JUNE 2003

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

ISSN 0956-9979

IN THIS ISSUE

VIRUS PLAYING THE P2P GAME

As the popularity of P2P technology continues to soar, the AV industry can expect to see an increasing number of P2P worms. W32/Bibrog keeps the user busy playing a shoot-the-celeb game while it drops its files. Rex Plantado investigates Bibrog.C and concludes we are lucky it failed to execute its payload.

page 4

THE XP PLAYGROUND

While WinNT.Adonai claimed to be 'the world's first virus able to jump to ring 0 on *NT* machines', it merely dumped a .DLL and a .SYS file to disk and played with the PC speaker. Matters are a little different with WinXP.Che, however. The author of this virus has proved that a ring 0 memory resident virus for *Windows 2000* and *XP* is not impossible to create. Mihai Chiriac explains why.

page 7

THE XP COMPARATIVE

Time, once again, for Matt Ham to eXPose the VB100% winners, losers and the must-try-harders.

page 15

'We all have an insatiable monster in us that needs to know.'

Carole Theriault
Sophos Plc

CURIOSITY KILLED THE CAT

Email is being targeted by opportunists whose ethics are equal only to the stereotypical medallion-wearing used-car salesman. I'm talking about the creators of email-aware viruses who employ cunning tactics to encourage users to run an infected attachment. This act of persuasion is often referred to as 'social engineering' – a pretty lousy term to describe the act of lying, conning and misleading innocent computer users. But are virus writers all that clever, or is it the users' baser instincts that make so many of them double-click on an infected attachment?

Several aspects of an email can be modified by a virus writer to lend their scam more credibility: the sender address, the subject line, the message and the name of the attached file. The approach varies, appealing primarily either to our feelings, our reason or our beliefs.

Lovegate-E is a typical example of a worm that attempts to use sex to exploit unwary users. The message is a stanza from Rudyard Kipling's *If* – which has no obvious ties with the files the worm may append: 'Britney spears nude.exe.txt.exe' or 'hardcore game-.pif'. Clearly, the attachments are aimed at the groins of bored computer users. This tactic is pretty fool-proof – there will always be a small percentage of the population who are unable to resist the invitation to view naughty pictures.

A version of Love Bug is now three years old but is memorable, not least for using one of the sneakiest ploys to encourage users to run its attachment:

```
Subject line: Mothers Day Order Confirmation
Message: We have proceeded to charge your credit
card for the amount of $326.92 for the mothers
day diamond special. We have attached a detailed
invoice to this email. Please print out the
attachment and keep it in a safe place. Thanks
Again and Have a Happy Mothers Day!
mothersday@subdimension.com
Attachment name: mothersday.vbs
```

Is this tactic not the pinnacle of deception? Recipients probably felt outrage as they contemplated being ripped off, worry as they wondered how to rectify the situation, and guilt when it dawned on them that they didn't love their mother enough to have bought her an expensive mother's day gift.

Alas, it seems that mass-mailing infectors do not need to do anything spectacular to get people to double-click. The Sobig worm, which caused quite a stir in January this year, used a number of bland subject lines and attachment names. Subject lines were 'Re: Movies', 'Re: Sample', 'Re: Document' or 'Re: Here is that sample'; the attached filename was chosen from a list which includes 'Document003.pif', 'Sample.pif', 'Untitled1.pif' and 'Movie_0074.mpeg.pif'.

The use of 'Re:' in the subject line is interesting – wouldn't recipients look at the subject line and say to themselves, 'I have never sent anyone an email with that subject line'? Moreover, the sender address was always the same: big@boss.com, making it easy to warn the general public about this email scam. Sadly, neither of these points seemed able to stop many people from launching the file.

Whether the authors of mass-mailing infectors use specific tactics or not, users seem pretty trusting, or rather, curious. Tempting curiosity is the virus writer's secret weapon because we all have an insatiable monster in us that needs to know. The message can promise a picture, a movie, information, a game or a screensaver, and we succumb to a base urge to see it.

Being wary of all unsolicited email and attachments is theoretically possible, but highly impractical. It is better to have up-to-date anti-virus installed at the gateway that strips suspicious or infectious attachments. Coupled with anti-virus on your desktop (because mass-mailers can also travel via other means), you have a fool-proof system. Whatever words are employed to dupe you into double-clicking, the software, oblivious to persuasive tactics, will remain immune.

NEWS

TREND MIND THEIR Ps... IF NOT THEIR Qs

Putting a new spin on the *Sesame Street* catchphrase, 'this programme was brought to you by the letter … P', *Trend Micro* quarantined the letter P last month. Technology news website www.crn.com reports that *Trend* alerted its customers after it was discovered that a bug in an update for email security product *eManager* resulted in the blocking of *all* incoming email containing the letter P. The bug was discovered shortly after the release of update Rule #915 and *Trend* swiftly issued Rule #916 to fix it just an hour and a half later. Customers wishing to retrieve emails that were inadvertently quarantined are advised to call *Trend Micro* technical su**ort.

On a different note, we cannot let this month slip by without congratulating the team behind *Trend Micro*'s latest advertising campaign. The slightly less-than-subtle advertisements, currently appearing in print format, leave it to the reader to imagine in what activity a pointedly yellow-suited (or, more accurately, birthday-suited) pair are engaged. An online animated GIF leaves significantly less to the imagination. This may certainly be a case of 'if you don't have anything good to say about yourself, have a dig at the competition', but hats off to them for their sheer audacity, and (ahem) bare-faced cheek.

SCHOOL WITHOUT THOUGHT?

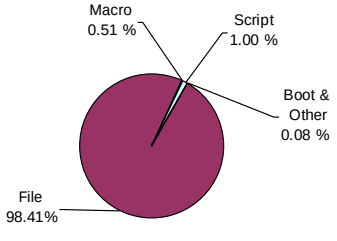
The University of Calgary has announced very proudly on its website that a new undergraduate course will 'focus on developing malicious software such as computer viruses, worms and Trojan horses that are known to wreak havoc to the tune of billions of dollars worldwide on an annual basis'. Members of the AV industry repeatedly assert the importance of education when it comes to secure computing, but surely a course that focuses on *developing* malware is an extreme case of barking up the very wrong tree. Dr John Aycock, professor for the course, thinks not. He believes that, by 'looking through the eyes of the people who develop viruses, [the] students will learn what their targets actually are and what needs to be protected.' While busily learning how to compile their own malicious code, undergraduates on the course will also study 'legal and ethical issues' – *VB* is intrigued as to what students learn on this part of the course. The University claims that 'this course is just one more way [in which] the … University of Calgary is helping develop students' skills as they become the leaders of tomorrow.' The AV industry had better brace itself if we are to expect a future in which virus writers are the leaders of tomorrow.

Prevalence Table – April 2003

Virus	Type	Incidents	Reports
Win32/Opaserv	File	6340	45.30%
Win32/Sobig	File	2250	16.08%
Win32/Klez	File	1863	13.31%
Win32/Dupator	File	1147	8.20%
Win32/Funlove	File	485	3.47%
Win32/Yaha	File	342	2.44%
Win95/Spaces	File	329	2.35%
Win32/Bugbear	File	200	1.43%
Win32/Magistr	File	123	0.88%
Win32/Lovgate	File	114	0.81%
Win32/Gibe	File	112	0.80%
Redlof	Script	102	0.73%
Win32/Lirva	File	60	0.43%
Win32/BadTrans	File	46	0.33%
Win32/SirCam	File	43	0.31%
Win32/Nimda	File	41	0.29%
Win32/Hybris	File	38	0.27%
Win95/Lorez	File	36	0.26%
Win32/Ganda	File	31	0.22%
Laroux	Macro	30	0.21%
Win32/Kovar	File	26	0.19%
Win95/CIH	File	22	0.16%
Win32/Braid	File	15	0.11%
Others[1]		200	1.43%
Total		13,995	100%

[1]The Prevalence Table includes a total of 200 reports across 66 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



VIRUS ANALYSIS 1

W32/BIBROG: WHAT YOU SEE IS (NOT) WHAT YOU GET

Rex Plantado
TrendLabs, Philippines

The increasing popularity of peer-to-peer (P2P) technology has resulted in a headache for the anti-virus industry: the advent of P2P worms. Since last year, a number of P2P worms have appeared in the wild – for example, W32/Surnova, W32/Lirva, W32/Lolol, W32/Benjamin and W32/Gnuman, to name just a few.

BIBROG

In March 2003, another family of P2P worms was released into the wild. This family of worms is more prolific than any previous P2P worms. W32/Bibrog, as it has been dubbed by AV vendors, is a destructive email- and P2P-borne worm with password-stealing capabilities.

W32/Bibrog propagates via *Microsoft Outlook* and *Outlook Express*, sending itself to all *Outlook* contacts disguised as a game, along with the text 'no es virus!' ('not a virus!'). In addition, the worm explicitly shares 20 copies of itself to P2P programs *KaZaA*, *ICQ*, *Grokster* and *Morpheus*. It uses the names of celebrities to disguise itself as a pornographic screen saver, in an attempt to increase its appeal to P2P downloaders. The worm attempts to delete files (MP3, MPG, EXE, DLL, GIF, JPG and ZIP) from the infected system.

At least five variants of this worm have been reported: W32/Bibrog.A, B, C, D and E. The variants behave in much the same way as each other, but differ in the filenames they use in P2P propagation, their embedded graphics files (which are used in the payload), file sizes, and the number of bugs or flaws in their code.

This analysis focuses on the W32/Bibrog.C variant (aka W32/Bibrog.B), since it has gained the most attention from the media and the AV industry.

The W32/Bibrog.C variant contains some flaws that are worthy of note – the worm's behaviour would be significantly different were the bugs to be fixed. Like the other variants, W32/Bibrog.C was written in high-level language Visual Basic, compressed with a UPX tool and appears to have originated in Mexico.

DON'T PLAY WITH ME

The first time it is run, W32/Bibrog.C launches a shooting simulation game (similar to the *Big Brother* game), in which the user targets moving pictures with a mouse click (see Figure 1).

The virus writer has managed to create a simple, yet exciting and addictive game. It uses pictures of personalities from http://www.laacademia.tv/ as the targets. The game increases the speed at which the pictures move across the screen and provides corresponding scores when the user targets the pictures successfully, making the game more realistic, challenging and addictive.

