6%	2560	96.2%	12669	86.9%	363	31.1%
Data Fellows FSAV	84	100.0%	856	99.9%	99.9%	2645	99.4%	14444	100.0%	1037	99.5%
ESET NOD32	84	100.0%	857	100.0%	100.0%	2657	99.5%	14444	100.0%	1041	99.5%
Kaspersky Lab AVP	84	100.0%	857	100.0%	100.0%	2636	98.7%	14444	100.0%	1046	99.7%
NAI NetShield NT	84	100.0%	845	99.6%	99.6%	2653	99.5%	14091	96.7%	1046	99.7%
Norman Virus Control	73	86.9%	844	99.6%	98.4%	2612	98.1%	14198	97.4%	1038	99.5%
Sophos Anti-Virus	84	100.0%	857	100.0%	100.0%	2614	98.6%	14444	100.0%	1035	99.2%
Symantec Norton AntiVirus	83	98.8%	856	99.9%	99.8%	2644	99.1%	14443	98.7%	1037	99.5%
Trend OfficeScan NT		n/a	856	99.9%	n/a	2502	94.0%	14319	96.8%	1026	98.7%

past test results show it does detect the same viruses in both modes consistently, so the divergence here probably was due to the problems noted with the service.

The on-access scanner would appear to have been rewritten, or at least seriously tweaked, since the previous *NT* comparative, as an overhead approaching 100% is nothing like the current incarnation's performance. On these tests, *eSafe Protect* joins *Vet AntiVirus* and *Sophos Anti-Virus* in returning a slightly negative 'overhead'.

Alwil Avast32 v7.70

ItW Overall	99.9%	Macro	96.7%
ItW Overall (o/a)	n/a	Polymorphic	98.7%
ItW Boot	100.0%	Standard	99.7%

Alwil's Avast32 turned in a highly creditable performance, being pipped at the VB 100% post by a single sample – the VxD form of Win95/Fono. Staking 96.7% against the Macro test-set as the weakest result should bring satisfaction to any developer, and with on-demand detection levels around 99% and higher on all other test-sets, this was yet another solid outing from this Czech product.

VB's standard on-access testing mechanism does not allow the detection rate of *Avast32's* resident scanning function to be assessed. This is due to the latter's dependence on file execution rather than 'file open' or 'file read' operations, which other products intercept. The clean hard disk speed test result appears unflattering but as we have noted before, this is a feature. *Avast32* runs on-demand scans in a low priority thread and thus can be left performing a full drive scan with minimal impact on other applications.

CA InoculateIT v4.5

ItW Overall	99.4%	Macro	97.9%
ItW Overall (o/a)	97.9%	Polymorphic	99.1%
ItW Boot	98.8%	Standard	99.7%

Returning good, solid-looking detection on-demand, *InoculateIT's* on-access detection may not be up to the mark these results suggest. It missed a VB 100% award by not detecting W97M/ColdApe.A and the polymorphic boot infector Win95/Fono in the ItW Overall test-set.

InoculateIT's on-access component has no 'deny access' option. Thus, a variation on the usual test method, which depends upon 'on open' detection and a 'deny access' response, had to be employed. In this case, the alternative process involved copying the complete test-set from the server to the test machine with the shield program set to detect only on writes and to delete infected files.

Once completed, about 75% of the test-set resided on the workstation's drive. This was a surprisingly high proportion of the total test-set. A further round of copying this partial

	Scanning Speed						False Positives
	Diskette - Clean		Diskette - Infected		Hard Drive - Clean		
	Time (seconds)	Throughput (KB/s)	Time (seconds)	Throughput (KB/s)	Time (min:sec)	Throughput (KB/s)	
Aladdin eSafe Protect	58	17	116	10	14:48	601	0
Alwil Avast32	64	15	76	16	45:32	196	0
CA InoculateIT	156	6	184	6	6:56	1284	0
Command AntiVirus	62	16	70	17	8:06	1099	1
Cybec Vet Anti-Virus	57	17	66	18	2:27	3633	0
Data Fellows FSAV	120	8	138	9	16:51	528	2
DialogueScience Dr Web32	70	14	170	7	24:00	371	19
ESET NOD32	35	28	65	18	3:20	2671	0
GeCAD RAV	60	16	63	19	11:18	788	8
Grisoft AVG	59	10	67	17	3:43	2395	0
iRiS AntiVirus	57	17	70	17	8:00	1113	0
Kaspersky Lab AVP	60	16	74	16	6:12	1436	2
NAI NetShield NT	241	4	266	4	8:13	1083	0
Norman Virus Control	59	17	95	12	5:24	1648	0
Proland Protector Plus	114	9	125	9	1:16	4606	61
Sophos Anti-Virus	57	17	64	18	3:40	2428	0
Symantec Norton AntiVirus	155	6	169	7	7:35	1174	0
Trend OfficeScan NT	60	487	62	20	5:41	1566	2

test-set to another folder on the PC, wiping the source directory, copying the remaining files back and so on was tried. This resulted in further detections. In total, more than thirty iterations of this procedure were required before three successive runs saw no further files being deleted.

The on-access results presented here were recorded at that point. Although close to the on-demand results, they are not the same and the testing procedure clearly uncovered a weakness in the scanner's architecture. Despite this, the general stability of the product seems much improved over recent-past outings in *VB* tests. Speed was middling and on-access overhead approached 75%.

Command AntiVirus v4.54 8 Dec 1998

ItW Overall	99.5%	Macro	99.4%
ItW Overall (o/a)	98.4%	Polymorphic	97.4%
ItW Boot	100.0%	Standard	99.2%

Command Software AntiVirus (*CSAV*) failed to detect the screen saver (SCR) samples of TPVO.3783.A and Win95/Marburg, as well as the Win95/Fono VxD in the In the Wild File test-set, thus missing out on a VB 100% award.

With detection rates in the high ninety percent range, *CSAV* performs well, if a little more slowly than most of its competitors. Its main weakness in these tests was 86.9% against the ItW Boot test-set under on-access scanning.

Samples with invalid BPBs simply resulted in 'not accessible' error dialogs, rather than notification of the viruses thereon. These same diskettes were correctly identified as infected by the on-demand scanner, so *CSAV* is its own proof that what we were asking of it was not unreasonable.

