FEBRUARY 2006

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

IN THIS ISSUE

WMF IN THE SPOTLIGHT

The Windows Meta File (WMF) format has received a lot of attention over recent weeks. Peter Ferrie finds out a bit more about it, and then explains why it has been in the spotlight.
page 5

AV UNDER THE MICROSCOPE

Dan Kaminsky takes a long, hard look at the reaction of the AV industry (or lack thereof) to the *Sony* rootkit incident and assesses the consequences.
page 8

WINDOWS NT ON TEST

Matt Ham fully expected a bumper harvest of VB 100% awards this month, simply due to the familiarity of the *Windows NT* platform to developers. Find out whether his expectations were met in his *NT* comparative review.
page 12

vbSpam supplement

This month: anti-spam news & events; John Graham-Cumming investigates whether Bayesian poisoning really exists.

ISSN 1749-7027

'A serious threat in 2006 will be multi-stage, targeted phishing attacks.'

Tomer Honen
Aladdin Knowledge Systems

WHAT THREATS MAY COME

It is said that money makes the world go round – well it certainly drives the malware community. Over the last year, we've seen a sharp increase in the number of backdoor attacks employed by various in-the-wild threats. Taking over computers is apparently quite a profitable endeavour, and there are many buyers for the scores of backdoor-infected PCs out there. It is safe to say that 2006 will be just as malware-filled as 2005 was, and probably even worse.

Mobile phone threats will make a few headlines this year as the number of smartphone users grows. We are likely to see new mobile phone threats, which may be able to spread to other platforms and infect them. Since the communication channels employed by mobile phones are often insecure, it will be easy to use these devices to implant remotely controlled Trojans in a corporate environment. In this way, hackers won't even need to make the initial contact with the infected PC; all they need to do is infect a mobile device and let the PCs come to them.

More disturbing than common spyware or worms are invisible targeted spyware and Trojans. Last year we witnessed several Trojan-related incidents that made headlines around the world – from corporate attacks in Israel and the UK, to major credit card information theft in the US. A Trojan operator needs access to a compromised system for just hours or even minutes to steal vital information. Since most attacks are unique, it is rare to see more than a handful of copies of each individual Trojan and traditional signature-based solutions are usually unable to block these threats. We are likely to witness more incidents of this nature in 2006. Or rather, we'll be lucky if we can spot them before they get to us, carry out their payload and disappear.

A serious threat in 2006 will be multi-stage, targeted phishing attacks. According to the Anti-Phishing Working Group (http://antiphishing.org/), thousands of phishing attacks are reported every month, but the attack methods are changing. Instead of luring victims to spoofed websites where they are fooled into entering confidential information such as their passwords, financial details etc., the latest trend is to use password-stealing malicious code in the phishing websites themselves. Even if the user does not enter the confidential information, they may be infected by malicious code that will extract it forcefully.

The number of malicious code phishing sites more than quadrupled in 2005, to over 1,000 reported sites. Money-driven attackers will exploit this obvious Achilles' heel by employing multi-stage targeted phishing attacks:

- Certain users will receive an email intended specifically for them. This will display content that is of interest to the recipient, trying to get them to click on a link to a site.
- The website will contain malicious code that drops and executes a backdoor Trojan on the victim's system with little or no user interaction. And voilà! Somewhere in the world a hacker obtains a new remote controlled system. This turns phishing attacks into a serious corporate threat.

Many content security solutions scan malicious content received by mail but neglect to analyse content downloaded from the web. By neglecting to inspect web traffic, users become exposed. Also, many users do not realize that email messages can execute content downloaded from the web as the message is viewed.

According to *Gartner*, 'Through 2010, each new technology transition point will result in 30 per cent more newly opened attack paths than old paths that are closed.' In layman's terms this means that the more advanced we become, the bigger the threat. It is a grim prediction, and there's little that can be done to make the world behind our firewall, anti-virus, anti-spyware and anti-spam products more hospitable. However, with the right protection, some of us should hopefully have a relatively quiet year.

NEWS

NEIGHBOURHOOD WATCH TO FIGHT 'BADWARE'

Google, *Lenovo* and *Sun Microsystems* have pledged their support for a new initiative aimed at fighting spyware, malware and deceptive adware. The Neighbourhood Watch-style campaign *Stopbadware.org* is being led by Harvard Law School's Berkman Center for Internet & Society in the US, and Oxford University's Oxford Internet Institute in the UK.

The effort aims to provide clear and objective information about downloadable applications so that consumers can make informed choices about what they download onto their systems. *Stopbadware.org* will gather stories and data from users, and use that data to inform its research efforts. The *Stopbadware.org* website invites less technical users to submit stories about their experiences with 'badware', as well as appealing to those who are more technically-aware to submit technical data reports. Those behind the campaign also plan to write standards and testing procedures to define 'badware', and spotlight the worst offenders with the help of the anti-malware community. Full details of the project can be found at http://www.stopbadware.org/.

VB2006 CALL FOR PAPERS

The deadline for submitting paper proposals for VB2006 is fast approaching. Abstracts of approximately 200 words must be sent as plain text files to editor@virusbtn.com, to arrive no later than **Thursday 9 March 2006**.

Submissions are invited on all subjects relevant to anti-malware and anti-spam. A list of topics suggested by the attendees of VB2005 can be found (along with further details of the paper submission and selection process) at http://www.virusbtn.com/conference/vb2006/2006call.xml.

VB2006, the Sixteenth Virus Bulletin International Conference, will take place 11–13 October 2006 at the Fairmont The Queen Elizabeth, Montréal, Canada.

ADDENDUM: THE FALSE POSITIVE DISASTER

Some concerns have arisen over the version of *ClamAV* tested for the article 'The false positive disaster', published in *Virus Bulletin* in November 2005 (see *VB*, November 2005, p.11). Rather than the product being compiled on the test machine directly from the source code, as is the developers' recommendation, a precompiled developer version of the product was downloaded from a third party's website (in good faith, since a link was provided on the *ClamAV* website). Therefore, the results obtained cannot be assumed to apply to the 'official' version of the product.