While the user is busy playing the game, W32/Bibrog.C drops its components into their respective target locations:

%Windows%\manzana.exe
%Windows%\mai.vbs
%System\academia.exe
%Start up folder%\itch.exe
%Start up folder%\itcj.exe

P2P UNDER ATTACK!

The author of this virus is aware of the weaknesses of a large percentage of peer-to-peer users – Hollywood celebrities, screen savers and pornographic materials are just some of them. W32/Bibrog.C drops 20 copies of itself into the following P2P share folders:

KaZaa\My shared Folder	ICQ\Shared
Grokster\My Grokster	Morpheus\My Shared Folder

The worm constructs filenames by combining the following celebrity names with the string 'porn screen_saver.exe':

Alessandra Ambrosia	Anna Kournikova
Britney Spears	Cameron Diaz
Charlize Theron	Christina Aguilera
Donna D'Erico	Halle Berry
Helena Christensen	Jenna Jameson
Jessica Alba	Karina Lombard
Kelly Hu	Kirsten Dunst
Kylie Minogue	Pamela Anderson
Salma Hayek	Sandra Bullock
Shakira	Stacey Keibler

For example, 'Shakira porn screen_saver.exe', 'Anna Kournikova porn screen_saver.exe', and so on. If all P2P clients are installed, the infected system is virtually sharing at least 80 copies of the worm to over one million online P2P users!

REBOOT AFTERMATH

The first time *Windows* reboots, W32/Bibrog.C runs ITCH.EXE and ITCJ.EXE. The first EXE opens the installed email client in order to send seven infected emails to all the contacts in the Address Book. The emails have the following format:

```
Subject: Fwd:La Academia Azteca
Message Body: La cacademia azteca (muy bueno)
!no es virus!
Attachment: academia.exe
```

The worm ensures that the files MANZANA.EXE and ACADEMIA.EXE exist in the system by dropping them into the Windows and System folders respectively every time *Windows* is restarted.

Showing a lot of aggression and tenacity in harming infected users, W32/Bibrog.C drops five HTM files in the My Documents folder in order to steal user account information from different online services and websites: hotmail.htm, yahoo.htm, msn.htm, citibank.htm and acafug.htm

W32/Bibrog.C checks the value of 'cuento' from its registry, 'HKCU>Software>VB and VBA Program Settings>ezzey>varia'.

When a certain count is reached, it is intended to delete all files with the extension DBF, MP3, MPG, EXE, DLL, GIF, JPG and ZIP – however, a bug in the worm's code means that this is never realized.

On the second reboot, ITCJ.EXE monitors certain addresses from the user's web browser. If any of the following targets are found, W32/Bibrog.C opens the forged HTM file as a replacement for the page opened by the user:

Figure 1. The game: shooting simulation.

Target	Replacement
hotmail.passport.com	%My Documents%\hotmail.htm
loginnet.passport.net	%My Documents%\hotmail.htm
mail.yahoo.com	%My Documents%\yahoo.htm
login.passport.net	%My Documents%\msn.net
www.fbi.gov	%My Documents%\acafug.htm
www.citibank.com/us/cards/	%My Documents%\citibank.htm

The file acafug.htm is a forged web page of the FBI's 'most-wanted' list. It uses the real names of people on the FBI's wanted list, but matches them with fake pictures gathered from http://www.laacademia.tv/. The hyperlinks on the web page are redirected to: http://www.korn.com/, http://www.drowningpool.com/ or http://www.spawn.com/.

The first four HTMs are forged login pages of legitimate web pages and Internet services. The worm gathers the username and password information that is entered by the user and sends it to pommedorato@yahoo.com, which is believed to be the worm's author.

The user account information is sent as a Yahoo greeting with the following format:

```
Title: Bear Hug
Brought to you by confetticards.co.uk
To: <user name>
<picture of Bear>
hola or @yahoo.com or hotmail
- <password>
```

where: <user name> is the name entered by the user and <password> is the entered password.

Sometimes the user is redirected to a non-malicious Mexican university website: http://www.cdj.itesm.mx/.

W32/Bibrog.C uses two image files alternately as wallpaper whenever *Windows* restarts. The files OSIRIS.BMP and QUIETTIME.BMP are located in the Windows folder (having been dropped on the first reboot).

The worm achieves this by modifying the Wallpaper value under the Desktop section of the WIN.INI file. However, this will not work under *Windows NT*, *2000* or *XP*.

IMPERFECTION DOES MATTER

First, W32/Bibrog.C is fully dependent on its run-time library, MSVBVM60.DLL. If this library does not exist in the system, this worm will not be able to function. In addition, the reverse engineering process is straightforward because the worm does not contain any event to trigger, anti-emulator or anti-debugging trick. The packer used by the worm is UPX, which is well known and widely

available, in order to unpack the program. A wiser option for the author of the worm would have been to use a packer or code protector that is not available on the Internet, or better still, to have created a tool to pack or protect the codes.

Secondly, the worm does not register its running code as service process. This could easily expose its presence on the system in the Task Manager (CTRL-ALT-DEL or CTRL-SHIFT-ESC). Aside from simultaneous execution and noticeable filenames (ITCH.EXE/ITCJ.EXE), the worm's copies in the start-up folder can easily be seen by looking in the Start>Programs>StartUp menu within *Windows*.

Thirdly, W32/Bibrog.C drops a two-byte text file, Main.VBS, and a copy to %Windows%\MANZANA.EXE, neither of which are used. The copy (%System%\ ACADEMIA.EXE) is used only as an attachment, wherein it can use either of the two files in the start-up folder. This only increases the likelihood of the worm being noticed.

The destructive payload (file delete) of the worm is never executed due to the bug in the worm code. The register counter is never incremented, which is the trigger for the destructive routine to execute.

Finally, W32/Bibrog.C's emails do not contain any exploit to execute the attachment automatically, such as the IFrame or 'Automatic Execution of Embedded MIME Type' vulnerabilities which have been used by some of the most successful worms. W32/Bibrog.C spams seven copies of infected emails to all recipients (including the infected user), every time *Windows* reboots. Its message appears to be forwarded but, unlike a valid forwarded email, there is no space between 'Fwd:' and 'La Academia Azteca (virus)!' in the subject line.

The worm constructs email messages in Spanish. This will immediately arouse suspicion in a recipient who does not speak Spanish. Without the automatic execution exploit, this limits the chances of the attachment being run by the user even further. If these flaws had been fixed, the worm would have had more flexibility and could have been more infectious and caused more damage.

CONCLUSION

Despite several flaws in its code, W3/Bibrog managed to deceive enough users (whether as a game, celebrity porn screen saver or faked Internet page) for it to propagate and it gained the attention of the AV industry.

We are lucky, however, that Bibrog failed to execute its destructive payload – otherwise, this could have been a very damaging worm like Loveletter, Opaserv, Nimda, or Klez. To avoid that happening in the future, we need to impress upon users the importance of being cautious when downloading any program from P2P networks or running any executable from an unverified email source.

W32/Bibrog

Type:	Email and P2P Worm with password stealing feature.
Payload:	Mails itself to all *Outlook* contacts; changes wallpaper; deletes files.
Removal:	Detect and delete the dropped executables; delete dropped files.

VIRUS ANALYSIS 2

XP, A NEW VIRUS PLAYGROUND?

Mihai Chiriac
SOFTWIN, Romania

New viruses appear every day; most of them are based on ideas which have already been proven to work. From time to time, however, new techniques are used in implementing viral engines, with varying degrees of success.

We all remember successful viruses like CIH, OneHalf and Nimda, which spread widely as a result of the new techniques that had been used in their development.

Some weeks ago we received a sample of WinXP.Che, the latest creation from the Czech virus writer Ratter, a member of the 29A virus-writing group. Immediately, we realised that we were facing a new type of virus: a ring 0 resident on-access infector for *Windows 2000* and *XP*, which disables the Windows File Protection by legitimate calls to *Windows*' own DLL files.

Previously we had seen WinNT.Adonai, 'the world's first virus able to jump to ring 0 on *NT* machines' (according to the author), but in this case no real work was done in ring 0; the virus merely dumped a .DLL and a .SYS file to disk and played with the PC speaker. However, matters are a little different with WinXP.Che.

TECHNICAL DESCRIPTION

In order to infect a computer, Che needs a little help: since it infects only drivers (.SYS files), the virus has to be loaded somehow. This can be achieved by a number of methods, ranging from writing a specific .SYS loader to overwriting an existing device driver file.

When the system loads an infected driver file, the main virus code (found at DriverEntry) receives control and attempts to find the OS Kernel (NTOSKRNL.EXE) in memory. For this operation the virus uses two hard-coded addresses: 0x80400000 and 0x804D0000, which are the default addresses for *Windows 2000* and *Windows XP* respectively. The use of these hard-coded addresses is one of this virus's weak spots: *Microsoft* has already changed the value for the SP1 version of *XP* to 0x804D4000. If the virus cannot find NTOSKRNL's base address it returns immediately to the host program. Otherwise, it starts scanning for the functions it needs.

To make the virus shorter (and to make its analysis a little more difficult) the author uses hashes instead of API names. The hashing algorithm proves itself reliable for alphanumeric entry values; it is also faster than the classic CRC32 algorithm. The virus imports the following eleven functions from NTOSKRNL:

KeNumberOfProcessors
ExAllocatePool
ExFreePool
KeServiceDescriptorTable
KeUserModeCallBack
ZwCreateFile
ZwAllocateVirtualMemory
ZwFreeVirtualMemory
ObReferenceObjectByHandle
ObQueryNameString
ObDereferenceObject

Then the virus checks the number of processors installed in the computer and bails out if there is more than one. The next check is the 'already-resident' marker: WinXP.Che writes an 0x72617461 ('atar' in ASCII) to offset 0x1C inside NTOSKRNL's memory area. If it is not already resident the virus proceeds, installing its file system hooks.

The virus allocates 1933 bytes of kernel memory by calling ExAllocatePool, then it moves itself to the newly allocated area and patches its own body with the API addresses that it fetched previously.

FILE SYSTEM HOOKING

When a ring 3 application calls CreateFile, kernel32.dll calls NtCreateFile (exported by ntdll.dll), which pushes the parameters onto the stack, causes the EDX register to point to the parameters, moves the ID for file creation/opening into the EAX register and goes to kernel mode for the real processing.