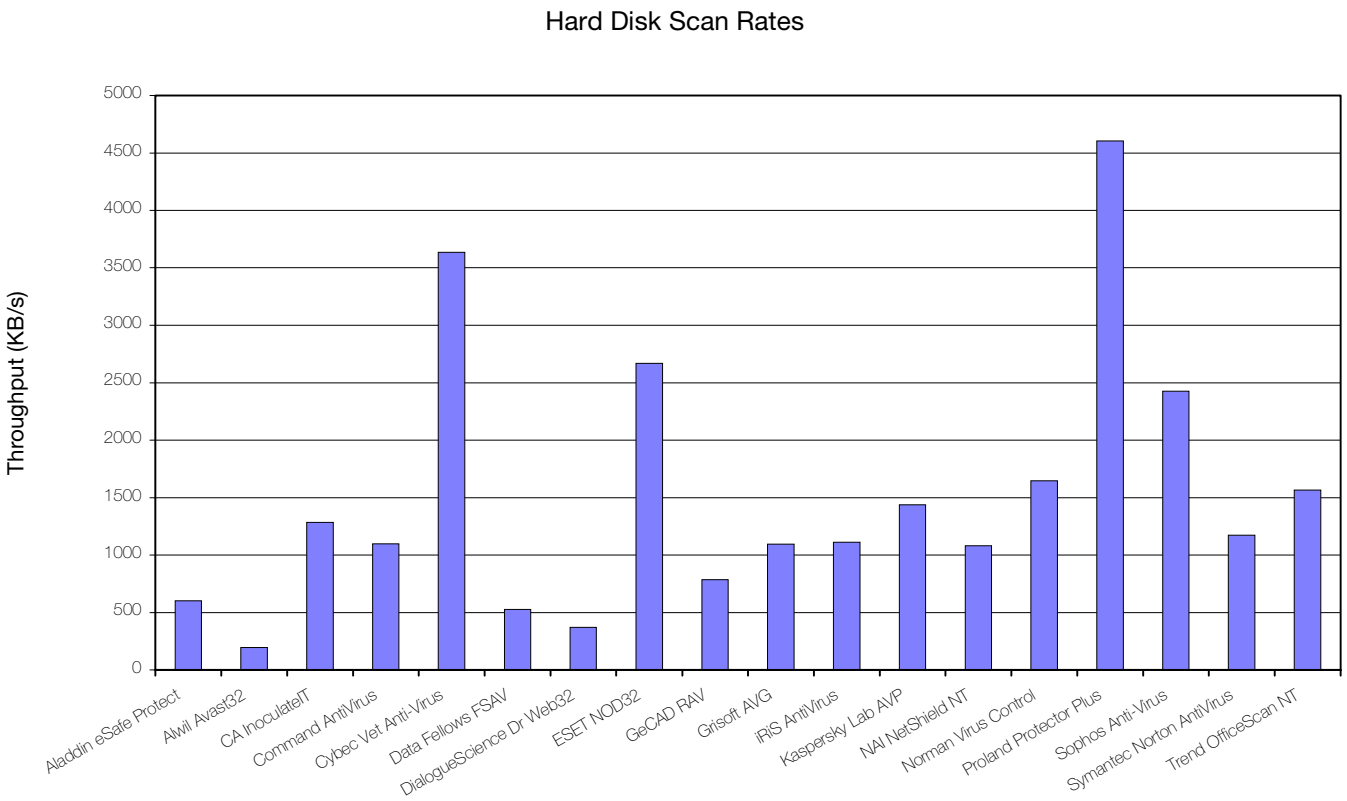
One false positive was registered against the Clean test-set – a 'destructive program'. In keeping with the less than meteoric speed, *CSAV's* overhead was on the high side at 143% once DVP (Dynamic Virus Protection) was enabled.

Cybec Vet AntiVirus v9.93

ItW Overall	99.5%	Macro	96.1%
ItW Overall (o/a)	93.6%	Polymorphic	97.3%
ItW Boot	100.0%	Standard	99.5%

Cybec's Vet was another product to miss SCR infections of TPVO.3783.A and Win95/Marburg in the In the Wild File test-set. It also missed three samples of XM/Compat.A in XLA files. These same Compat and Marburg factors accounted for all its misses in the Polymorphic test-set.

As usual, speed was of the essence with *Vet* and, ignoring *Proland Protector Plus*, it returned 40% higher throughput than the next fastest product. It again returned reliably



negative on-access overhead – as with *eSafe Protect* and *Sophos Anti-Virus*, some file I/O operations are actually faster when its on-access scanner is installed and enabled, than prior to installation of the product.

Overall, on-access detection rates are somewhat lower than their on-demand counterparts. This appears to be by design, with the less common members of the Standard test-set more likely to be missed relative to on-demand performance. The oddity among these results occurred in the Macro test-set, where a slightly higher detection rate was recorded on-access – this was accounted for by *Vet* detecting the XLM samples, generated naturally by five of the *Excel 95* viruses in that set.

One may question the wisdom of electing not to detect viruses 'officially recognized' as being in the wild. Even if one's customers have not (yet) reported the vermin in question, their detection would seem important given these viruses are (or have been), in some strong sense, 'common'.

Data Fellows F-Secure Anti-Virus v4.03

ItW Overall	99.9%	Macro	99.8%
ItW Overall (o/a)	99.9%	Polymorphic	100.0%
ItW Boot	100.0%	Standard	99.5%

The Win95/Fono VxD was all that stood between *Data Fellows' F-Secure Anti-Virus* (*FSAV*) and another VB 100% award for the Finnish developer's trophy room. Returning results within 0.5% of perfect detection in all test-sets is certainly a laudable performance.

The differences between on-demand and on-access detection were all in the Macro test-set, with twenty fewer samples being detected on-access. These comprised two of the four A97M/AccessiV.A and all four A97M/AccessiV.B samples, plus the eleven XM/Compat.A and three of the four XM/Dado.A samples.