Prevalence Table – December 2005

Virus	Type	Incidents	Reports
Win32/Sober	File	11,597,547	98.16%
Win32/Mytob	File	96,390	0.82%
Win32/Netsky	File	84,022	0.71%
Win32/Mydoom	File	11,540	0.10%
Win32/Bagle	File	7,854	0.07%
Win32/Sdbot	File	4,889	0.04%
Win32/Lovgate	File	3,648	0.03%
Win32/Zafi	File	2,426	0.02%
Win32/Funlove	File	1,993	0.02%
Win95/Spaces	File	1,166	0.01%
Win32/Bagz	File	358	0.00%
Win32/Mimail	File	333	0.00%
Win32/Bugbear	File	311	0.00%
Win32/Pate	File	270	0.00%
Win32/Klez	File	222	0.00%
Win32/Mabutu	File	207	0.00%
Win32/Gibe	File	183	0.00%
Win32/Dumaru	File	181	0.00%
Win32/Valla	File	163	0.00%
Win32/Maslan	File	152	0.00%
Win32/Reatle	File	120	0.00%
Win95/Tenrobot	File	100	0.00%
Win32/Elkern	File	79	0.00%
Win32/Kriz	File	76	0.00%
Win32/Brepibot	File	72	0.00%
Win32/MyWife	File	72	0.00%
Win32/Bobax	File	55	0.00%
Win32/Mota	File	55	0.00%
Redlof	Script	52	0.00%
Win32/Swen	File	51	0.00%
Win32/Fizzer	File	42	0.00%
Win32/Chir	File	38	0.00%
Others[1]		406	0.00%
Total		11,815,073	100%

[1]The Prevalence Table includes a total of 406 reports across 65 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

NOT WORTHY

Peter Ferrie
Symantec Security Response, USA

The members of the RRLF virus-writing group were very proud when they released the first viruses for *Microsoft Shell* (see *VB*, November 2005, p.4), believing that these were the first viruses on the *Vista* platform. Of course, they were wrong: those are *Microsoft Shell* viruses, not *Vista* viruses. Then *Microsoft* announced that it would no longer be shipping *Microsoft Shell* with the first release of *Vista* in any case.

So what did the group do? They tried again. The second attempt at the 'first' *Vista* virus is called Idonus. However, this is not a *Vista* virus either – it's an *MSIL* virus. Give it up, guys.

IT GETS BETA AND BETA

MSIL/Idonus runs only on the *.NET* framework version 2.0, which has just been released. It is freely available from *Microsoft*, and can be installed on *Windows 98* (yes, indeed!), *Windows ME*, *Windows 2000, Windows XP* (if SP2 is installed), *Windows 2003* (if SP1 is installed) and, of course, *Vista* (which is currently at the Beta 1 stage).

The virus also requires the *WinFX Runtime Components Core* 3.0 to be installed (this includes the *Windows Presentation Foundation*, which is used to display the payload of the virus). *WinFX* is currently at the Beta 2 stage, is also freely available from *Microsoft*, and can be installed on *Windows XP* and *Windows 2003*.

The virus author wanted to call the virus 'Idoneus', from the Latin meaning 'suitable' or 'worthy'. If any virus were worthy of anything at all, this isn't it. The code looks awful, it was built in debug mode, which makes it look even worse, and it appears to be unfinished. Perhaps it is in the beta stage, too.

REGISTER HERE

Whenever the virus is executed, it creates a list in memory of all subdirectories under C:\. Then it attempts to open the registry key 'HKCU\Software\Retro'. If the registry key does not exist, the virus will create that key, then create the registry value 'Idoneus' within it. The virus sets the registry value data to 'c:\', followed by a directory name chosen randomly from the list it created. This is followed by the filename of the currently running program. The virus will also copy itself to the same randomly chosen directory, maintaining the name of the currently running program.

If the registry value 'Idoneus' exists, the virus reads it and deletes the file to which the registry value points, then copies itself to another randomly chosen directory, and rewrites the registry value with the newly chosen directory name. Thus, the virus moves around the drive each time it is executed.

The virus also creates the registry value 'Idoneus' under the registry key 'HKCU\Software\Microsoft\Windows\CurrentVersion\Run', and sets the registry value data to 'c:\', followed by the randomly chosen directory name and the filename of the currently running program. This ensures that the virus is executed each time the machine starts.

START HERE

The virus searches only in the current directory for files to infect, and only for those files whose suffix is '.exe'. For each file that the virus finds to infect, the virus reads its own code entirely into memory, then reads the victim's code entirely into memory, but then writes out only the virus code. Thus, the virus overwrites the host file.

It appears that the virus was intended to be a prepender (by writing out the host code afterwards, and including code to extract the host and run it), but perhaps the virus author was under pressure to release sooner.

GET THE MESSAGE

After the infection process has completed, the virus displays a message containing the virus name, the group's website and the text 'GeNeTiX is EVIL!'. It is not clear if the virus author is targeting a particular molecular biology industry company of that name, or the group that is campaigning against genetically-modified foods, or another group entirely.

CONCLUSION

The expression of controversial opinions in viruses is nothing new. We have seen, for example, anti-Israel comments in W32/Simile (see *VB*, May 2002, p.4), and other political messages in viruses such as W32/Maldal. However, using a virus to get the message across is not a good way to do it, especially when that virus destroys user data. Now that's evil.

MSIL/Idonus	
Size:	16,384 bytes.
Type:	Direct-action overwriter.
Payload:	Displays message box.
Removal:	Delete infected files and restore them from backup.

TECHNICAL FEATURE

INSIDE THE WINDOWS META FILE FORMAT

Peter Ferrie
Symantec Security Response, USA

The Windows Meta File (WMF) format has received a lot of attention over recent weeks. In this article we will find out a bit more about it, and then discover why it has been in the spotlight.