The *NT* kernel copies the parameters from the user-mode stack to the kernel stack and uses the value of EAX as an index in a table called the 'Service Dispatch Table'. Using this table, the *NT* kernel jumps to the specific function.

In order to locate the Service Dispatch Table the virus uses the legitimate (but undocumented) export KeServiceDescriptorTable. Then it simply saves the address of the original handler for ZwCreateFile and overwrites the entry in the Service Dispatch Table with its own address. As a result, the virus's handler is called on every file operation.

When it has finished installing the hook, the virus restores all the registers and returns to the original program.

INFECTING FILES

When an application tries to open a file, the virus's hook receives control. To prevent re-entry the virus makes use of semaphores: if the infection routine is busy the virus avoids calling it again; instead, it simply jumps to the next handler. Then it checks for the file extension – it must be '.sys' in order for the virus to infect.

If the file extension is '.sys', the virus locates the kernel32.dll image in memory and attempts to load another eleven functions to play with:

CreateFileW
CloseHandle
LoadLibraryA
GetProcAddress
FreeLibrary
GetFileAttributesW
SetFileAttributesW
CreateFileMappingW
MapViewOfFile
UnMapViewOfFile
GetFileSize

If the loading of any of these functions fails, the virus exits the infection routine and jumps to the next handler.

The interesting thing about the infection routine is that the infection is actually carried out in ring 3: the virus writer needed to call some functions that are not available in kernel mode.To do this, the virus gets a handle to the current thread (from the Process Environment Block), and from there, a pointer to the current process object; it copies itself to user-mode memory and uses yet another undocumented export from NTOSKRNL, the 'KeUserModeCallBack'. This function is used when a ring 0 routine needs to access data or to call a function in ring 3. *Microsoft* attempted to restrict the use of this function: one of the parameters is an index to a special table of predefined functions. However, the virus exploits the function and is able to call its own ring 3 code from the ring 0 file system handler. The advantage of using this function is its speed, which is critical in applications like file system filters.

RING 3 ROUTINE AND DISABLING WFP

The entire ring 3 routine is protected by Structured Exception Handling, so should any error occur, the virus simply jumps to the next handler. First it checks the initial characters of the file's name – they must be '\??\', in UNICODE – then the virus saves the file's attributes and continues.

Probably the most interesting part of this virus is its routine for disabling the Windows File Protection. We expected dirty memory patches and code injection – the virus did neither. It simply loaded the 'sfc_os.dll' library, retrieved the address of the fifth ordinal in the library and called it with three parameters: 0, the file's handle and 0xFFFFFFFF. After this simple library call the virus is able to modify the file as needed, without worrying about nasty message boxes and file replacing.

Next, the virus sets the file attributes to NORMAL, saves the file's size and creates a file mapping, then checks whether the file is a valid portable executable and checks that it has not already been infected. The infection marker is the dword 0x72617461 ('rata' in ASCII), stored in the 'Win32 Version Value' field of the PE Header. In this initialization, the virus does not check whether the file is a driver (the Subsystem is set to Native) or a regular PE executable or DLL, so it infects any valid PE file with a .sys extension.

The infection technique is classic: the virus appends itself to the last PE section by increasing its size and modifying its attributes (the section is marked as containing initialized code and as being writable and executable). The section is then aligned to the file alignment and the infection marker is set.

A very important part of the infection routine is the recalculation of the file's checksum: an attempt to load a driver with an incorrect checksum will fail, producing an error message: ERROR_BAD_DRIVER. The virus loads IMAGEHLP.DLL and attempts to find (using its hash value) the function 'CheckSumMappedFile' and calls it in order to re-validate the infected file, which is then written back to the disk.

THE BOX IS OPEN ...

By writing this virus Ratter has proved that a ring 0 memory resident virus for *Windows 2000* and *XP* is not impossible to create. Some features in the *NT* kernel that were not documented by *Microsoft* are now being used successfully by virus writers.

This kind of virus may require a complete redesign of some anti-virus engines, which work only at application level and therefore cannot remove the viral hook in memory. This may also mean that, in time, we will see a complete stealth virus for *2000/XP*, or even a mass-mailer that uses some of Che's features.

Pandora's box has been opened: we, as anti-virus researchers, need to treat this kind of virus with care and add proper detection routines for them. Otherwise we risk being taken by surprise in the future.

TECHNICAL FEATURE

THE SCALABLE STEALTH TROJAN: AN UPCOMING DANGER

Eyal Dotan
R&D, TEGAM International, France

Combining firewalls, desktop firewalls and anti-virus products provides protection against most popular ready-to-use Trojan horse kits such as the infamous BackOrifice and SubSeven. However, the issue of hand-crafted Trojan horses is quite different.

While ready-to-use Trojan horses are used by people who don't have the skills or the time required to create their own attack tools, hand-crafted Trojans are written to perform an attack on a specific target, hence they do not often become known – and if they do, it's often after the Trojan has stolen and/or destroyed data.

This article describes a combination of techniques, both existing and new, that can render Trojan horses unseen by most networking and *Windows* security tools. These techniques do not rely on any vulnerabilities, but rather on concept flaws in existing protection systems. Although many of these techniques have been used already by one Trojan or another, they have not yet been combined to create a 'super-stealth' Trojan horse.

This article describes a new category of Trojan horses I refer to as 'Scalable Stealth Trojans' (SST), and which potentially could become a standard for upcoming Trojan horses. Since prevention is better than cure, I have written this article to initiate a constructive discussion and exchange of ideas on this subject.

PROCESS IDENTITY FALSIFICATION

The first thing that is notable about SST Trojans is that their process is invisible, and their network activity is not seen by desktop firewalls. How does one achieve such a 'hack'? In fact, this does not require any hack at all. It can be achieved using elegant and well-documented mechanisms. SST uses a technique which I call 'Process Identity Falsification' (PIDF). It consists of riding on legitimate processes in order to perform malicious operations.

First, let's take a look at how desktop firewalls prevent unauthorized processes from accessing the network and the Internet.

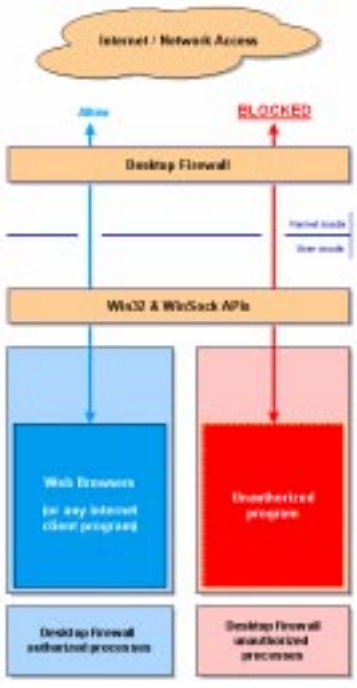


Figure 1: Desktop firewall outbound traffic control.

When desktop firewalls intercept Internet or network traffic, they check which process originated the request. If the process belongs to an 'authorized processes' list, it is allowed. If not, either the user is alerted or the process is denied automatically. There are a number of ways in which SST Trojans can bypass this protection.

SST Trojans don't necessarily need to act from their own process. They can install in-memory hooks, as shown in Figures 2 and 3.

Once run, the SST Trojan installs system-wide hooks (in either user mode or kernel mode), which allow it to intercept API requests coming from all programs in the system. One method of hooking service functions in user mode is called 'API hooking' – it works by modifying the in-memory import addresses of main system DLLs such as KERNEL32.DLL. This technique requires no administrator privileges to install into user-owned processes (i.e. web browsers, email programs, *Explorer* etc.). Once the hook is installed, the Trojan process can terminate, therefore will not be visible in the task list, while its hooks remain in memory.

Each time the hook code intercepts an event (e.g. a web browser requesting connection to a website, *Explorer* requesting to open a file, or simply by sniffing the keyboard keys), SST's hook code is executed in the context of the calling program. At that point it can open files, connect to Internet addresses and send and receive data for its own needs – all while the desktop firewall treats the traffic as legitimate, coming from your web browser.

Another way of performing PIDF is through a layer called WinSock's Service Provider Interface (SPI). SPI, also called LSP (Layered Service Provider), is an interface for hooking all socket operations within the system. In other words, whenever any program accesses the Internet, the SPI hook (the Trojan's DLL in this case) is called as if it were loaded by that program. Any I/O request that is performed from

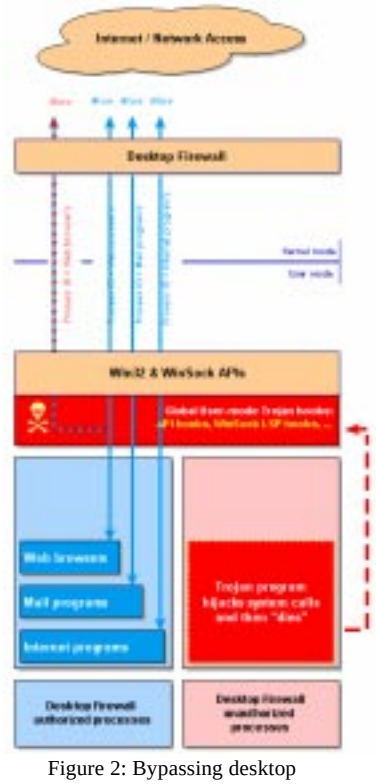


Figure 2: Bypassing desktop firewalls through user-mode hooks.

within the SPI hook will be seen by the system (and by the personal firewall) as having come from the legitimate program that initiated an Internet operation.

So, in addition to falsifying process IDs, SPI allows the Trojan to be launched at the machine's startup, with no easily detectable traces. Neither does SPI execute any process – SPI is merely a DLL that is loaded by any and all trusted Internet programs on the machine. Hence, it is not visible in the task list either. However, administrator privileges are required for its initial installation.

Another way of implementing PIDF and bypassing desktop firewalls is through thread injection (see Figure 4). This method consists of injecting a thread into an authorized process. On *Windows NT/2000/XP* there exists a documented API named CreateRemoteThread (derived from the Native API ZwCreateThread) and another named WriteProcessMemory. These APIs do not require administrator privileges, as long as they target processes that are owned (executed) by the same user.

From the point of view of the desktop firewall, this represents a real problem. If a Trojan injects code into processes such as a web browser or *Explorer*, then every operation this Trojan performs through the injected/modified thread code will be seen by the firewall as having come from these legitimate processes.

Just as with system hooks, SST Trojans can disappear from the task list, while the injected code continues to run in the memory space of legitimate processes. This technique has been demonstrated by the BackStealth program.