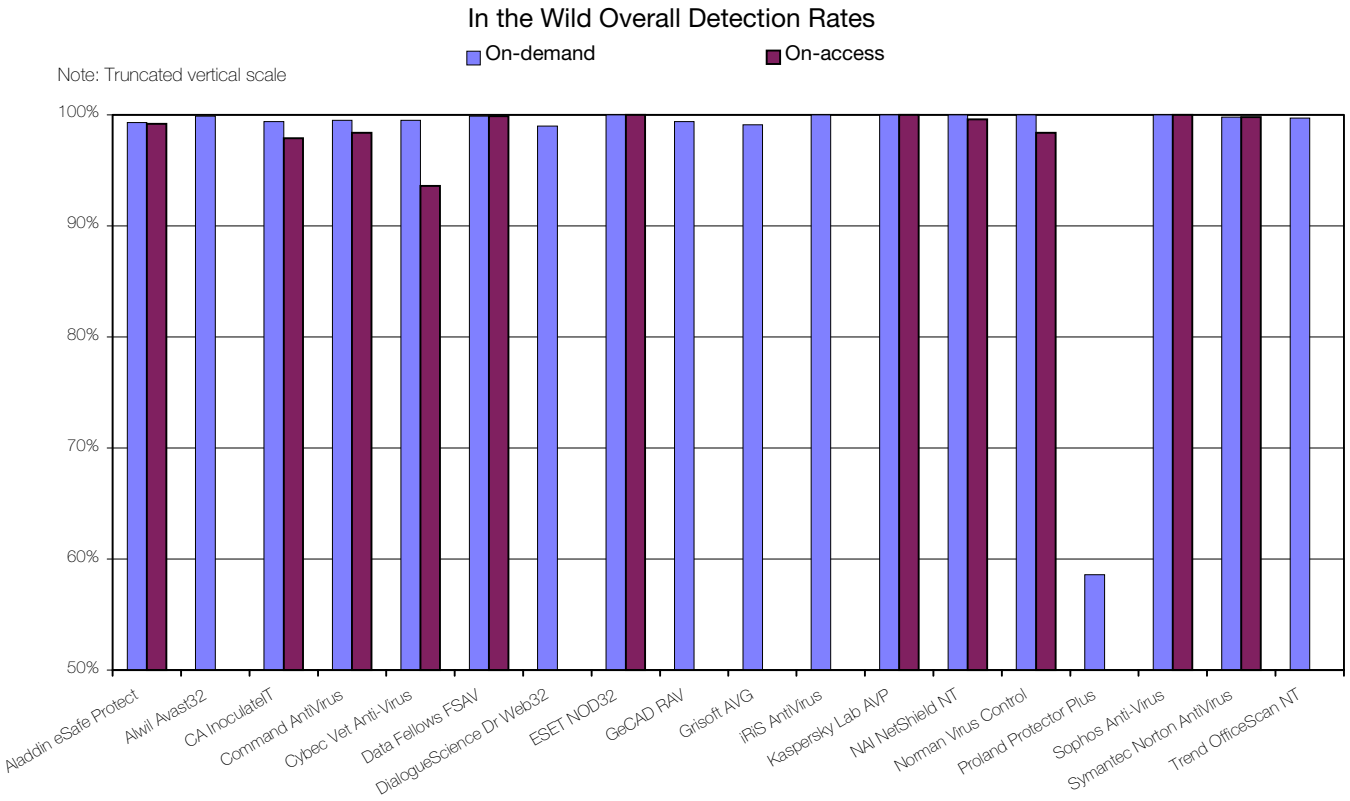
FSAV's great strength is that two good detection engines are glued together in one package, in such a way as to avoid the potential problems of running two active, independent scanners simultaneously. However, this contributes to what is, perhaps, its greatest drawback – neither of the engines it uses are renowned for their speed, so the combined effect of the two causes *FSAV* to place poorly in the speed stakes.

DialogueScience DrWeb32 v4.03aß

ItW Overall	99.0%	Macro	94.2%
ItW Overall (o/a)	n/a	Polymorphic	100.0%
ItW Boot	89.3%	Standard	99.7%

Detecting all the In the Wild File samples is a feat not matched by several of its better-known foes. Unfortunately for *DrWeb's* developers, it is not sufficient to pick up a VB 100% award either.

The scanner found nothing amiss with the diskettes holding the ItW Boot samples of viruses that have invalid BPBs (at least, invalid on the host media in *VB's* test-set – 3.5-inch DD or HD diskettes). Eight viruses that caused similar detection problems for other on-demand and/or on-access scanners were thus missed.



High detection rates were otherwise the norm. Historically the slowest scanner in *VB* reviews, performance has been sufficiently improved for this version to leave that 'honour' to *Avast32*, which as noted elsewhere, runs its scanner as a low priority thread and certainly was not as sluggish in the recent DOS comparative as it appears in those on the Win32 platforms.

ESET NOD32 v1.13

ItW Overall	100.0%	Macro	99.5%
ItW Overall (o/a)	100.0%	Polymorphic	100.0%
ItW Boot	100.0%	Standard	99.7%

Following an impressive debut in the DOS comparative of February 1998 and picking up a VB 100% award with its first Win32 incarnation in the May 1998 comparative, *ESET's NOD* has continued to impress. Capturing another VB 100% award here, this Slovak product detected more samples across all the test-sets than any other.

The only viruses missed were amongst the newest in the test-set – W97M/Marker.A and B, Win32/Redemption and XF/Sic.A. These were 'supplemented' under on-access testing with four rare viruses from the Standard test-set.

NOD32 was second fastest of the useful products, but surprisingly this did not translate into a very low overhead. The on-access scanner's impact on the test machine's performance was not onerous, but certainly not as slight as that of some others. No false positives were recorded.

GeCAD RAV v6.53

ItW Overall	99.4%	Macro	98.6%
ItW Overall (o/a)	n/a	Polymorphic	94.5%
ItW Boot	98.8%	Standard	96.1%