PICTURE THIS

A metafile is a collection of records. Most commonly (although not always), these are used to describe a picture. The contents of the records correspond to particular graphics device interface (GDI) functions which, when 'played', will produce the image.

The minimum size for a metafile is 18 bytes. A file of this size would contain the header and no records.

The format of the file header is as follows:

Offset	Size	Description
00	2	type: 1 (memory) or 2 (disk)
(some documentation states, incorrectly, that 0 is a valid value)		
02	2	number of words in header (must be 9)
04	2	version (0x100 or 0x300)
06	4	filesize in words
10	2	number of objects
12	4	maximum record size
16	2	number of parameters

Each record has the following format:

Offset	Size	Description
00	4	length of record
04	2	function number
06	n	record data

There is an extension to the WMF format, which is created by *Aldus*, and known as the 'placeable meta file'. The details of this format are not relevant here, except for the fact that a number of vulnerabilities in WMFs do not work if a placeable meta file is used. This is because the placeable meta file can only be used in a display device context, and cannot be printed.

Microsoft claims that the Escape function is disabled in placeable meta files, but in fact only certain subfunctions (most importantly, the SetAbortProc subfunction) are disabled. The relevance of this will become clear later.

STOP BUGGING ME

The final record in a WMF should be an EOF record. This is three words long, and its function number is zero. If the last record is not an EOF, *Windows* will parse the file searching for the EOF record. However, there are several bugs in the parsing process due to the fact that the file is assumed to be well formed.

First, if a zero-length record is encountered by *Windows* versions prior to *XP SP2*, the result is an infinite loop. This can be achieved with a 24-byte file. Although the bug was fixed in *Windows XP SP2*, it remains (at the time of writing) unpatched in previous versions of *Windows*, nearly two years after it was first disclosed.

The parser is supposed to scan the records from the start of the file to the end of the file, searching for the EOF record. However, since the values of the pointers are not checked in any way, the pointer to the next record may point backwards instead of forwards. It is possible for a backwards pointer to be followed by one or more forwards pointers, followed by another backwards pointer, and so on. Thus, it is vulnerable to circular linkages if a backwards pointer points to a list of forwards pointers that eventually point again to the same backwards pointer. All versions of *Windows*, including *XP SP2*, are vulnerable to this bug.

If the EOF record is found during the parsing, the in-memory copy of the file is truncated at that point, and the record count in the header is adjusted to account for the smaller size.

THE WHOLE HALF-TRUTH

The following is a list of the functions that, according to *Microsoft*, are the only functions supported by Windows Meta Files:

SetBkColor (1)	SetBkMode (2)
SetMapMode (3)	SetROP2 (4)
SetPolyFillMode (6)	SetStretchBltMode (7)
SetTextCharacterExtra (8)	SetTextColor (9)
SetTextJustification (10)	SetWindowOrgEx (11)
SetWindowExtEx (12)	SetViewportOrgEx (13)
SetViewportExtEx (14)	OffsetWindowOrgEx (15)
ScaleWindowExtEx (16)	OffsetViewportOrgEx (17)

ScaleViewportExtEx (18)	LineTo (19)
MoveToEx (20)	ExcludeClipRect (21)
IntersectClipRect (22)	Arc (23)
Ellipse (24)	FloodFill (25)
Pie (26)	Rectangle (27)
RoundRect (28)	PatBlt (29)
SaveDC (30)	SetPixel (31)
OffsetClipRgn (32)	TextOutA (33)
BitBlt (34)	StretchBlt (35)
Polygon (36)	Polyline (37)
Escape (38)	RestoreDC (39)
FillRgn (40)	FrameRgn (41)
InvertRgn (42)	PaintRgn (43)
SelectClipRgn (44)	SelectObject (45)
SetTextAlign (46)	Chord (48)
SetMapperFlags (49)	ExtTextOutA (50)
SetDIBitsToDevice (51)	SelectPalette (52)
RealizePalette (53)	AnimatePalette (54)
SetPaletteEntries (55)	PolyPolygon (56)
ResizePalette (57)	CreateDIBPatternBrush (66)
StretchDIBits (67)	ExtFloodFill (72)
DeleteObject (240)	CreatePalette (247)
CreatePatternBrush (249)	CreatePenIndirect (250)
CreateFontIndirect (251)	CreateBrushIndirect (252)

The truth is a little different however. We find that the DIBBitBlt (64) and DIBStretchBlt (65) functions exist, but are not listed. The CreateRectRgn (255) function is not listed either, but this exists in all versions of *Windows* including *Windows 3.x*. Finally, the SetLayout (73) function is not listed, but exists in *Windows 2000* and later.

SOMETHING LIKE THAT

As is often the case with unusual file formats, when an exploit appears, bad documentation follows it. In this case, there were descriptions of which of the fields were meaningful, and which were not.

While some of the documentation was correct (for example, the upper byte of the WMF function number is not checked, it serves merely as a hint to the number of parameters that are expected to be passed), some of it was not. For example, the 'number of objects' field was documented as being unnecessary, when in fact a valid value is required by the *Rgn functions and by SelectObject.

IN ... SECURE

Security seems not to have been a prime consideration when the WMF format was first introduced, and the programmer of the parser was incredibly trusting. As several of us found, a total of eight functions were vulnerable to 15 different buffer overflow conditions that could allow remote code execution. This prompted *Microsoft* to release security bulletin MS05-053. The vulnerable functions were:

AnimatePalette	SetPaletteEntries
PolyPolygon	DIBBitBlt
DIBStretchBlt	CreateDIBPatternBrush
CreatePalette	CreatePatternBrush

In fact, the CreateRectRgn function was also vulnerable, but an attack against this would require a file that was one gigabyte in size.