There are other techniques for making legitimate processes execute foreign code. For instance, Trojans can implement a plugin (some web navigators and mail programs support plugin DLLs). One could even implement the Trojan's code in an ActiveX component, and register it on the local machine. At a later stage, the malicious code will be executed from the web navigator's process – and will appear, to the desktop firewall, to be legitimate.

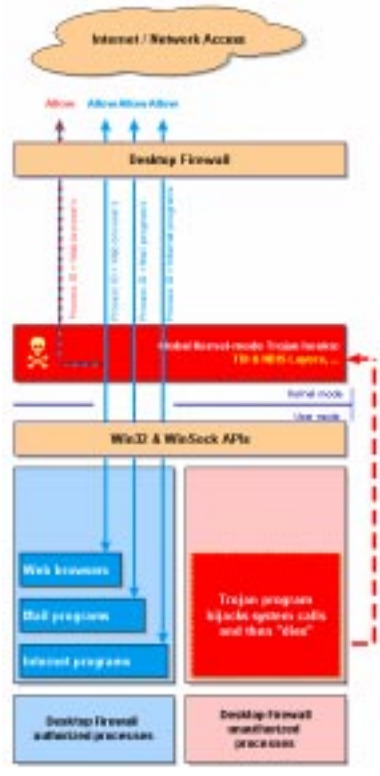


Figure 3: Bypassing desktop firewalls through kernel-mode hooks.

The same techniques work against behaviour blockers that monitor file or registry I/O – if SST needs to modify files or registry keys that are protected by behaviour monitors, PIDF can be used to perform the modifications to appear as if they originated from clean, authorized processes.

PASSIVE COMMUNICATION PROTOCOL AND HQ SERVERS

Some regular Trojans act as servers – typically, they open a TCP/IP port into which the hacker calls, commands the Trojan, and eventually receives stolen data in return. This is why most Trojans don't work properly behind firewalls; firewalls allow only servers to which the administrator has allowed access. SST Trojans do not act as servers. They do not wait for the attacker to call in. Instead, it is the Trojan that contacts its 'home', hence being a client, not a server. Since mid-2002, the Optix Lite Trojan has been using this technique.

SST Trojans have a dynamic list of IP and DNS addresses, or simply anonymous web URLs, through which they can contact the 'home' server. This list of servers changes from time to time, to avoid being discovered and/or being shut down. Each time the server changes, SST chooses a new server address at random. 'HQ servers' function using standard protocols such as HTTP. That way, they will neither be blocked nor trigger suspicion by the firewalls and logs on the systems where the SST Trojan is installed. Firewalls are configured to allow outbound traffic on the most commonly used services and ports. Outbound HTTP

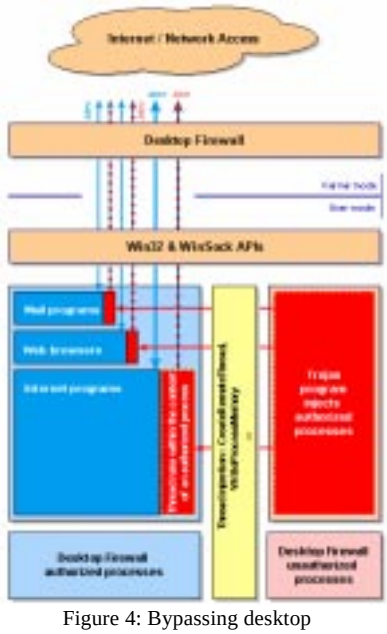


Figure 4: Bypassing desktop firewalls through thread injection or process memory modification.

traffic is usually allowed. SST Trojans synchronize with their HQ servers for two reasons: to receive and interpret instructions and to send information and stolen data.

RECEIVING INSTRUCTIONS

Assuming the HTTP protocol is chosen for communicating with the HQ servers, SST can receive instructions simply by visiting a website which, in reality, is one of the HQ servers. When 'visited', the HQ server issues commands to the Trojan, hidden within standard HTML pages. For instance, an HTML page containing the tag 'NewServers' means the Trojan should download a new list of HQ servers. Instructions are inserted into a normal-looking HTML page:

```
<html>
<head>
<title>Race cars</title>
      <NewServers>
          208.193.101.0/mypage.html
          202.113.101.31/index.html
          geocities.com/myownpage/mypage.html
</NewServers>
</head>
<p>My favorite racing cars are Porsche cars.</p>
</html>
```

Hence, the HQ server can send instructions to the SST Trojan indirectly, without opening ports on the infected machine and without using unusual transfer protocols. Other commands may be used for listing files, stealing data, gathering information about the attacked machine etc.

REMOTE MALWARE MUTATION

Many viruses use polymorphic engines to change the appearance of their code and render detection more difficult. Yet anti-virus developers can analyse the virus samples in their possession and write algorithms to detect the virus in its different forms. This is because samples of the polymorphic virus are sufficient for writing emulation routines or generating other infected samples and hence predicting the appearance of all infections resulting from that polymorphic engine. But what if there was no mutation engine other than a human brain?

Along with the 'mutate' command, the HQ server sends a new binary version of the Trojan horse (or a URL of a web/ftp site from which it can be downloaded). SST then replaces itself with the new version provided by the HQ server.

Just as auto-update features are included in many applications, Trojan horses can implement their own updates! This allows those behind the Trojan to fix bugs and adapt to new needs. But more importantly, it avoids leaving a constant signature for the SST Trojan's binary, in case a Trojan's sample is discovered and sent for analysis at a given time and place (which is less likely to happen if each SST sample is used on a small number of targets).

Since these modifications are made manually, and not by a polymorphic engine, an anti-virus signature is only useful until the next SST binary update. If one SST sample is discovered by anti-virus developers, it becomes an update-against-update fight.

Since most Trojans are written in a high-level language, only a small amount of code modification is required to change significant portions of the binary program and keep SST Trojans in an ever-changing shape. Hence, SST can render signature detection a more challenging task for anti-virus developers (although this depends on how many sites are targeted by the attack, and how much manpower is put to work).

SENDING INFORMATION AND DATA

The second goal of SST's synchronization is to send data to the HQ server. Again, this can be done using standard protocols such as FTP or SMTP (mail).

By using a standard outbound connection protocol, the Trojan avoids being blocked or detected by firewalls and proxies. Most Internet protocols include commands that let the client machine send chunks of data to the server (i.e. 'SEND', 'DATA', 'POST').

HIDING FURTHER

To minimize suspicion, the SST Trojan should use TCP/IP protocols that are coherent with the processes from which it performs the connections. If it reads instructions via HTTP, it should only do so via web-browsing processes (such as

IEXPLORE.EXE, NETSCAPE.EXE). If it uses SMTP to send data, it should do so via mail processes.

Some anti-virus products produce a notification window when emails are sent in order to inform users that their mails are being scanned. This is a useful feature, since it alerts the user to the fact that emails are being sent – even though the anti-virus product does not detect the Trojan specifically (because it isn't known).

In tests, however, we found a number of ways to bypass this obstacle – for example, by triggering SST's send activity when the user is away (by detecting activation of the screen saver and/or idle time), or when the session is locked.

Another method of avoiding suspicion is for the Trojan to send mails only when the email application itself sends an email, thus concealing the Trojan's malicious activity. This method of hooking the system and network activity may be seen increasingly in malicious software in general. It allows malware to behave in a more intelligent way and brings even more stealth locally and on the network level. When intercepted by SST, a user's actions, such as visiting a website or sending mail, trigger synchronization with the HQ server (with a maximum number of synchronizations per hour).

As a result, all of SST's TCP/IP activity due to HQ synchronization is lost amongst a mass of legitimate TCP/IP traffic. SST will generate TCP/IP activity only when the user performs an operation that is supposed to trigger such activity. Thus, neither the user nor the cautious administrator are likely to perceive anything suspicious.

Furthermore, hooking network activity allows malware to learn more about the attacked network. By observing packets going in and out, it can learn which servers and proxies are used. It can even steal passwords this way.

REAL-SCALE TESTING AGAINST PROTECTION SOFTWARE

In order to verify these findings and observe how protection programs react, my R&D team and I built sample SST Trojans using Visual C++ and tested them on *Windows 2000 Pro*.

We tested our samples against five of the most popular desktop firewalls, using their highest security settings, and using HTTP, FTP and SMTP to make the Trojan receive instructions and send data.

None of the programs we tested alerted on any of the ingoing and outgoing traffic. Since we were using the highest security settings, one of the products suggested putting the SST sample in a 'restricted group', watching it specifically for unauthorized actions (because it encountered this EXE file for the first time). Unfortunately, this did not help and our SST samples worked as expected.

Having carried out these tests behind a firewall, with the HQ on the other side of the firewall, we also verified that firewalls allow this kind of traffic go in and out (which is normal, since it is not the job of a firewall to block malware). However, as soon as we added a password-authenticated HTTP proxy, the Trojan was unable to connect to HTTP servers outside. Password-authenticated proxies seem to be an obstacle to this kind of Trojan. However, using a socket sniffer (not necessarily through WinSock hooks), we verified that it is possible for the Trojan to steal the user's proxy password when the web browser sends it to the server, in Base64. Hence, even this can bypassed.

Anti-virus products did not alert our SST samples since they were not known. We tested the SST auto-update function. The binary replaced itself, overwriting the old sample with the new one, hence simulating remote signature mutation.

HEURISTICS AND BEHAVIOUR DETECTION

You might wonder why this kind of behaviour wasn't flagged by heuristic scanners or dynamic code checkers. Well, detecting SST actions using a generic method is not that simple.

Let's take a closer look at PIDF: first, PIDF can be performed in a multitude of ways. WinSock hooks, thread injection and kernel-mode hooks are only a few examples.

Secondly, even if you focus on a behaviour that may look suspicious, such as injection, you should be aware that such *Windows* features were created in order to allow applications to modify the behaviour of other processes and allow software running on the operating system to be more flexible.

In fact, many legitimate programs inject processes in the system; for changing the graphical look and feel of your applications, implementing real-time dictionaries, monitoring performance and so on. Thus, there is a very fine line between legal and illegal and it would be quite a challenge to alert these kinds of operations without generating false alarms.