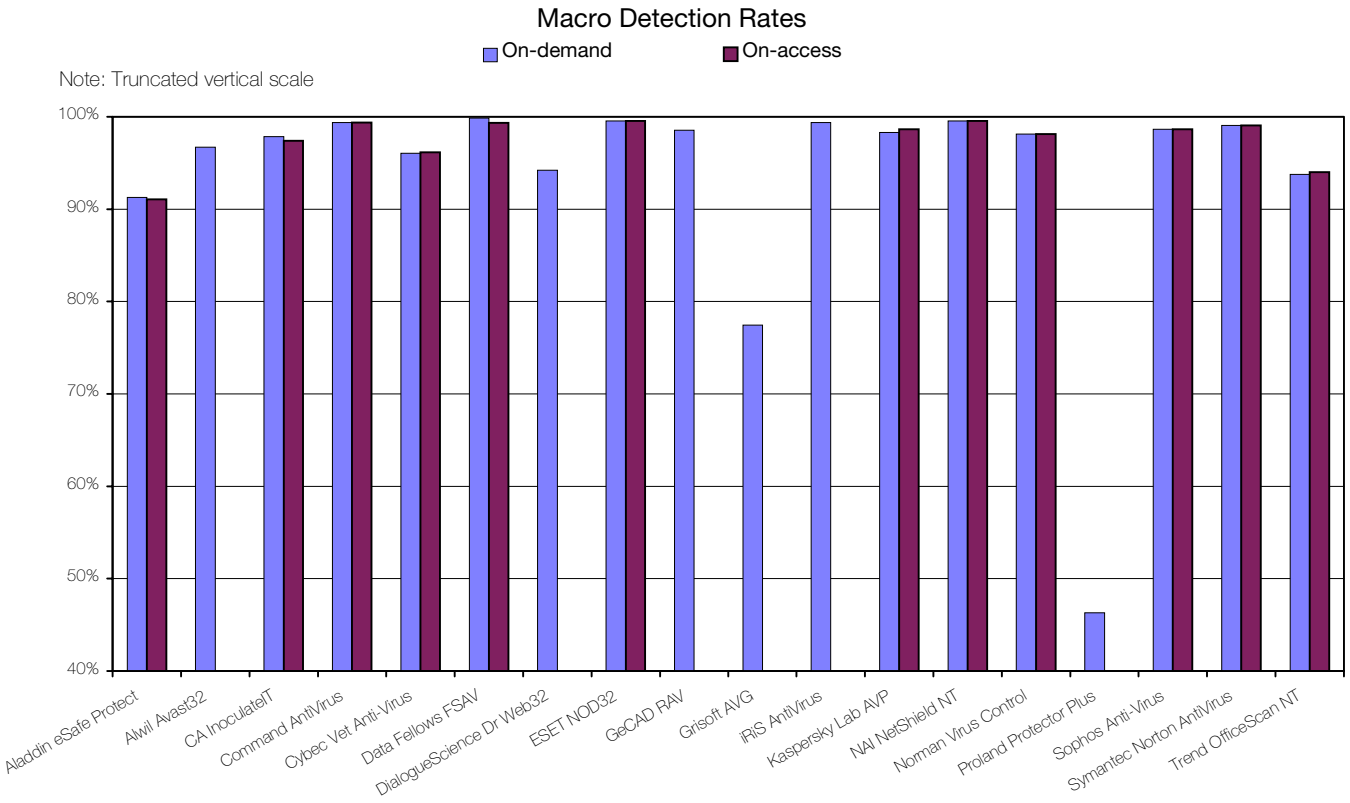
Another relative newcomer from Eastern Europe, *GeCAD's RAV* has performed well through recent comparatives. Showing steady improvement, it has not yet reached VB 100% standard but is clearly striving for it. Marburg is something of an Achilles heel for *RAV* at present – it missed all samples in the Polymorphic test-set but managed to detect five of the eighteen in the ItW File set.

Speedy it is not, but nor is it unusably slow. With some reliance on heuristics, it is not unusual that it produces a number of false-positives (eight this time). Not having an on-access component, there is little else to comment on.

Grisoft AVG v5.0 build 1238

ItW Overall	99.1%	Macro	77.4%
ItW Overall (o/a)	n/a	Polymorphic	93.3%
ItW Boot	90.5%	Standard	87.9%

Another Eastern European product striving for wider acceptance, *Grisoft's AVG* was dealt a cruel hand in the In the Wild Boot test, failing to detect any viruses on the diskettes with invalid BPBs. The only other mark against it from the ItW tests was that it missed the Win95/Fono VxD. Oddly, this version scored 5% lower on the unchanged Polymorphic test-set than the DOS version did in January.



AVG has several pre-configured scanning configurations and, as such, none is clearly the 'default' mode. All detection and speed tests in this review were run in the so-called 'Complete test' mode. This results in *Grisoft's* speed appearing slower than in previous *VB* reviews. Happily, no false positives were reported.

iRiS AntiVirus v22.16 6 Jan 1999

ItW Overall	100.0%	Macro	99.4%
ItW Overall (o/a)	n/a	Polymorphic	99.1%
ItW Boot	100.0%	Standard	99.7%

The second of the Israeli contingent in this comparative, *iRiS AntiVirus* picked up its third VB 100% award. The small handful of misses on the rest of the test-sets were due to the most recently added samples, apart from the eleven Cryptor.2782 samples missed in the Polymorphic test-set.

On the Clean test-set, *iRiS AntiVirus* returned a mid-range throughput and no false alarms. The *NT* product still does not sport an on-access component, and the user interface, whilst functional and familiar to users of earlier *Windows* versions, is starting to show its age.

Kaspersky Lab AVP v3.0.128 29 Dec 1998

ItW Overall	100.0%	Macro	98.3%
ItW Overall (o/a)	100.0%	Polymorphic	100.0%
ItW Boot	100.0%	Standard	99.7%

Regular readers of recent comparative reviews will not be surprised to see *AVP* from *Kaspersky Labs* achieve yet another VB 100% award. The detection levels against the non-ItW test-sets should not be surprising either.

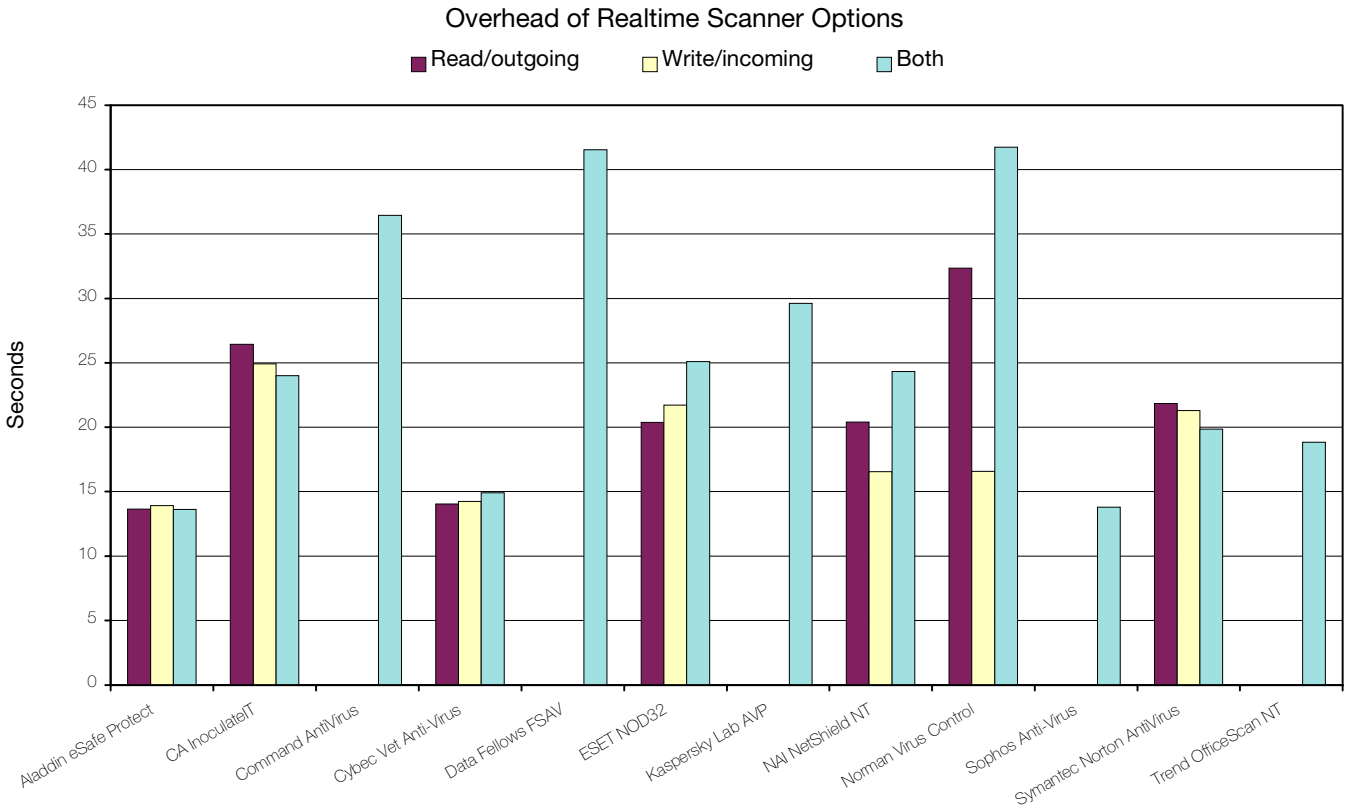
What *is* surprising, perhaps, was that a slightly greater number of macro viruses were detected on-access than on-demand. Throughput of the Clean test-set is at the top end of a large group of middling performances and overhead approached 100%, which may sound daunting but was certainly not the highest recorded.