In addition to the denial-of-service attacks described above, at least 14 functions are known to be vulnerable to conditions that cause *Internet Explorer* on all platforms, and *Windows Explorer* on *Windows XP* (including *SP2*), to crash instantly upon opening malformed files. The vulnerable functions are the same as those listed for the buffer overflow functions above, including the CreateRectRgn function, with the addition of the following functions:

SetBkMode	TextOutA
BitBlt	StretchBlt
ExtTextOutA	SetDIBitsToDevice
StretchDIBits	

[W]ANT [M]ORE [F]REEDOM

One function is of particular interest in WMF format: the Escape function. The Escape function enables applications to bypass the GDI layer, and communicate directly with a particular device. This communication is intended to be directed to a printer, but the display device will accept some of the commands too.

The Escape function supports a number of subfunctions, most of which are related to printer control, such as StartDoc and StartPage, and the corresponding EndDoc and EndPage. Not surprisingly, at least three of these subfunctions contain bugs.

The bugs appear if a non-placeable WMF calls the StartDoc (3 or 4110) or StartPage (10) subfunction before any call is

made to CreateDC(). This is possible in *Windows Explorer* on *Windows XP*, for example, because there the created device context is compatible with both printer and display devices. The result is that the viewing application will crash. In order to attack the *Windows XP* platform, where the GDI+ layer exists, the minimum file length is 62 bytes.

Finally, we reach the most trusting part of the WMF format parser, which is the cause of most of the trouble: the SetAbortProc subfunction.

SETABORTPROC

The SetAbortProc function has existed since the days before *Windows 3.0*. That's over 15 years! It was implemented in the days of cooperative multi-tasking – before there were threads – in which an application was required to yield CPU control explicitly to other applications.

The function was designed to allow an application to cancel a print job once it had started, and the only way in which that could happen was through the use of a callback function that was called periodically. This was fine until the WMF format was introduced and the abort functionality was added to it. At that point, the WMF itself could carry its own abort handler. An image file containing executable code? It's unthinkable today, but that was then, this is now.

The Escape record subfunctions exist as part of the standard record data:

Offset	Size	Description
00	2	subfunction number
(all 16 bits are checked here)		
02	2	size of input structure
04	n	input structure

The SetAbortProc subfunction number has a value of 9, the value in the 'size of input structure' field is ignored, and the 'input structure' is the handler code.

While only one function handler can be registered at any one time, the SetAbortProc function can be called multiple times from within a WMF, so it is possible to register different handlers at different times during the parsing of the file. This allows for a variety of effects, and could have been used for a multi-stage attack, which would potentially have been difficult to detect.

Once the function handler is registered, it is called before each of the following records is parsed. Although the function is documented as being used to abort the printing of the image, an undocumented side effect is that it can also be used to abort the rendering of the image. It is not clear whether this particular behaviour is intentional, but if it is, that would explain why a device context does not have to refer only to a printer.

The function handler accepts two parameters. This leads to another bug: *Windows* does not check that those parameters are removed from the stack when the handler returns, so a sufficiently large (or circularly-linked) WMF can exhaust the stack space and cause a stack fault. If *Windows Explorer* attempts to display such a file, *Explorer* exits silently and suddenly, and no error message is displayed.

SERVICE D.E.P.ARTMENT

Windows XP SP2 introduced the Data Execution Prevention technology, which prevents pages that are marked as data from executing code. Its primary goal is to make it harder for buffer overflows to gain control of the CPU. A side effect is that it also stops the SetAbortProc function handler from executing, since the GDI does not mark the pages as executable.

AREA 51

Security researcher/commentator Steve Gibson has recently aired some controversial opinions on the WMF vulnerabilities, suggesting that they may not, in fact, have been accidental. However, in this case, his examination of 'exactly how it works' proved to be about as incorrect as it can get.

He claimed that the attack worked on *Windows 2000*. Presumably, that was by playing it through a dedicated application, since there is no default handler for WMFs on that platform, and *Internet Explorer* plays only placeable meta files which, as mentioned before, will not run the SetAbortProc function.

Gibson claimed that the record length must be set to 1 in order to run the code. This is untrue. The record length can be any value, as long as it remains within the bounds of the file *and* the next record function is not EOF. This last part is critical. The function is called only when the next record is reached, but processing stops when EOF is encountered. Thus, if the WMF contains only SetAbortProc and EOF, then only a malformed record length will point to something that remains within the file but does not point to an EOF record.

The reason why a value of 2 would not work is that the 'function number' field in the next record corresponds to the 'size of input structure' field in the SetAbortProc record. If the input size is set to zero, it will look like EOF.

The reason why a record length of 0 does not work is related to the zero-length bug described above. That bug

actually exists in two locations – one when parsing the file to find the EOF record, and one while parsing the file in order to render it. While the first case was fixed only in *Windows XP SP2*, the second bug was fixed in *Windows 2000* too. The second fix is relatively recent, though, since a default *Windows XP SP1* installation, for example, is vulnerable.

Gibson claimed that a thread is created to run the SetAbortProc handler. In fact, no thread is created to run the handler – it is a callback, which is called by the parser, and the parser has to wait until the callback returns, otherwise the whole point of the function (to abort the printing) is lost.

By his own admission, Gibson did not read the documentation (in fact, he claimed that he couldn't find it, although it is freely available on *Microsoft*'s website), and he claimed that the device context is not available to the function handler. Of course the device context is available to the function handler – it is one of the two parameters that is passed to it (see above), and it is required in order to abort the printing.

Finally, Gibson claimed that the control flow could not return to *Windows*. It is simply a matter of the function returning and discarding the parameters that were passed on the stack. If the record is well formed, *Windows* will continue to parse the file, as before.

I GUESS ...

Gibson admits that he was guessing about a number of things. Unfortunately, he guessed poorly. I guess we know better now.

CONCLUSION

So what are the consequences of the WMF bug and who really is vulnerable? It all comes down to the software that is installed on the machine.

Machines running *Windows XP* are vulnerable without user interaction, because *XP* has a default handler for WMFs that can be launched from within *Internet Explorer* without user interaction. Email programs, such as *Microsoft Outlook*, which support the display of media through an IFrame, are also a vector for system compromise when previewing or opening an email.