And let's not forget that while heuristics are good at detecting some types of virus, they are less efficient when it comes to detecting unknown Trojan horses – even primitive ones. Generic detection of Trojan horses and, in particular, hand-crafted Trojan horses, is more in the realm of desktop firewalls. The fact, as we have demonstrated, that desktop firewalls can be bypassed is bad news indeed.

CONFERENCE REPORT

WONDERFUL WONDERFUL: EICAR 2003 IN COPENHAGEN

Helen Martin

The 12th Annual EICAR Conference in Copenhagen saw the introduction of new conference streams dedicated to critical infrastructure protection and to IT law and forensics. Alongside the newcomers, the anti-virus stream comprised presentations from some of the industry's finest speakers.

Jakub Kaminski spoke about 'the grey zone' in which the AV industry finds itself. AV vendors are increasingly finding themselves with a 'should we?/shouldn't we?' dilemma regarding the detection of programs that are not strictly viruses or Trojans. Jakub highlighted the fact that pressure from users, marketing departments, and even testers and reviewers, means that AV researchers no longer have control over what is and what is not detected by anti-virus software.

Sarah Gordon called for the implementation of industry standards, beginning with synergistic naming schemes. The suggestion of a numerical naming scheme for viruses prompted discussion amongst delegates – many agreed such a scheme would be a helpful solution, while others felt that their users would not respond as well to warnings about a numerically named virus as they would to viruses with more memorable names.

Andrew Lee's simple response to the myth that there are no *Linux* viruses was, 'Yes, there are.' However, Andrew went on to explore the situation in more detail, providing ample evidence of the threat to *Linux* operating systems.

Ex-*VB* editor Richard Ford provided AV testers with food for thought. He began by pointing out the discrepancy between what today's AV products detect (a range of programs *other* than viruses) and what AV testing bodies currently test them against (viruses only). Critical areas of product functionality that are failing to be measured include speed of response to new threats, their ability to handle mass updates, and their sundry virus information. Richard asserted that, if they are to have any meaning, tests must concentrate on the way in which users use the software.

Larry Bridwell presented an update on the WildList since its legal rights were signed over to *ICSA Labs* in December 2002. Another update came from Eddy Willems, on the changes to the EICAR test file (see following pages). Other speakers included Jeannette Jarvis, who provided practical advice for protecting an organisation's infrastructure and David Perry who, using WildList data, attempted to answer questions such as 'how long is the lifecycle of a virus?'.

The 13th Annual EICAR conference is scheduled for 8–11 May 2004 in Bratislava, Slovakia. More information is available at http://www.eicar.org/.

OVERVIEW

THE WINDS OF CHANGE: UP-DATES TO THE EICAR TEST FILE

Eddy Willems
Data Alert International, Belgium
EICAR Director Information and Press

In 1991 the inaugural meeting of the European Institute for Computer Anti-Virus Research (now better known as EICAR) took place in Brussels. A few years later, as the result of cooperative effort between a number of anti-virus researchers, the EICAR standard anti-virus test file was created in order to provide an industry standard solution for a number of common questions – the test file remained unchanged until May 2003.

WHAT IS THE EICAR TEST FILE?

The purpose of the EICAR test file is to provide an industry standard solution for the following questions:

- Is my anti-virus program installed correctly – that is, does it intercept and/or detect viruses as it is supposed to?
- What happens when my anti-virus program detects a virus?
- Which messages are displayed?
- What about 'custom warnings', batch files and system admin notifications over the network?

The idea is that anti-virus programs detect the test file exactly as they would detect a virus, and effectively treat it as a virus.

Of course, a custom file for each anti-virus program would serve the same purpose, but the standard test file is intended to simplify the testing process – in particular in cases where multiple products are being tested and evaluated. The anti-virus industry adapted their products to the EICAR test file very well.

The following is a short abstract from the original EICAR test file definition:

> The file is a legitimate DOS program, and produces sensible results when run (it prints the message 'EICAR-STANDARD-ANTIVIRUS-TEST-FILE'). It is also short and simple – in fact, it consists entirely of printable ASCII characters, so that it can easily be created with a

regular text editor. Any anti-virus product that supports the test file should detect it in any file providing that the file starts with the following 68 characters:

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

To keep things simple, the file uses only upper case letters, digits and punctuation marks, and does not include spaces. The only thing to watch out for when typing in the test file is that the third character is the capital letter 'O', not the digit zero.

The string should be saved to a file with a .COM extension (EICAR.COM being the most obvious choice). So, when it is run, it will display the string 'EICAR-STANDARD-ANTIVIRUS-TEST-FILE!'.

As a side note, the file is printable so that it can easily be printed in a manual, included in software documentation, dictated over the telephone, or sent by fax.

It is not recommended that the EICAR file be included as a 'standalone' file in the anti-virus package in 'binary form', as users might run the anti-virus program on the package before realising what the test file is for.

The complete definition of the EICAR test file can be found at http://www.eicar.org/anti_virus_test_file.htm.

BAT/BWG.A@MM

In May 2002 the MS DOS batch worm Bat/Bwg.a@MM appeared. This worm was generated using a virus construction kit called Bwg ('Batch worm generator'). To date, this virus has not been seen in the wild.

The virus arrives as an email attachment, b.bat. Using *Outlook*, the virus will send an email to all recipients in the address book. When the attachment is double-clicked, the virus drops several copies of itself:

C:\a.bat C:\pro\a.jpg.bat

C:\b.bat %Windir%\b.arv.bat

Then it drops a VBS script, c:\dkhcz.vbs, which contains the code needed to mass-mail the virus.

The virus checks whether *mIRC* or *pIRCch* is installed on the machine. The worm will edit *mIRC*'s script.ini to send the file C:\pro\a.jpg.bat and drops b.arv.bat into the Windows directory. If *pIRCch* is installed the virus modifies events.ini to send b.arv.bat.

The virus can infect %windir%\startm~1\progra~1\autost~1*.bat and drop %windir%\Start Menu\Programs\StartUp\bjits.bat. In addition, it can copy itself to %windir%\Desktop*.ifk and rename %windir%\Desktop*.ifk to *.bat.

THE PROBLEM

The most significant feature of Bat/Bwg.a@MM is the fact that it is an attack on the EICAR test file. The virus starts with the EICAR string, which means that when it is run, a 'File not found' error is generated, but the execution of the virus continues.

A large number of anti-virus products misdetected this virus as the EICAR test file when it first appeared. This resulted in a lot of debate on various anti-virus discussion forums. Some members of the anti-virus community advocated changing the test file completely. After a while, each anti-virus vendor worked out their own way for their products to detect the EICAR test file properly.

THE CHANGE: TAKE ONE

The members of EICAR observed the problems that had arisen as a result of Bat/Bwg.a@MM and wanted to help the anti-virus vendors by changing the definition of the test file *slightly*, so that EICAR would have a fully correct and safe definition in use.

After consulting a number of anti-virus experts EICAR proposed the following change to the file:

The file is a legitimate DOS program, and produces sensible results when run (it prints the message 'EICAR-STANDARD-ANTIVIRUS-TEST-FILE'). It is also short and simple – in fact, it consists entirely of printable ASCII characters, so that it can easily be created with a regular text editor. Any anti-virus product that supports the test file should detect it in any file providing that the file starts with the following 68 characters, **and is exactly 68 bytes long:**

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

To keep things simple …

However, this proposal provoked some strong, negative response from members of the anti-virus industry – most anti-virus experts felt that the definition was neither explicit nor exact enough. Again, a lot of discussion took place on various anti-virus forums and even at WildList and CARO meetings.

THE CHANGE: TAKE TWO

EICAR decided to change the file again, in such a way that we would have mutual agreement between most anti-virus vendors. In order to achieve agreement, several anti-virus experts were asked for their input.

Responses were gathered from more than 40 members of the anti-virus industry. Afterwards I tried to combine all the

ideas into the latest definition change which was published 1 May 2003 on the EICAR website:

> The file is a legitimate DOS program, and produces sensible results when run (it prints the message 'EICAR-STANDARD-ANTIVIRUS-TEST-FILE'). It is also short and simple – in fact, it consists entirely of printable ASCII characters, so that it can easily be created with a regular text editor. Any anti-virus product that supports the test file should detect it in any file providing that the file starts with the following 68 characters:
>
> X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
>
> **The first 68 characters is the known string. It may be optionally appended by any combination of whitespace characters with the total file length not exceeding 128 characters. The only whitespace characters allowed are the space character, tab, LF, CR, CTRL-Z.**
>
> To keep things simple …

The new definition was sent to the various anti-virus forums at the beginning of this year in order to give every anti-virus vendor sufficient time to prepare for the necessary changes within their documents or programs.

To date, we have received no new reactions to or additional comments about the latest definition from members of the anti-virus industry.

In fact, one advantage of the 'new' test file is that it is not a complete change to the file. It is only the test file *definition* that has been narrowed in order to make it impossible to use the EICAR test file in a malicious way. This means that the majority of detection mechanisms within anti-virus programs do not need to be changed.

THE LAST WORD

I am certain that this will not be the last word concerning the EICAR test file.

During this year's EICAR conference (see this issue p.13) I presented an FAQ list concerning the file. Why not create different test files to test the other functionalities of the programs? Why not create a file to test for VBS viruses, macro viruses or blended threats? These questions were raised, but reach far beyond the original purpose of the file and we don't want to touch it in that way. The complete FAQ list will be available on the EICAR website shortly (see http://www.eicar.org/). I am happy to hear other suggestions about the test file, providing they are reasonable and that we are able to meet the needs of all anti-virus vendors. Any comments or suggestions should be sent to press@eicar.org.

COMPARATIVE REVIEW

WINDOWS XP PROFESSIONAL

Matt Ham

This month we revisit *Windows XP*. The last *XP* review (see *VB*, June 2002, p.16) was the first time the current testing machines were employed. Since both operating system and hardware were identical to a previous, fairly uneventful, comparative, this review seemed likely to go ahead without major hitches. Sure enough, the number of problems encountered with the products was at an all-time low. It is almost unheard of for no product to have caused the machines to freeze or crash. The greatest hurdle in this review was the sheer number of products on offer: 25 in all.