NAI NetShield NT v4.0.2.4008

ItW Overall	100.0%	Macro	99.5%
ItW Overall (o/a)	99.6%	Polymorphic	96.7%
ItW Boot	100.0%	Standard	99.7%

This is the first showing of an *NT* product from *NAI* powered by the *Dr Solomon's* engine. Characteristic of the high detection rates of that engine in its former incarnation, a VB 100% performance was returned against the ItW Overall test-set.

Interestingly, the screen savers (SCR files) infected with Win95/Marburg and TPVO.3783.A, which were troublesome to some other products, were detected on-demand, but not on-access. Failure to check SCR files by default, thus missing a large chunk of the Marburg samples therein, also explains much of the uncharacteristically low score against the Polymorphic test-set. The other 'problem' *NetShield*



faced in that test-set was with W97M/Splash.A. This virus' practice of morphing its code by inserting ever more random comments into itself has been noted in previous reviews as causing trouble for several products.

NetShield's traditionally very slow speed has been improved markedly by the change of engine. Given this, it should not be surprising that its high overhead has reduced commensurately. There were no false alarms.

Norman Virus Control

ItW Overall	100.0%	Macro	98.1%
ItW Overall (o/a)	98.4%	Polymorphic	100.0%
ItW Boot	100.0%	Standard	99.7%

Another product with an all but unbroken string of VB 100% awards, *Norman Virus Control* (*NVC*), provided a typically staunch, Scandanavian showing. The gloss of its VB 100% award was slightly tarnished by its on-access component ignoring SCR and VxD files, thereby failing to detect TPVO.3783.A and Win95/Marburg in several of the former and one sample of Win95/Fono in the latter.

Testing on-access detection was complicated slightly because, as with *CA's InoculateIT*, *NVC* only has detect on read and/or write operations. This was simply resolved by copying the test-set from the server, scanning on file writes and deleting infected files. Also as with *InoculateIT*, detection in this mode was not as thorough as on-demand and repeat testing led to further detections. This took several iterations to converge on three successive runs with no further detections occurring, but performance was still lower than in the on-demand case.

NVC's mid-range throughput on the speed test is not a reliable guide to its overhead. Oddly, its overhead is significantly lower when only intercepting write operations than in other modes. No false positives were recorded.

Proland Protector Plus

ItW Overall	58.6%	Macro	46.3%
ItW Overall (o/a)	n/a	Polymorphic	10.7%
ItW Boot	57.1%	Standard	54.1%

Indian *Protector Plus* was the newest entrant to *VB's* comparatives in the previous *NT* scanner round-up in September 1998. The current performance represents an improvement of 20–25% over that first showing.

The 'on-line scanner' seemed to be more of a scheduler for the on-demand scanner. More could not be decided however, as the initial scan that starts immediately on enabling this component always caused Dr Watson to object in its strongest terms, stopping the service.

As in the previous *NT* comparative, *Protector Plus* blitzed the field in the speed tests. Outpacing *Vet* by more than 25% would be the envy of most anti-virus developers, but coupled with this product's detection rate, such speed provides little comfort. Add the 61 false-positives against the Clean test-set and the formula is even more lopsided.

With lower than 50% detection of macro viruses (presumably benefiting from its default 'detect suspicious macros' option) and clear stability problems, this is a product with quite some maturing ahead of it.

Sophos Anti-Virus v3.17

ItW Overall	100.0%	Macro	98.6%
ItW Overall (o/a)	100.0%	Polymorphic	100.0%
ItW Boot	100.0%	Standard	99.2%

Another regular recipient of VB 100% awards, *Sophos Anti-Virus* (*SAV*) was not to disappoint on this outing. As with several other products, the relative stasis of the non-ItW test-sets since the major update prior to the January DOS comparative has allowed *SAV* to catch up to its more typical performance on those tests. The viruses missed were the very newest added to the test-sets, plus Positron which SAV only detects in 'full scan' mode.

On-access and on-demand detection was identical – as alluded to earlier, something of a rare occurrence. *SAV's* speed is quite respectable on NT, resulting in a throughput of almost 2500 KB/s – a result that is somewhat anomalous with *SAV's* speed on other platforms.

Symantec Norton AntiVirus v5.01.01

ItW Overall	99.8%	Macro	99.1%
ItW Overall (o/a)	99.8%	Polymorphic	98.7%
ItW Boot	98.8%	Standard	99.5%

As with several other recent top-performers, *NAV's* run at another VB 100% award fell foul of Fono. *NAV* detected the EXE samples of this virus but missed its VxD and a Fono-infected diskette boot sector. Aside from this, *NAV* detected all but one or two viruses in each of the other test-sets. It still misses a single EXE sample of Marburg in the Polymorphic test-set. The product's results were the same under both on-access and on-demand conditions.

Returning a throughput rate a little over 1000 KB/s placed *NAV's* speed solidly in the middle of the pack. No false positives were reported.

Trend OfficeScan NT 98.5 VPN 489

ItW Overall	99.7%	Macro	93.8%
ItW Overall (o/a)	n/a	Polymorphic	96.8%
ItW Boot	97.6%	Standard	98.7%

The first showing of *Trend's OfficeScan* in a *VB* review shows the promised improvement in detection rates and speed seem to have been realized. It will take time to tell if the product's stability has improved, though it reported two false alarms. VB 100% status was denied by Fono's VxD and boot sector forms, and the ancient V-Sign boot virus.