Earlier platforms, such as *Windows 9x*, *NT*, and *2000*, all contain the same vulnerability, but without a default handler they cannot be exploited in the same way. However, anyone using those platforms who has installed software that handles WMFs will be vulnerable to the same kind of attacks.

OPINION

LEARNING FROM SONY: AN EXTERNAL PERSPECTIVE

Dan Kaminsky
DoxPara Research, USA

'What happens when the creators of malware collude with the very companies we hire to protect us from that malware?' Bruce Schneier, one of the godfathers of computer security, was pretty blunt when he aired his views on the AV industry's disappointing response to the *Sony* rootkit (for an overview of the rootkit and its discovery see *VB*, December 2005, p.11). His question was never answered, which is fine, but his concerns were not addressed either, and that's a problem.

The incident represents much more than a black eye on the AV industry, which not only failed to manage *Sony*'s rootkit, but failed intentionally. The AV industry is faced with a choice. It has long been accused of being an unproductive use of system resources with an insufficient security return on investment. It can finally shed this reputation, or it can wait for the rest of the security industry to finish what *Sony* started. Is AV useful? The *Sony* incident is a distressingly strong sign that it is not.

All things being equal, I'd rather have the AV industry on our side. We take it for granted that there are customers for private computer security services. It didn't have to be this way: someone had to convince users that they were responsible for their own security. Because of the pioneering work of the AV industry, effective cryptography was non-negotiable, security research could be legitimate, and a free market for security technologies could form. Indeed, even the spread of broadband and WiFi would have failed if users hadn't been motivated to purchase firewalls to protect their new high-speed networks. The AV industry made sure the users knew they needed to protect themselves, which is why it is such a great problem when the AV industry refuses to protect them.

TAKING CONTROL

Let's be honest; the AV industry is blessed. What other software producers can depend on the operating system (for home users) or corporate IT departments (for the office) essentially demanding that their product runs on every system? When was the last time you saw a machine banned from a network for not running *Photoshop*?

What is it that customers think they're purchasing when they buy anti-virus software? Is it just safety? The safest machine is the one that is turned off. In fact, users are looking for something beyond mere safety. Users want control – and they're willing to pay for it.

We are in the business of putting force behind consent. Put simply, why ask for something if you can just take it? And let's make no mistake, *Sony* took control of people's systems. Whatever consent people may have granted initially to give *Sony* access to a system, it cannot be denied that *Sony* provided no mechanism for users to revoke that consent.

I often invite people into my home. I expect them to leave at some point, particularly if I ask them to. I certainly do not expect them to hide in my closet and pretend that they have gone. And if I call the police because the visitors won't leave, I don't expect them to argue with me about precisely what I agreed to when I first let them in.

Sony had a choice. DRM is unpopular software, as its primary purpose is to override user intent. *Sony* knew that some portion of users would want this stuff removed from their systems. They had the option to accept the revoked consent, and provide an uninstaller. Alternatively, they could simply ignore the need for user consent, take control of the system permanently, and simply prevent users from knowing there was anything to uninstall, by deploying a rootkit.

That the rootkit was exploitable by black hat hackers was bad, but ancillary to the argument. When the way you deal with users wanting to remove your code is by preventing users from *knowing* your code is running, not only are you operating without consent, but you know it, and everyone can tell.

BEEN THERE

Do we really expect the anti-virus industry to square up against companies when they are just trying to defend their copyright? Yes, absolutely. It's where the AV industry started.

We have just acknowledged the 20-year anniversary of the first PC virus, and almost everyone has missed the most interesting thing about it. Brain was not written by some random hacker, nor was it the nefarious creation of shadowy criminal groups. Brain was all too happy to identify its source:

```
Welcome to the Dungeon (c) 1986 Basit * Amjad (pvt)
Ltd. BRAIN COMPUTER SERVICES 730 NIZAM BLOCK ALLAMA
IQBAL TOWN LAHORE-PAKISTAN PHONE:
430791,443248,280530. Beware of this VIRUS....
Contact us for vaccination...
```

The first virus was written by an incorporated company. And why? *Brain* is still in business, so we can ask them. The following explanation can be found on the company's website (http://www.brain.net.pk/aboutus.htm):

> 'What no American journal had the courage to admit at that time was how badly the virus had hurt America's painfully cultivated image of the world's leading copyright protector. Almost overnight, it had shown Americans to be the world's biggest copyright violators. Every time the virus found a new home in the USA, it signalled one more copyright violation by an American.'

Malware in the pursuit of copy protection is nothing new; the first PC virus was an unambiguous and unapologetic attempt to protect copyrights, by any means necessary.

It has been 20 years. It is time to recognize the threat of corporate malware. It is not as if corporate malware is a concept with which anyone is unfamiliar. One of the most glaring failures of the computer security industry in recent years has been the failure to prevent the spread of spyware. It took years for the anti-virus industry to start responding to spyware. The first anti-spyware code was released in 2000. One major vendor released nothing until 2003. Millions of systems were infected while nothing was done.

Eventually, the AV industry adapted. I attended a wonderful talk, not long before the *Sony* story broke, where I heard about the extraordinary steps the anti-virus industry was taking to deal with what can basically be summarized as 'hackers with lawyers'. As awful as it is that we have to deal with peer businesses, instead of kids and criminals, it certainly seems that the industry has finally learned to respond to these threats.

DEAFENING SILENCE

But there really was no response to the *Sony* situation. *Sony* claimed a few AV companies as allies, in order to give its actions the patina of legitimacy. That didn't work. The idea of *Sony* and AV companies in talks was received about as well as if the AV companies had been negotiating with the author of Slammer, agreeing on which exploit he was allowed to hit next.

A few AV companies added code to their products to remove the cloaking component of the rootkit, but as far as I know, nobody actually removed the DRM components for which users were so clearly trying to retract consent. Only banal excuses, such as 'we're waiting for Sony to write an emergency uninstaller', were heard.

Do we wait for the authors of worms to release uninstallers to clean up their mess? Even if they did release one, should we trust people who have written malware in the past?