AhnLab V3 VirusBlock SP2

ItW Overall	99.96%	**Macro**	97.76%
ItW Overall (o/a)	99.96%	**Standard**	86.29%
ItW File	99.96%	**Polymorphic**	44.63%

As far as detection was concerned, *V3* was very mixed in its performance. *V3*'s detection of polymorphics was relatively poor, though detection of samples in the standard and macro test sets was good, if not astounding. Detection of ItW viruses is clearly the developer's primary concern, with detection here being all but perfect. However, the default engine settings did not allow detection of the extensionless copy of O97M/Tristate.C in this test set. This was sufficient to disbar *V3* from a VB 100% award. On the clean test set the results were much better. No false positives were generated in the clean sets and the scan rate on the non-archived files was at the fast end of the scale. Scanning of archives is not enabled by default and was slower.

Alwil avast! 4.0 Professional 4.0.208

ItW Overall	100.00%	**Macro**	99.56%
ItW Overall (o/a)	100.00%	**Standard**	99.57%
ItW File	100.00%	**Polymorphic**	91.21%

Alwil's developers have been hard at work recently, producing new features and entire new products. This has not prevented them from applying their time to the *XP* platform, however. *avast!* has a new appearance, the giant looming beetles of old having been replaced by a more conventional look. However, the new look did not seem to affect detection. Although there were slightly more missed detections on demand than on access, detection rates were perfect for viruses in the ItW test set. This, combined with the fact that no false positives were generated on the clean sets, earns *Alwil* a VB 100% award.

One problem was encountered during the testing of *avast!* Despite being set to overwrite and delete infected files, it seems that *avast!* is configured to back up all files in the Virus Chest. This should not prove a problem on a real-world machine – although, drive capacity seemed not to be checked, which led rapidly to the usage of all space on the partition where *avast!* was installed. This slowed down processing of files considerably, but was easily remedied, by deleting the archived infected files manually.

CAT Quick Heal X Gen 6.09

ItW Overall	100.00%	**Macro**	97.54%
ItW Overall (o/a)	100.00%	**Standard**	80.67%
ItW File	100.00%	**Polymorphic**	91.08%

Quick Heal has a tendency towards better detection of more recent viruses or those which are currently in the wild. This selectivity is commonly associated with a fast throughput rate for clean files, as was indeed the case for *Quick Heal*. With such selectivity the chance of false positives is reduced – *Quick Heal* generated none. With complete detection of viruses in the ItW test set, a VB 100% is netted by *CAT*.

Returning to old woes, the report files produced by *Quick Heal* were brimming with annoyances. In common with several other companies the report was in 8+3 format rather than using long file names. Rather more annoyingly, the logs also had extensions which changed case randomly, despite the names of all the test samples being upper case. Quite what is the reasoning behind such changes is anyone's guess; they certainly do not seem helpful under any circumstance that I can imagine.

Command AntiVirus for Windows 4.80.3

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.78%
ItW File	100.00%	**Polymorphic**	95.21%

Command AntiVirus proved its usual friendly self as far as testing was concerned, although the logging proved as intractable as ever. Logs were available only in rtf format, which is impenetrable to the scripts that are used for processing plain-text files. In such cases logs can often be obtained by choosing to print the log from within the program and diverting the printer output to a text file. However, this method resulted in a very truncated report and deletion of infected files was used to obtain results. When the results were processed there were few surprises. The misses were dominated by a selection of polymorphics, with W32/Heidi.A being missed only on access in its archive embedded form. None of the misses were within the ItW test set or the macro test set. When scanning clean files, *Command AntiVirus* proved to be among the faster products, especially on OLE files. With no false positives, the third VB 100% award of this review goes to *Command*.

Computer Associates eTrust Antivirus 7.0.139

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	100.00%

Although an old-timer in the *VB* Comparatives, this was the first test for version 7 of *eTrust* on *Windows XP*. The new version had one major advantage as far as installation was concerned, in that only one update file was needed rather than the accumulation of patches required in the past. Detection rates for *eTrust* were also good. No files were missed in the ItW test set and, combined with no false-positives in the clean set, another VB 100% award is earned by *CA*. *eTrust*'s scanning rates were at the more speedy end of the scale.

Computer Associates Vet Anti-Virus Protection 10:58.0.3

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.90%
ItW File	100.00%	**Polymorphic**	98.50%

Vet has a history of ease of installation and testing which was displayed again in this test. *Vet* is unusual in that it still supplies updates on floppy disk, in addition to the main CD media – most other products rely on the user to obtain electronic updates on first installation. *Vet's* performance on the clean test set was good, with fast overall scanning speed and no false positives. There were no misses in the ItW test set, meaning that *Vet Anti-Virus* gains another VB 100% award.

DialogueScience Dr.Web for Windows 95-XP 4.29c

ItW Overall	99.52%	**Macro**	100.00%
ItW Overall (o/a)	99.52%	**Standard**	100.00%
ItW File	99.51%	**Polymorphic**	100.00%

DialogueScience suffered a rare miss of a VB 100% award in last month's *Linux* tests, in which W95/Bodgy.A proved

On-access tests	ItW File		ItW Boot		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
AhnLab V3 VirusBlock	1	99.96%	0	100.00%	99.96%	95	97.76%	8830	44.63%	304	86.29%
AlwilI avast!	0	100.00%	0	100.00%	100.00%	18	99.56%	157	91.18%	13	99.57%
CA eTrust Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.70%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	437	98.50%	4	99.78%
CAT Quick Heal	0	100.00%	0	100.00%	100.00%	104	97.51%	0	100.00%	718	62.15%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	91	95.21%	11	99.64%
DialogueScience Dr.Web	5	99.51%	0	100.00%	99.52%	0	100.00%	0	100.00%	3	99.70%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	100.00%	0	100.00%	4	99.82%	4	99.73%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	3	99.86%
GDATA AntiVirusKit	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	0	100.00%
GeCAD RAV	0	100.00%	0	100.00%	100.00%	0	100.00%	35	97.61%	3	99.70%
Ggreat ZMW32	-	-	-	-	-	-	-	-	-	-	-
Grisoft AVG	0	100.00%	0	100.00%	100.00%	23	99.44%	425	83.72%	43	97.44%
HAURI ViRobot	0	100.00%	0	100.00%	100.00%	43	98.84%	10795	33.63%	530	73.69%
Kaspersky KAV	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	0	100.00%
MicroWorld eScan	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	0	100.00%
NAI VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	4	99.70%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	2	99.95%	178	91.27%	11	99.64%
NTW Virus Chaser	5	99.51%	0	100.00%	99.52%	0	100.00%	0	100.00%	5	99.61%
SOFTWIN BitDefender	0	100.00%	0	100.00%	100.00%	17	99.59%	45	95.11%	72	97.57%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	11	99.73%	60	95.79%	15	99.31%
Symantec AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend PC-cillin	0	100.00%	0	100.00%	100.00%	0	100.00%	215	95.77%	7	99.84%
VirusBuster VirusBuster	0	100.00%	0	100.00%	100.00%	0	100.00%	159	89.13%	11	99.55%

to be a bugbear for the product. Discussions with the product's developers revealed that all samples received by *DialogueScience* had been non-replicable and therefore the virus had been classified as an intended threat, rather than an actual threat. With a very short span of time between the *Linux* and *XP* tests, this hitch was redressed the day after

the submission deadline for this review – not quite in time to be reflected in these results. Thus *Dr.Web* missed the offending samples of W95/Bodgy.A in this test and miss out on a VB 100% award as a result. Newer versions of the software do not have this problem.

In other respects the product performed admirably, with fewer suspicious files than usual occurring in the clean test set. The on-access scan did show some slight quirks, however – it seemed that files containing embedded information bypassed the 'automatic action setting' and required user intervention. Since there are few of these files in the test set, this was only a momentary distraction.

Eset NOD32 Anti-virus 1.405

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	100.00%

It seems that the developers of several products have opted for a change in interface. *NOD32* is no exception. The version reviewed was noted as being the last in its current form – the alien-esque imagery being consigned to history. With this impending change it seemed likely that the current version would be subject to a freeze in features and development, and indeed its appearance was identical to that which it has had for the last two years or so. *NOD32*'s performance was also all but identical to its past performances: fast scanning, no false positives and full ItW detection combining to earn *NOD32* yet another VB 100% award.

FRISK F-Prot Antivirus 3.13a

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.73%
ItW File	100.00%	**Polymorphic**	99.82%

Before reporting on *FRISK*'s performance, a comment about last month's *Linux* comparative (see *VB*, May 2003, p.20). In the *Linux* comparative it was stated that the *F-Prot Antivirus* on-access scanner scans only HTTP GET requests. This statement was incorrect, having been the result of a misinterpretation of the documentation. The product is currently undergoing re-testing in consultation with the developers.

Back to the current tests and no problems were apparent for *F-Prot Antivirus*. The product ran quickly through the clean set scans without incident or false positives of any sort. With less than a dozen misses overall, none of which were in the ItW test set, *F-Prot Antivirus* qualifies for a VB 100% award.

F-Secure Anti-Virus 3.12.410

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.86%
ItW File	100.00%	**Polymorphic**	99.92%

Since it incorporates the *F-Prot* engine, the fact that *FRISK*'s product earned a VB 100% boded well for *F-Secure*'s performance – the inclusion of *Kaspersky*'s engine being an additional line of defence. Sure enough, detection rates were similar, though slightly improved by the additions inherent in *F-Secure*'s multi-engined product. There was, however, an oddity in the *F-Secure* scan settings. Despite being set to leave all infected files and simply log results, several disinfected files were left after scanning was completed. Test sets are refreshed from images after each scan has been performed, so the results cannot be affected by such behaviour, but this activity certainly rates as unexpected. As mentioned, detection rates were good, with only a handful of misses, none of which were in the ItW test set. Scanning speeds on clean files were respectable, and no false positives were seen. As a result, *F-Secure Anti-Virus* is the recipient of a VB 100% award.

GDATA AntiVirusKit Professional 12.0.4

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	99.92%

AntiVirusKit is, like *F-Secure Anti-Virus*, a multi-engined beast and it too features *Kaspersky* technology – this time alongside the *RAV* engine. This pairing performed well in detection, with one sample of W32/Etap being the only miss across all the test sets. No false positives were generated in the clean test sets, thus *AVK* achieves a VB 100%. The double layer of detection does not come without a price, however. *AVK*'s scanning speed was slower than the average by a considerable degree, most noticeably on the executable test sets.