OfficeScan provides no on-access boot sector scanning, save at shut-down – a feature that all products should provide. Given *NT's* legendary shut-down and restart speed, running the on-access Boot test via this mechanism was not even considered an option.

Following in the footsteps of *InoculateIT* and *NVC*, testing on-access detection of the viruses in the file-based test-sets with *OfficeScan* required copying the test-sets from the server to a local disk. This was required for a different reason from that of those products. *OfficeScan* adamantly refused to intercept file I/O requests involving remote files. This is an intriguing way to require your users to install your product on both servers and workstations. This philosophy of ignoring network file sources extends throughout the workstation product, with network drives never appearing in selection lists and the like. Oddly, however, the context menu in Explorer lists *OfficeScan* as an option for network drives and folders, and *OfficeScan* happily obliges by scanning the selected object.

As a beta version was submitted for testing, it may seem churlish to point out stability issues, but some things should be 'too obvious'. For example, *OfficeScan* adds an option to scan a drive or folder to the context menus in Explorer. This consistently disappeared following the first reboot after installation, thus removing the only available method of scanning the test-sets stored on the server.

Conclusion

Several products missed small numbers of *Excel* macro viruses because they do not look at a wide enough range of file extensions. The extension XL? is a highly recommended one to add to default extension lists, if the product supports wildcards in that list. If it does not, then users have to pray the developer is keeping up with the state of play or be very alert themselves. These results suggest some are not. There are related issues with SCR and VxD files.

A surprising observation was that some products do not provide a 'deny access' action for infected objects. A product that leaves system administrators trusting that their users will 'do the right thing' when warned of a virus, seems unduly optimistic to me.

Technical Details

Test Environment: Server: *Compaq* Prolinea 590, 80 MB of RAM, 2 GB hard disk, running *NetWare 3.12*. Workstations: Three 166 MHz Pentium-MMX workstations with 64 MB of RAM, 4 GB hard disk, CD-ROM drive and a 3.5-inch floppy, all running *Windows NT v4.0 (SP4)*. The workstations could be rebuilt from image backups and the test-sets were in a read-only directory on the server. All timed tests were performed on one machine that was not connected to the network for the duration of the timed tests, but otherwise configured identically to the detection test condition.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/199903/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.c om/Comparatives/Win95/199801/protocol.html.

PRODUCT REVIEW

Trend Micro ServerProtect for Windows NT v4.6

Martyn Perry

VB last conducted an in-depth, standalone review of *Trend ServerProtect* (*SPNT*) about a year ago (see *VB*, February 1998, p.21). Read on to discover what is new since then. *SPNT* is serialised and licensed on a per server basis. The licence covers the number of users defined in the shipping documentation.

Presentation and Installation

The review product arrived in a boxed set of three manuals, and a CD. The manuals comprise ServerProtect 4.5 Quick Start Guide, ServerProtect 4.5 Administrator's Guide and an addendum called Managing ServerProtect from the Trend VCS. The documentation (version 4.5) keeps referring to disks but the software is shipped on CD-ROM. As part of the documentation, there are electronic copies of on-line help, an Administrator's guide in *Acrobat* format and a Virus Encyclopaedia.

Following automatic loading of the CD, the initial screen of the installation provides options to view information about *Trend* and its other products. Choosing Install option displays a list of the various products on offer. Your reviewer chose *ServerProtect for NT Server* and selected the Install button. The familiar InstallShield is used to provide the installation.

The Licence Window is followed by a window for User Information. This comprises Name, Company (picked up from machine) and the licence number on the licence card. If only required for 30 days evaluation, this can be left blank. The licence number must be filled in correctly to continue – slightly obvious, but it is important to include the hyphen character between the groups of four characters. The font used for the licence number makes the hyphen look like an elevated full stop.

A destination is requested for the software, the default being C:\Program Files\Trend\SProtect. The next choice selects whether to create a Personal or Common Program Group (common is the default). Then I could either create a new folder ServerProtect (default) or use an existing one.

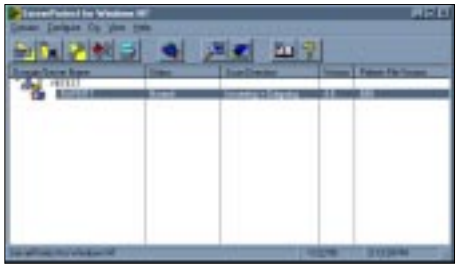
The ServerProtect Information Server must be selected next. The default is to choose an existing one, but as this is an initial install, it is necessary to set the current server as the Information Server, which automatically selects the Server's name. The next step is to choose a name for the ServerProtect Domain. Then I had a choice of what action to take on detecting viruses when handling a real-time scan. The defaults are to move the suspect file to the SUSPECT directory and the Real-Time scan operates for incoming files. For this review, these were changed to Leave Alone and Off, respectively. The other options available are Rename (changes extension to VIR), Delete or Clean for Actions and Incoming/Outgoing for Real-Time scan.

After copying the files across, the boot sector on each of the drives is checked. In order to activate the *ServerProtect* service, it is necessary to log on to an account. The default is the System account, although another can be used if required. Finally, I was prompted with the option to view the readme file which gives details of the upgrade history.

ServerProtect 4.6E

ServerProtect operates a security domain structure. Within a domain there can be multiple servers. To group multiple *ServerProtect* domains together, Information Servers are used. The Information Server (IS) provides a common location where configurations of member domains are shared and stored. This also provides security facilities for validating passwords and logon restrictions.

There is a separate utility (IS Utility) which is used to manage the Information Servers. This utility provides for the backup of the IS along with the complementary restore function, the merging of multiple IS to form larger groups and the ability to create new Information Servers from existing *SPNT* Servers. *ServerProtect* itself provides the usual range of facilities expected from a Server scanner, i.e. Real-time, Manual and Scheduled scanning.



Scanning Options

All files or specific file extensions can be selected for scanning. The default list is BIN, COM, DLL, DOC, DOT, DRV, EXE, OVL, SYS, XLS. Additional extensions can be added, and I think that they will need to be. The Virus Behaviour monitor option was selected in all the tests.

The choices to run are – no scanning, scan incoming only, outgoing only, and both incoming and outgoing. In the event of a virus being found, one of the following actions can be chosen – Delete, Clean, Leave Alone, Rename with a VIR extension, or Move to a quarantine directory (default C:\Program Files\Trend\Sptrotect\SUSPECT).