Given that *Sony*'s first uninstaller consisted of a patch to the latest version, and given that *Sony*'s code already had a history of unintended security side effects, it was not a surprise to witness multiple useless uninstallers coming out of *Sony* over the next six weeks. (And this was after *Sony*

had decided to behave and respond in an extraordinarily responsible manner!)

There is one industry that knows how to write an emergency uninstaller, one that's safe, effective and that can be released quickly. But the AV industry did nothing.

Some have claimed that it would actually have been illegal to have interfered with the *Sony* DRM, due to the Digital Millennium Copyright Act (DMCA). These claims have some merit – the US's DMCA does indeed take a rather dim view of subverting copyright protection mechanisms. Ignoring the fact that not all AV companies are American, and that not all victims were American, this legal interpretation opens up an astonishing attack vector.

Imagine a startup – we'll call it *MP3Solutions*. *MP3Solutions* would combine spyware with DRM. First, they'd design some code that detected watermarks in MP3 audio. Then, they'd offer $10 per deployment to independent third parties, 'no questions asked'. The code could be spread via worms, botnets, or drive-by web installs, but since the payload was copy protection software, the DMCA-fearing AV industry would just have to sit back and fail to protect anyone.

It really is amazing what happens once a user's consent to operate is considered optional. If this is really how the AV industry is interpreting the DMCA, that's astonishing and newsworthy. But the DMCA restrictions certainly would not have prevented the AV industry from complaining, or even asking for explicit permission to remove this particular piece of malware. Such permission seemed likely to be granted in this case: by the end of November, *Sony* was taking aggressive steps to manage the situation responsibly, providing free MP3s to affected customers and displaying a banner ad to inform users of their situation. The only thing *Sony* was having trouble with was an effective uninstaller – certainly they could have used the assistance of the AV industry!

Perhaps such a request was made, and permission was not forthcoming. It's possible. But another thing the DMCA does not do, is ban the provision of a warning to customers that the service they've purchased would be illegal in this instance: 'Software has been detected on your system whose operation you may not consent to, but which we are legally forbidden to remove. The vendor refuses to provide consent for us to remove this software for you. Please contact the following vendor address [link] to ask why.'

But instead, *Sony* got the benefit of the doubt.

We don't pay the AV industry to give *Sony* the benefit of the doubt. The AV industry cannot take money from users and provide services to *Sony*. I call upon every anti-virus company to state publicly that, the next time a media company tries to take control of users' PCs, and decides that the continued consent of the computer owner isn't necessary, they will act.

IN CONCLUSION

The AV industry in general failed to handle the *Sony* situation responsibly. I am confident that such a widespread failure will not be repeated – which means that those in the AV industry who do stand up and act will be well placed to take business from those in the industry who do not. By all accounts, the failure to respond to *Sony* was a business decision. *Sony* is a massive organization, one that possibly represents a significant opportunity. Why anger the giant?

You don't have to stand up for users. But your customers don't need to pay you. I spent many years working through security policies. Many demanded anti-virus software on every system. Not one of them cared about the size of the organisation that wanted the malicious code installed.

However big *Sony* is, the AV industry left even larger customers out to dry. (Military sites were hit. Does the military operate without anti-virus?) AV sales people should expect to be asked a simple question: why should anyone pay you to protect someone else?

I call upon every anti-virus vendor to state, solemnly and verifiably, that what happened with *Sony* was an anomaly – a misunderstanding based on an incomplete understanding of what customers demanded. No AV company expected this reaction. Certainly, *Sony* had no idea of the firestorm they were walking into. Did they ask? What were they told? Regardless, with this new data can come new policy.

I also call upon the AV industry to stop releasing bad data. I do apologise for implying publicly that AV companies knew precisely how many *Sony*-infected nodes were out there. You can't manage what you can't measure, and thus I had assumed that AV companies were measuring what they were trying to manage. I know now that some of them just look at how many tech support calls they get, and extrapolate.

That is awful. The plural of 'anecdote' is not 'data'. Expect any further releases of numbers to have their methodology questioned. As for my own data – those who are curious about my own methodology for tracking the *Sony* rootkit are welcome to look through the 85 gigabytes of anonymised compressed DNS traffic that I used to build my estimates (see http://www.doxpara.com/?q=sony). One researcher with a high-speed net connection should not have better data on a global scale malware attack than companies with customers paying them to manage that malware. And yet, I have almost a tenth of a terabyte, and they have tech support calls.

I invite the AV industry to do better.

FEATURE

SIGNATURE UPDATES VS BCP

Aleksander Czarnowski
AVET Information and Network Security, Poland

From time to time we witness events that seem so unlikely or unwanted that they almost defy belief. Occasionally, such an event occurs within the IT security business. This article has been inspired by events that have been described by many as 'unbelievable'.

THE STORY

The story is short and simple: recently, a local AV vendor had some serious problems with producing signature updates for its product, and failed to update its scanning engine for as long as two weeks. (Unfortunately it seems that, at the time of writing, the problems have yet to be resolved and updates are still sporadic and infrequent.)

Would you ever expect a well established anti-virus company to fail to provide you with signature updates while the company was still operational? Many security officers probably did not. We have seen buyouts of anti-virus companies, and even bankruptcy in the past, but in these cases measures have usually been put in place to ensure continuity in malware protection for customers.

Broken or invalid signature updates are also something with which we are familiar, but this situation is something new and worrying – especially considering that the ie_xp_pfv_metafile [1] exploit was used widely and *Microsoft* security bulletins MS06-001 [2], MS06-002 [3] and MS06-003 [4] were all released during the period in which no updates were provided.

LEARNING FROM THE PAST

This got me thinking about the past. Historically, security policies have been shaped by critical events. Consider, for example, corporate security policies dated prior to 2001. In how many would you find reference to scenarios involving terrorist attacks or BCP (business continuity planning)?