GeCAD RAV for Windows 8.6.104

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.88%
ItW File	100.00%	**Polymorphic**	97.61%

Misses for *RAV* were again few in number, although W32/Etap was undetected in this case, rather than partially detected. None of the files that were missed were in the ItW test set,

On-demand tests	ItW File		ItW Boot		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
AhnLab V3 VirusBlock	1	99.96%	0	100.00%	99.96%	95	97.76%	8830	44.63%	304	86.29%
Alwil avast!	0	100.00%	0	100.00%	100.00%	18	99.56%	153	91.21%	13	99.57%
CA eTrust Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	437	98.50%	2	99.90%
CAT Quick Heal	0	100.00%	0	100.00%	100.00%	101	97.54%	1543	91.08%	367	80.67%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	91	95.21%	8	99.78%
DialogueScience Dr.Web	5	99.51%	0	100.00%	99.52%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	100.00%	0	100.00%	4	99.82%	4	99.73%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	3	99.86%
GDATA AntiVirusKit	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	0	100.00%
GeCAD RAV	0	100.00%	0	100.00%	100.00%	0	100.00%	35	97.61%	2	99.88%
Ggreat ZMW32	99	83.67%	9	0.00%	82.29%	1550	62.90%	14737	10.37%	525	74.31%
Grisoft AVG	0	100.00%	0	100.00%	100.00%	20	99.51%	257	85.97%	22	99.21%
HAURI ViRobot	0	100.00%	0	100.00%	100.00%	43	98.84%	10795	33.63%	530	73.69%
Kaspersky KAV	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	0	100.00%
MicroWorld eScan	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.92%	0	100.00%
NAI VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	4	99.70%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	2	99.95%	178	91.27%	9	99.76%
NTW Virus Chaser	5	99.51%	0	100.00%	99.52%	0	100.00%	0	100.00%	0	100.00%
SOFTWIN BitDefender	0	100.00%	0	100.00%	100.00%	17	99.59%	45	95.11%	62	97.93%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	11	99.73%	60	95.79%	15	99.31%
Symantec AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend PC-cillin	0	100.00%	0	100.00%	100.00%	0	100.00%	215	95.77%	7	99.84%
VirusBuster VirusBuster	0	100.00%	0	100.00%	100.00%	4	99.90%	160	89.13%	9	99.64%

however. Clean set scanning proved *RAV* to be slightly slower than the average, with a default setting of archives remaining unscanned. However, there were no false positives, and *RAV* adds to its collection of VB 100% awards. Despite an admirable performance as far as detection was concerned, there were some peculiarities with

the product's interface. Whenever launched the software reverted to its simple configuration, rather than the advanced interface that had been selected. This was probably related to a number of error messages that appeared on launching the program – problems appeared to be related to configuration rather than engine difficulties and certainly did not impair scanning performance.

Ggreat ZMW32 Virus Scan M7.5+

ItW Overall	82.29%	**Macro**	62.90%
ItW Overall (o/a)	n/a	**Standard**	74.31%
ItW File	83.67%	**Polymorphic**	10.37%

Ggreat's *ZMW32* remains the only program in this review with no on-access component in the traditional sense. It does contain amail and http filters which operate in real time, although these do not fall under the functionality tested in these comparatives. The rather spartan command set proved a slight hindrance to testing: files may only be disinfected, there being no option to delete. The program has seen much improvement since its earlier versions were reviewed – on previous occasions the product suffered from general instability and logging did not appear to work fully. These problems now seem fully solved.

Detection rates saw an improvement too – although there was a certain degree of unpredictability. The test sets were scanned several times with slightly different results being obtained on each occasion. In the end disinfection was used repeatedly and a log taken of those files still noted as infected. Disinfected files, counted as detections, were discovered by the use of CRCs. All in all, the product has improved, though it still has a long way to go until it can qualify for a VB 100%.

Grisoft AVG Anti-Virus System 6.0.478 275

ItW Overall	100.00%	**Macro**	99.51%
ItW Overall (o/a)	100.00%	**Standard**	99.21%
ItW File	100.00%	**Polymorphic**	85.97%

Grisoft's *AVG* is another of those products which continue to put in stalwart performances. Misses were, as usual, mostly in the polymorphic test sets, with a small number in the standard and macro test sets. However, no misses were noted in the ItW test set. In the clean set *AVG* found five suspicious files, but there were no outright declarations of infection, thus *AVG* achieves a VB 100% award.

In the *Windows 2000 Advanced Server* comparative (see *VB*, November 2002, p.16) *AVG* was noted to have missed an ItW sample of W32/Zoek.D (in addition to one other file which denied the product a VB 100% award). Investigation has since shown the Zoek file to be a dropped backdoor portion of the virus, rather than an infective object. As a result, the file has been removed from the test sets and *AVG* should not have been logged as missing W32/Zoek.D.

HAURI ViRobot Expert 4.0

ItW Overall	100.00%	**Macro**	98.84%
ItW Overall (o/a)	100.00%	**Standard**	73.69%
ItW File	100.00%	**Polymorphic**	33.63%

The last time *HAURI*'s *ViRobot* appeared in a comparative review, the product was among the speedier entrants in the clean test sets, and the state of affairs this time was much the same. It was noted on the last occasion that the product's speed could, in part, be attributed to *ViRobot*'s non-detection of a number of older viruses. The same lack of detection of older viruses was seen this time, with large numbers of the polymorphic viruses being missed en masse. Despite these misses, *ViRobot* performed well on newer viruses and missed none of the samples in the ItW test set. In the clean sets one suspicious file was noted, though it was not declared infected, and thus *HAURI* is awarded a VB 100%.

Kaspersky Anti-Virus 4.0.5.37

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	99.92%

Kaspersky Anti-Virus (*KAV*) was tested with high hopes for good detection rates. Sure enough, just the one missed sample of W32/Etap was noted (the same sample that was missed by *GDATA*'s product).

However, the impressive detection rate was rather spoiled by the presence of a false positive in the clean test sets. The problem file was declared to be a rebooting Trojan – in fact it is designed as a rebooting utility. The old adage (in computer terms at least) that renaming format would be enough to make it a Trojan, is brought to mind. Although this was an understandable misdiagnosis on the part of *Kaspersky Anti-Virus*, it was sufficient to deny a VB 100% award on this occasion.

MicroWorld Services eScan 2003 10.1.02 (2.6.198.6)

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	99.92%

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files	
	Time (s)	Throughput (KB/s)	FPs [susp]	Time(s)	Throughput (KB/s)	FPs [susp]	Time (s)	Throughput (KB/s)	Time(s)	Throughput (KB/s)
AhnLab V3 VirusBlock	35	15626.6		11	7212.2		134	1189.7	28	2664.6
Alwil avast!	112	4883.3		24	3305.6		64	2490.9	21	3552.7
CA eTrust Antivirus	94	5818.4		4	19833.4		43	3707.4	8	9325.9
CA Vet Anti-Virus	77	7103.0		5	15866.8		50	3188.3	10	7460.7
CAT Quick Heal	56	9766.6		12	6611.1		44	3623.1	13	5739.0
Command AntiVirus	103	5310.0		4	19833.4		49	3253.4	5	14921.5
DialogueScience Dr.Web	226	2420.1	[12]	14	5666.7		84	1897.8	15	4973.8
Eset NOD32	27	20256.7		3	26444.6		27	5904.3	6	12434.6
FRISK F-Prot	102	5362.1		5	15866.8		57	2796.8	6	12434.6
F-Secure Anti-Virus	208	2629.5		8	9916.7		118	1351.0	18	4144.9
GDATA AntiVirusKit	457	1196.8		12	6611.1		209	762.8	26	2869.5
GeCAD RAV	276	1981.6		5	15866.8		130	1226.3	5	14921.5
Ggreat ZMW32	29	18859.7	4	18	4407.4		2070	77.0	400	186.5
Grisoft AVG	57	9595.3	[5]	7	11333.4		60	2656.9	12	6217.3
HAURI ViRobot	35	15626.6	[1]	21	3777.8		81	1968.1	27	2763.2
Kaspersky KAV	188	2909.2	1	12	6611.1		110	1449.2	30	2486.9
MicroWorld eScan	149	3670.7		18	4407.4		90	1771.3	30	2486.9
NAI VirusScan	105	5208.9		14	5666.7		72	2214.1	18	4144.9
Norman Virus Control	190	2878.6		8	9916.7		103	1547.7	12	6217.3
NTW Virus Chaser	139	3934.8	[12]	8	9916.7		58	2748.6	10	7460.7
SOFTWIN BitDefender	862	634.5	[1]	6	13222.3		416	383.2	17	4388.7
Sophos Anti-Virus	69	7926.6		9	8814.9		38	4195.2	11	6782.5
Symantec AntiVirus	138	3963.3		20	3966.7		59	2702.0	21	3552.7
Trend PC-cillin	77	7103.0		4	19833.4		43	3707.4	12	6217.3
VirusBuster VirusBuster	170	3217.2		7	11333.4		105	1518.3	14	5329.1

eScan is yet another product that is derived from third-party engines – this one being a derivative of the *GDATA* product, which, in turn, incorporates the *Kaspersky* and *RAV* engines. What was surprising about *eScan* lay in the matter of scanning speeds on the clean test sets. In most cases, the further from the ultimate source of the engine, the slower the product becomes. In this case, however, scanning speeds were faster than for any of the other products involved. On the less positive side, however, the detection of boot sector

viruses on access was (although complete eventually) rather a hit and miss affair. Detection rates for *eScan* were identical to those seen in the *GDATA* product, as was the lack of false positives in any clean set. In combination, this performance was sufficient to gain *eScan* a VB 100%.

NAI VirusScan Enterprise 7.00 4.2.40 4261

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.70%
ItW File	100.00%	**Polymorphic**	100.00%

NAI was notable by its absence in last month's *Linux* review and makes a welcome return this month. It should be pointed out that *NAI*'s lack of submission in last month's review was a result of a combination of errors on the part of both *VB* and *NAI*, rather than a deliberate absence from the testing line-up on the part of the developer. This month the review process for *VirusScan* started smoothly enough, although initial scanning tests were thwarted by the non-appearance of logs if the default log location and name were used. Changing these resolved the problem, and scanning progressed unhindered. No false positives were noted on the clean set tests, while scanning rates remained around the average. Misses of infected files were limited to the archived versions of W32/Heidi.A and the now defunct JS/Unicle. This performance was sufficient to earn *VirusScan* a VB 100% award.