The Manual option is used for local scanning and specific mapped drives on servers. Manual Scanner Options allow for scanning of All Files, or Specific files. The default list is the same as that for Real-time. There is a separate selection for compressed files ARJ, LHA, ZIP and MS-COMPRESS extensions. The choice of actions in the event of a virus infection is the same as for the Real-time scan.

The scheduled scan can be configured to run Daily, Weekly or Monthly. The time of day, the day of the week and the date in the month can be altered. The default selection is all local drives and directories with selected file types – BIN, COM, DLL, DOC, DOT, DRV, EXE, OVL, SYS, XLS. Here, the virus action choices are the same as before with the exception that there is no clean file facility.

Administration

A password unlocks the domain before configuration changes can be made to a server. Then the scanners can be reconfigured as needed. There is an on-line virus encyclopaedia to provide additional information as to what actions may be needed to clear a particular virus. If a virus is detected, *ServerProtect* provides a notification service via one or more of the following methods – message box, printer, pager or email.

SNMP Trap

In each case, a predefined set of variables can be chosen to state the virus detected, the location of the file, the User's name and the type of scan performed. There is an additional scanning option to provide exception lists – lists of files to be ignored when scanning. These can include directories to ignore, specific files which are giving false positive reports, removing scan patterns which are giving problems, write protect directories and even protecting an area where viruses can be stored for reference purposes.

To help with analysis, there are three event logs, namely: System, Security and Application which keep track of the various activities that occur on a specified computer. Filters can be set to limit the data viewed at any one time.

Trend VCS

Trend has added an additional component to assist with managing *ServerProtect* along with other anti-virus products within a domain. The product is called *Trend VCS* (*Virus Control System*). It is made up of two components, namely *Trend VCS Server*, which installs on a *Windows NT* Server and *Trend VCS Agents*, which typically install on the same server as an anti-virus product.

This then allows products such as *ServerProtect* and other scanners to be administered remotely using a Web browser. It makes use of *Microsoft's* Internet Information Server to manage this facility. During testing the server gave a number of out of memory errors even when *ServerProtect* and *Trend VCS* were the only programs running.

Trend VCS does give version information, though it was reporting v422 when the installation version was v488. Pattern updates can be scheduled from *Trend's* web site. *VCS* provides the various features required for managing an anti-virus regime across multiple sites and servers by making use of Intranet technology to provide the communication mechanism. These features include Notification, SNMP and SMTP setups, Outbreak Alert, Administration, Agent Setup, Upgrade and Help. Many of the *ServerProtect* functions can be administered from a central point. These may include Status information, Scanner configuration, Scan Now, Deploy Pattern, Remove Server and View Log.

Updates

The shipped CD version sent for review contained the virus Pattern File 450, whereas the tests were performed using the pattern file 488 supplied in LPT$VPN.488.

Updates are available by accessing the update site by FTP or from BBS via modem or from another server. There are options to configure these sources. There is an additional option to update from a floppy. The implication is that it is sufficient simply to insert the disk in the drive and proceed. Not so fast – this file is 1.63 MB, which means that it does not fit on a floppy. Copying it to hard drive is no good – it cannot be read from a hard drive on a local server.

The trick is to copy the update file manually to the *ServerProtect* directory and rename the current version. To activate the new version, close *ServerProtect* then stop the ServerProtect Service. Restart the ServerProtect Service and reload the program – the updated version is displayed.

Scanning Overhead

To measure the extra work performed in detecting a virus, a diskette comprising 26 EXE and 17 COM files was scanned. The scan was repeated with the files infected with Natas.4744 virus. Full results of all the tests can be seen in the summary box. The times were as follows. It took 4 minutes 57 seconds to scan 5500 clean files. Two false positives were thrown up.

Detection Rates

The scanner was checked using the usual *VB* test-sets – ItW, Standard, Polymorphic, Macro and Boot Sector. Detailed results are in the summary. The tests were run using the default scanner file extensions supplied and the scan action option was to delete infected files. The residual file count was then used to determine the detection rate.

It came as some surprise that, during the boot sector test, *SPNT* missed one sample of V-Sign. Against the Polymorphic test-set it missed all SCR versions of Marburg, plus 70 samples of Gripe.1985 and 53 samples of Cryptor.2582. Twenty samples were missed against the Standard test-set, while in the Macro test *SPNT* missed 463 samples, many due to lack of file extensions e.g. MDB, in the file list.

Re-running the tests with all files selected resulted in some improvements, with *SPNT* achieving a 100% detection rate in the ItW test-set. In the polymorphic tests, it identified the SCR versions of Marburg and improved slightly against the Macro test-set, missing 446 instead of 463 – primarily due to the identification of Access macros. Against the Standard test-set, results were left unaffected.

Real-time Scanning Overhead

To determine the impact of the scanner on the workstation when it is running, the following time test was run. 200 EXE and COM files of 21.24 MB bytes were copied from one folder to another using XCOPY. The folders used for the source and target were excluded from the scan, avoiding the risk of a file being scanned while waiting to be copied.

The default setting of Maximum Boost for Foreground Application was used for consistency. Due to the different processes which occur within the server, the time tests were run ten times for each setting and an average taken. The tests were as follows:

- Program not loaded: ServerProtect service off, establishes the baseline time for copying files on the server.
- Program installed but not scanning, Real-time disabled: tests the impact of the application in a quiescent state.
- Program loaded, Real-time enabled, Incoming Files only: tests the impact of the scan on incoming files.
- Program loaded, Real-time enabled, Outgoing Files only: tests the impact of the scan on outgoing files.
- Program loaded, Real-time enabled, Incoming and Outgoing Files: tests the impact of the scan for incoming and outgoing files.
- Program loaded, Real-time enabled, Incoming and Outgoing Files; Manual scan included: tests the full impact of scan for incoming and outgoing files as well as the normal scanning of files.
- Program unloaded: run after the server tests to check how well the server is returned to its former state.

Summary

The tests showed significantly improved results over those of *SPNT v4.5*. The earlier problem with selecting individual directories has been fixed. The false positive count against the Clean test-set has improved, reducing from seven to two. One false alarm is still the same and a file which used not to be a problem is now falsely detected.

Scan speed has improved markedly. The overhead when scanning the floppy has been cut by 50%. Similarly, the time to scan the Clean test-set has reduced from just over eleven minutes to just under five minutes. The overall detection rates are slightly better across the board except against the Macro test-set. Curiously, the one missed boot sample was detected a year ago.