Despite the various mathematical models we use for risk analysis, we always seem to learn the hard way in the security area. So what we can learn from this story? I think the following are the most important points:

- The failure of a safeguard may not always be the result of a direct, easily foreseen technical issue. Even risk management-driven security policy can be flawed simply due to incomplete threat and risk catalogs. This might pose an even more important question: is risk management the right approach? After all, in evaluating risks and threats we rely partly on historical data. If a particular event has a very rare occurrence, then we might wrongly ignore it.
- The defence-in-depth strategy suggests that we should never rely on one safeguard to protect a particular asset. This may be tricky to implement in the case of malware protection as many organizations use a single product that operates at different levels of the network.
- The use of a multiple-engine product won't necessarily provide continuity in malware protection if, for example, the vendor of that product encounters problems.

Some might say that the situation described above is unlikely to happen where the well established vendors who operate worldwide (the 'big players') are concerned. Try telling that to *Arthur Andersen* or *Enron* shareholders.

We have to ask whether our security policies and BCPs are ready to deal with such a situation. It seems that using two different products from different vendors (based on different engines from different vendors) could be a wise move.

The introduction of stack protection mechanisms and IDS/IPS systems might seem like a good solution too. But we could be very wrong – for example, the MS06-001 [2] vulnerability is not stack overflow-based – so we should remember that DEP and MS/GS mechanisms are not the final solution to system security. While it's easy to filter out well known attack web servers that contain exploits, it's far from being the final solution – even in the case of this particular vulnerability. Not every vulnerability exploitation process is easy to detect using a signature-based approach – even methods based on code emulation can have serious problems. Along with Dave Aitel [5], I'm curious as to how IDS/IPS vendors will approach this problem.

So as you can see, we have entered the new year with new vulnerabilities and new challenges. I wonder what the maximum length of time is that an AV vendor can stay operational without providing updates. I hope that none of *VB*'s readers will ever have to find out.

BIBLIOGRAPHY

[1] MetaSploit Framework: http://www.metasploit.com/.

[2] MS06-001: http://www.microsoft.com/technet/security/bulletin/ms06-001.mspx.

[3] MS06-002: http://www.microsoft.com/technet/security/bulletin/ms06-002.mspx.

[4] MS06-003: http://www.microsoft.com/technet/security/bulletin/ms06-003.mspx.

[5] Message: [Dailydave] Commander Keen in Fonts, 14 January 2006, http://lists.immunitysec.com/pipermail/dailydave/2006-January/002828.html.

COMPARATIVE REVIEW

WINDOWS NT 4 WORKSTATION

Matt Ham

Windows NT is such an ancient platform that writing a review for it seems more akin to writing about history than present-day affairs. The platform is still used by a fair number of people the world over though, so the review will be relevant to many.

For a reviewer, both very old and very new platforms are of great interest. When products are tested on very new platforms one tends to see many oddities as developers struggle to accommodate unexpected technology, while products tested on the very old platforms have the potential to be utterly broken due to these very struggles. *Symantec*, for example, no longer supports *Windows NT* in its most recent product line (*SAV 10*), and thus *SAV 9* was submitted for test here.

That said, I was fully expecting a bumper harvest of VB 100% awards on this occasion, simply due to the familiarity of the platform to developers.

TEST SETS

The test sets were aligned to the October 2005 WildList, which was the most recent edition available on the product submission date, 9 January 2006.

The overwhelming majority of new samples in the test sets were of W32/Mytob, with close to 50 new variants added this time. Other additions were also predominantly bot-related – perhaps *VB* should consider *Bot Bulletin* as an alliterative name change.

Alwil avast! Professional 4.6.750 0602-1

ItW Overall	100.00%	**Macro**	99.56%
ItW Overall (o/a)	100.00%	**Standard**	98.92%
ItW File	100.00%	**Polymorphic**	93.58%

avast! is the first of a small number of products in this test in which archive scanning is not activated by default. Where scanning and detection were concerned, however, default settings seemed to have been well chosen.

A selection of files were missed – primarily polymorphics and some macro samples – though none of these were in the In the Wild (ItW) test set and no false positives were generated when scanning clean files. *avast!* is thus the first product to receive a VB 100% award in this test.

Avira Avira Desktop 1.00.00.80

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

With a string of recent good results behind it, *Avira* had ample opportunity in this test to fall from grace and little to improve. In the event, however, test results were exactly the same as the last time the product was tested: full detection in all sets.

With no false positives in the clean test sets, this performance gains *Avira* another VB 100%.

CA eTrust Antivirus 7.1.501 (InoculateIT engine 12.4.2034)

ItW Overall	100.00%	**Macro**	99.90%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	99.89%

As usual, this version of *eTrust* is included here for information only, since the *InoculateIT* is provided within the *eTrust* installation, but not activated by default.

Also as usual, the logging functions within *eTrust* remain utterly abominable – screenshots being more useful than the dumped log versions available from within the scanner.

It was notable that the *InoculateIT* version of *eTrust* detected more viruses than the default *Vet* engine on this occasion.

CA eTrust Antivirus 7.1.501 (Vet engine 23.71.42)

ItW Overall	100.00%	**Macro**	99.82%
ItW Overall (o/a)	100.00%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	99.95%

Much of the comment about *eTrust* has already been made under the previous section, and with the interface here being identical, there remains little to comment on other than the scanning results.

With 100% detection of samples in the ItW test set and no false positives generated, these were sufficient to guarantee a VB 100% for *eTrust* when using its default *Vet* engine.

CAT Quick Heal 2006 8.00

ItW Overall	100.00%	**Macro**	98.18%
ItW Overall (o/a)	100.00%	**Standard**	96.48%
ItW File	100.00%	**Polymorphic**	96.26%

Quick Heal remains a fast and easy product to test, and its performance was once more sufficient for the product to earn a VB 100% award. There was little else to note about *CAT*'s product, so this remains a short write up.