New Technology Wave Inc. Virus Chaser 5.0

ItW Overall	99.52%	**Macro**	100.00%
ItW Overall (o/a)	99.52%	**Standard**	100.00%
ItW File	99.51%	**Polymorphic**	100.00%

Virus Chaser is another rebadged product – in this case *DialogueScience* is the engine developer. The product's detection rates and behaviour in the clean sets were all but identical to those exhibited by *Dr.Web*. Unfortunately this included the missed samples of W32/Bodgy.A and thus *Virus Chaser* does not obtain a VB 100% award this month.

Norman Virus Control 5.50 5.40.42

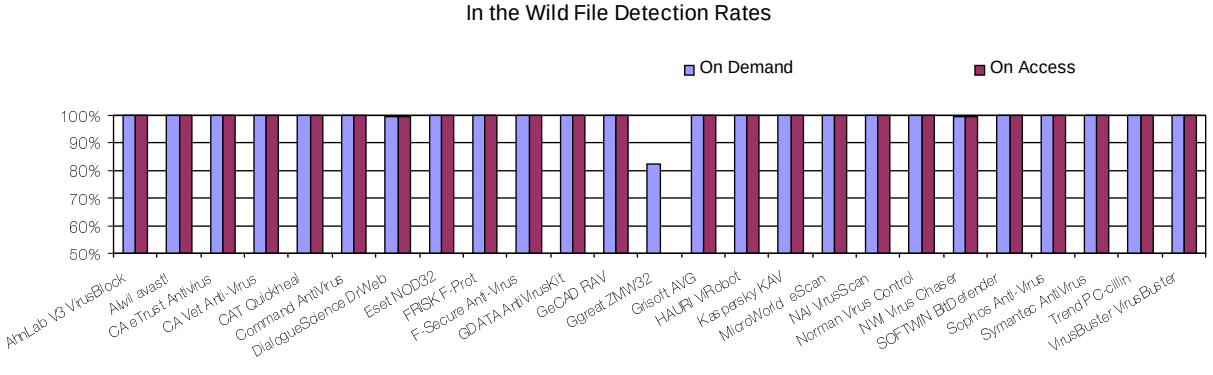
ItW Overall	100.00%	**Macro**	99.95%
ItW Overall (o/a)	100.00%	**Standard**	99.76%
ItW File	100.00%	**Polymorphic**	91.27%

Norman Virus Control has been notable over the last year for its changing log file status. Configurations have moved through no logs, logs of both missed and detected files, and have now stabilised at logs of infected files only. The log files proved easy enough to parse in this form and showed *NVC* to have strong detection rates against all but some modern polymorphics, none of which have yet entered into the ItW test set. The clean set files were scanned without any problems or false positives, thus *NVC* earns a VB 100% award. *NVC* suffered the same problem in last month's *Linux* comparative as *AVG* suffered in the previous review: *NVC* should not have been logged as having missed W32/Zoek.D. However, the lack of an on-access scanner means that *NVC* still did not qualify for a VB 100%.

SOFTWIN BitDefender Standard Edition 7 72112

ItW Overall	100.00%	**Macro**	99.59%
ItW Overall (o/a)	100.00%	**Standard**	97.93%
ItW File	100.00%	**Polymorphic**	95.11%

BitDefender has had a few ups and downs in its performance over the years. On this occasion the product showed perfect detection of ItW samples in addition to good detection rates in the other test sets. There was only one disappointment, this concerning the speed of scanning. Although by no means



the worst upon OLE files or zipped executables, the scanning rate of non-archived executables was sluggish. No false positives were obtained, however, and thus *BitDefender* earns a VB 100% award.

Sophos Anti-Virus 3.69

ItW Overall	100.00%	**Macro**	99.73%
ItW Overall (o/a)	100.00%	**Standard**	99.31%
ItW File	100.00%	**Polymorphic**	95.79%

Sophos Anti-Virus has, in the past, lagged somewhat behind the pack in fully-automated daily update technology and currently its developers are working on various projects to lessen this gap. It was therefore a happy surprise to be presented with a new option for updating the product, in the form of an executable file. However, it was merely a self-extracting zip file, rather than an updating tool as such – it was still necessary to position the update files by hand and to restart the *Sophos Anti-Virus* application. With this process complete, the application performed in its usual smooth fashion. Results were good, with perfect detection of ItW files and misses elsewhere comprising a selection of files which have been missed more or less constantly for several months. With no false positives, and a fairly speedy rate of scanning, *Sophos* takes home a VB 100% award.

Symantec AntiVirus Corporate Edition 8.00.9374 4.1.0.15

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	100.00%

Symantec AntiVirus did not disappoint on this occasion. With no false positives and full detection of all files in the ItW test set a VB 100% award is earned. There was one less than ideal feature of the product, however. On samples of W32/CTX and W32/SK variants the scanning speed was very slow indeed, with delays of several seconds between the scanning of some files. This is not a problem which is exhibited on clean files, however, so is more than likely a side-effect of the fact that exact virus identification is regarded as important by the developers.

Trend Micro PC-cillin 2003 10.02 1072

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.84%
ItW File	100.00%	**Polymorphic**	95.77%

Where detection was concerned, *PC-cillin*'s misses were confined exclusively to various polymorphic viruses, including some which are also located within the standard test set. On-access scanning revealed one rather bizarre piece of behaviour – after a short time the display became rather garbled in those areas where screen refreshes were not being forced. However this seemed to affect neither the performance of *PC-cillin* nor that of other applications on the machine. Performance in both detection and the clean set tests was ample for *Trend* to gain a VB 100% award.

VirusBuster VirusBuster for Windows Antivirus Solution 4.2 build 16

ItW Overall	100.00%	**Macro**	99.90%
ItW Overall (o/a)	100.00%	**Standard**	99.64%
ItW File	100.00%	**Polymorphic**	89.13%

VirusBuster displayed few faults or pieces of outstanding behaviour. The test procedures all ran smoothly, with no untoward false positives in the clean set, and misses of infected samples were mostly among the polymorphic samples. There was a smattering of misses in the macro test set, but none in the ItW set. *VirusBuster* deservedly gains a VB 100% award.

CONCLUSION

In comparison with the non-*Windows* test of the last comparative reivew, this month's results show a large number of VB 100% awards being achieved.

Of course, *Windows XP* is sufficiently similar to *NT* that lessons learned on products for that platform have helped in the smooth production of products for *XP*. What remains to be seen, however, is whether those lessons are specific to the architecture or whether they can be applied more generally within any *Microsoft*-designed environment. The answer to that question will come in due course, when 64-bit *Windows* operating systems move from being strapped-on afterthoughts to mainstream platforms in their own right. How soon that will be is anyone's guess, but the tests should make for interesting reading.

Technical details:

Test environment: Identical 1.6 GHz Intel Pentium machines with 512 MB RAM, 20 GB dual hard disks, DVD/CD-Rom and 3.5-inch floppy drive running *Windows XP Professional*.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/WinXP/2003/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

END NOTES & NEWS

RSA Conference 2003 Japan will be held 3–4 June 2003 at Tokyo International Forum. This e-security conference and exhibition in Japan is modelled on the US-based RSA Conference. For details see http://www.rsaconference.com/.

Infosecurity Canada Conference and Exhibition takes place 4–5 June 2003 in Toronto, Canada. For registration and exhibitor details see http://www.infosecuritycanada.ca/.

The 15th Annual Computer Security Incident Handling Conference takes place 22–27 June 2003 in Ottawa, Canada. For more information see http://www.first.org/conference/2003/.

NetSec 2003 Conference and Exhibition takes place at the Hyatt Regency, New Orleans 23–25 June 2003. For the conference programme, exhibitor list and registration information, see http://www.gocsi.com/.

The Third World Conference on Information Security Education takes place 26–28 June 2003 in Monterey, USA. For details see http://cisr.nps.navy.mil/wise3/.

The Black Hat Training and Briefings USA 2003 take place 28–31 July 2003 at the Caesar's Palace hotel, Las Vegas. For full details and registration see http://www.blackhat.com/. DEFCON 11 will take place 1–3 August 2003 in Las Vegas, following the Black Hat Training and Briefings. See http://www.defcon.org/.

COMDEX Canada 2003 will be held 16–18 September 2003 in Toronto, Canada. See http://www.comdex.com/.

The 13th *Virus Bulletin* International Conference and Exhibition (VB2003) takes place 25–26 September 2003 at the Fairmont Royal York hotel in Toronto, Canada. For exhibition details, call +44 1235 555139 or email vb2003@virusbtn.com. For more information including full programme details and online registration see http://www.virusbtn.com/conference/.

The 5th NTBugtraq Retreat takes place in the days immediately following the *Virus Bulletin* conference in Ontario, Canada. A welcome event on the evening of 26 September will be followed by the Retreat from 27–29 September 2003. Full details can be found at http://www.ntbugtraq.com/party.asp.

Black Hat Federal 2003 takes place 29 September to 2 October 2003 in Washington D.C. For more information and online registration see http://www.blackhat.com/.

InfowarCon 2003 takes place 30 September to 1 October 2003 in Washington D.C. Military leaders, political forces, academics, and industry members will discuss the concepts of the latest on-going initiatives in the Homeland Security and Critical Infrastructure Protection communities. For details see http://www.infowarcon.com/.

The Workshop on Rapid Malcode (WORM) will be held 27 October 2003 in Washington D.C. The workshop aims to bring together ideas, understanding and experience relating to the worm problem from academia, industry and government. See http://pisa.ucsd.edu/worm03/.

COMPSEC 2003 will be held 30–31 October at the Queen Elizabeth II Conference Centre in Westminster, London, UK. This year's conference will include the Compsec 2003 Poster Session, featuring a review of the latest scientific advances in computer security research and development (deadline for poster contributions 30 June 2003). Early registrations close on 15 June 2003. For full details see http://www.compsec2003.com/.

The European RSA Conference will be held 3–6 November at the Amsterdam RAI International Exhibition and Congress Center, The Netherlands. Further details will be announced in due course at http://www.rsaconference.com/.

AVAR 2003 will be held on 6 and 7 November 2003. This year's AVAR (Association of anti Virus Asia Researchers) conference will be held in Sydney, Australia. More details will be announced in the near future at http://www.aavar.org/.

COMDEX Fall 2003 takes place 15–20 November 2003 in Las Vegas, USA. See http://www.comdex.com/.

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: +44 (0)1235 555139 Fax: +44 (0)1235 531889
Email: editorial@virusbtn.com www.virusbtn.com