The default file selections have stayed the same despite new viruses, over the last twelve months, attacking different file types, e.g. MDB Access data files and SCR screen savers. However, administrators find it time-consuming enough to deploy new updates, let alone keeping abreast of the latest threats from the virus community. They rely heavily on their anti-virus supplier to issue products which can be deployed with minimum reconfiguring. This puts the vendor in a difficult position, since the obvious alternative is to check all files with the attendant performance issues.

The provision of *Trend's Virus Control System* using an Intranet is an interesting development. I am sure other vendors will start trying to integrate virus management into an overall desktop management environment. The possible downside is committing to a particular vendor's standard for communication, even if it is the market leader. All in all, it looks as if someone has been busy improving *SPNT's* performance and introducing new ideas. It just remains for the macro detection to be brought under control.

Trend ServerProtect for NT v4.6

Detection Results

Test-set[1]	Viruses Detected	Score
In the Wild Boot	81/82	98.8%
In the Wild File	724/738	98.1%
Standard	1020/1040	98.1%
Polymorphic	14051/14444	97.3%
Macro	2169/2632	82.4%

Overhead of On-access Scanning:

The tests show the time (in seconds) taken to copy 200 COM and EXE files (20.5 MB). Each test was repeated ten times, and an average taken.

	Time	Overhead
Not loaded	13.4	–
Loaded, disabled	13.8	3.2%
— + incoming, no scanning	20.6	53.66%
— + outgoing, no scanning	20.6	54.0%
— + both, no scanning	26.9	101.0%
— + — + manual scan	40.6	203.7%
Program unloaded	13.7	2.6%

Technical Details

Product: *Trend Micro ServerProtect for NT Server v4.6.*

Developer: *Trend Micro Inc*, 10101 N De Anza Blvd, 4th Fl, Cupertino, CA 95014. (A UK office opens in April 1999).

Vendor: Contact Penny Brennan; Tel + 44 (0)1189755188, fax +44 (0)118 9314145, email feliciano_rivera@trendmicro.com and WWW http://antivirus.com/.

Price: £1,050 for a server with up to 50 users.

Hardware Used: Workstation: *Compaq* Prolinea 590, 80 MB of RAM, 2 GB hard disk, running *NT Server v4.0 (SP3).*

[1]Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/DOS/199901/test_sets.html.

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, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YP, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

Virus Bulletin, 18 Commerce Way, Woburn, MA 01801, USA

Tel (781) 9377768, Fax (781) 9320251

END NOTES AND NEWS

Full details of the upcoming VB'99 conference in Vancouver, British Columbia can be found at http://www.virusbtn.com. The 9th annual *Virus Bulletin* conference will run from Thursday 30 September to Friday 1 October at the Hotel Vancouver. Contact conference co-ordinator Jo Peck for details about new sponsorship opportunities and exhibition space; Tel +44 1235 555139, fax +44 1235 531889, or email Joanne.Peck@virusbtn.com.

***Sophos* will be hosting an introductory computer virus workshop on 17 March 1999 to be followed on 18 March by an advanced session.** The two-day course will be held at the organization's training suite in Abingdon, UK. To register for a place, contact Karen Richardson; Tel +44 1235 544015, fax +44 1235 559935, or find more information at http://www.sophos.com/.

CompSec'99, the 16th World Conference on Computer Security, Audit and Control will take place from 3–5 November 1999 at the QE2 Centre, Westminster, London, UK. For more information contact Tracy Stokes at *Elsevier*; Tel+44 1865 843297, fax +44 1865 843958, or email t.stokes@elsevier.co.uk.

Washington DC, USA is the location for the 8th *USENIX* Security Symposium, to run from 23–26 August 1999. The event is planned around two days of tutorials followed by two days of technical sessions, papers, talks, works-in-progress, panel discussions and a product exhibition. For further details about the conference visit the *USENIX* web site http://www.usenix.org/events/sec99/cfp/.

***NetSec'99*, the 9th *Computer Security Institute* (*CSI*) Annual Network Security Conference,** is to be held from 14–16 June 1999 in St Louis, Missouri at the Hyatt Regency Hotel. Over 1500 computer and information security professionals are expected to attend the conference and its concurrent exhibition. For the latest calendar of events or more details on the conference, contact *CSI*; Tel +1 415 9052626, fax +1 415 9052218, email csi@mfi.com or visit the *CSI* web site at http://www.gocsi.com/.

Symantec* has announced the recent release of *Norton AntiVirus for O/S2. The product is currently available on the *Norton AntiVirus* Solution CD v3.03. For further information and pricing details, see http://www.symantec.com/.

InfowarCon'99 will take place at the Copthorne Tara Hotel in London from 27–28 May 1999. On Wednesday 26 May optional, full-day tutorials will be held. The conference focuses on military operations, infrastructure protection, and the growing threat of high-tech terrorism and espionage. It is aimed at corporations, infrastructure firms, and finance, military, intelligence and law enforcement organizations. Registration is from 7am on Thursday 27 May. For more details about the conference, contact organizers *MIS* in London; Tel +44 171 7798944, fax +44 171 7798293.

Data Fellows* announces the release of the fourth generation VPN product, *F-Secure VPN+ v.4.0, which began shipping in February. The supported platforms are *Windows NT v4.0* and *Windows 95*. The new version is priced at $59 per user for a 100-user licence, $495 for a server licence, $2495 for a gateway licence and $4990 for an enterprise gateway. For more information, contact Product Manager Topi Hautanen: Tel +358 9 859900, fax +358 9 85990599 or email Topi.Hautanen@DataFellows.com.

The 13th Annual Vanguard Enterprise Security Expo'99 is to be held in Dallas, Texas from 6–11 June 1999. For more information about the conference and the concurrent exhibition, contact *Vanguard Security Professionals*; Tel +1 714 9390377, fax +1 714 9390273, or visit the web site http://www.vipexpo.com/.

Until 31 March 1999 *Secure Computing* (US) is to distribute its secure access product for free. *SecureWire 1.6 for NT* is available via the *Secure Computing* and *Microsoft* TechNet web sites. The offer includes the software, a 25-to-unlimited user licence and free customer support for 30 days. For more information contact Sales Manager Tony Caine; Tel +44 1753 826000, fax +44 1753 826001, email tony_caine@securecomputing.com or visit the company web site http://www.securecomputing.com/.

iRiS Software* announces the release of *Macro Defender* (*MD*) *v1.0 for *Word* and *Excel* protection from macro virus attacks. The product works with *Office 97* and *Office 95*, protecting users from macro viruses inside email attachments as well as in files downloaded from the Internet. *iRiS* claims that, unlike traditional anti-virus products, *MD* requires no upgrades. *iRiS Macro Defender v1.0* is available for downloading from http://www.irisav.com/ for $34.99.