Command AntiVirus 4.93.6

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.72%
ItW File	100.00%	**Polymorphic**	100.00%

Logging became a problem while testing *Command AntiVirus*, with logs available only in RTF format – one of the least friendly formats for automated parsing. Since all logs were truncated in any case, they were sufficiently useless that parsing was not attempted. Thankfully, the number of misses was small enough that manual inspection of the truncated logs, combined with scan summary information, could easily pin down the missed files on demand. Such a small number of misses is always a promising sign, and indeed *Command AntiVirus* receives a VB 100%.

Dr.Web Dr.Web Scanner 4.33

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

Before installation of *Dr.Web* would complete, a version of psapi.dll needed to be installed on the machine, in order to make on-access scanning possible. A great change was noted in the on-access scanner: it seems that a reboot is no longer necessary after making changes in the on-access scanner configuration. After many years of constant restarts during testing, this came as a happy event.

Probably more happily for the developers, the number of missed files remains very low – the only files missed were during on-access scanning, and then only if in EML or ZIP format. It comes as no surprise, therefore, that a VB 100% award is in order.

Eset NOD32 1.1358

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

NOD32 was the second product in this test that required archive file scanning to be activated when testing the zipped clean files. With an otherwise uneventful set of tests I was able to come up with only one event of note: the log file for on-demand scanning was 1337 kb in size – clearly this is highly significant if one favours conspiracy theories or numerology. Less shocking will be the news that *Eset* gains a VB 100% as a result of the tests.

Fortinet FortiClient 2.0.180

ItW Overall	100.00%	**Macro**	99.90%
ItW Overall (o/a)	100.00%	**Standard**	99.62%
ItW File	100.00%	**Polymorphic**	96.21%

FortiClient's performance was sufficient for another VB 100% to be added to *Fortinet*'s collection. The misses that remain are scattered through the test sets to such an extent that no real pattern emerges. One suspects that results will improve gradually.

FRISK F-Prot Antivirus 3.16f

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

Perhaps to make the log files seem more interesting, a very large amount of information was included – though it was at least easy to filter out during parsing. Only one sample was missed during on-demand scanning, in the standard test set, though a few more misses were added on-access. None of these are currently rated as In the Wild, however, so a VB 100% is awarded.

F-Secure Anti-Virus 5.44 11411

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.85%
ItW File	100.00%	**Polymorphic**	100.00%

On-demand tests	ItW File		ItW Boot		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
Alwil avast!	0	100.00%	0	100.00%	100.00%	18	99.56%	112	93.58%	22	98.92%
Avira Avira Desktop	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
CA eTrust Antivirus (I)	0	100.00%	0	100.00%	100.00%	4	99.90%	1	99.89%	0	100.00%
CA eTrust Antivirus (V)	0	100.00%	0	100.00%	100.00%	12	99.82%	1	99.95%	0	100.00%
CAT Quick Heal	0	100.00%	0	100.00%	100.00%	75	98.18%	311	96.26%	100	96.48%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	2	99.72%
Dr.Web Dr.Web	0	100.00%	0	100.00%	100.00%	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%
Fortinet FortiClient	0	100.00%	0	100.00%	100.00%	4	99.90%	264	96.21%	11	99.62%
FRISK F-Prot Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	1	99.82%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.85%
GDATA AntiVirusKit	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	257	85.97%	27	98.56%
Hauri ViRobot Desktop	0	100.00%	0	100.00%	100.00%	12	99.71%	12	99.75%	21	98.81%
H+BEDV AntiVir	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
McAfee VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
MicroWorld eScanWin	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
NTWI Virus Chaser	0	100.00%	0	100.00%	100.00%	3	99.93%	5	99.98%	13	98.96%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	180	91.24%	4	99.71%
SOFTWIN BitDefender	0	100.00%	0	100.00%	100.00%	34	99.12%	9	99.71%	17	99.27%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	8	99.80%	0	100.00%	13	99.43%
Symantec AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
UNA UNA	2	99.85%	0	100.00%	99.85%	1904	54.75%	11991	32.40%	381	83.38%
VirusBuster Professional	1	99.98%	0	100.00%	99.98%	0	100.00%	124	92.59%	25	98.90%

FSAV is a product where very small numbers of misses are something of a habit. On this occasion the product missed only the stored .TMP sample of W32/Nimda.A on demand. On access, the total was increased by the two zipped samples of W32/Heidi. However, since these are all currently in the standard test set (not In the Wild), *F-Secure* is also the recipient of a VB 100%.

GDATA AntiVirusKit 14.1.2

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

Continuing the theme, *AVK* managed to miss even fewer samples than the previous products – no samples went

undetected. With no false positives, it goes without saying that these results earn *AVK* a VB 100% award. However, the product's scanning performance does come at a small expense, with a slightly slow scan rate as a side effect. The trade off between detection and scanning speed is a common dilemma for anti-virus developers, with many misses in these tests occurring as a result of pragmatism, with developers opting for faster on-access scanning at the cost of some detection when files are being manipulated rather than executed.

Grisoft AVG Anti-Virus 7.1 371

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	98.56%
ItW File	100.00%	**Polymorphic**	85.97%

AVG missed a number of files in the various test sets, though none were classified as In the Wild. Of those missed, the majority were polymorphic in nature or packaged in slightly unusual formats. With no false positives and full detection of ItW viruses, *AVG* earns itself a VB 100%.

Hauri ViRobot Desktop 5.0

ItW Overall	100.00%	**Macro**	99.71%
ItW Overall (o/a)	100.00%	**Standard**	98.81%
ItW File	100.00%	**Polymorphic**	99.75%

ViRobot held the dubious distinction of having by far the largest number of false positive detections in this test. Six files were reported as infected, while a further clean file was declared to be suspicious. As a result, the product does not qualify for a VB 100% this month. This will be something of a disappointment, since detection rates were respectable.

H+BEDV AntiVir 6.33.00.02 1127

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

Other than a lower price and older graphics, *AntiVir* is essentially identical to *Avira* internally and thus similar scanning results were expected. This was indeed the case. Minor variations in the scanning throughput rates were noted, though with *Windows* being host to numerous unpredictable background processes, it would be surprising if results here were found to be identical.

Kaspersky Anti-Virus Personal 5.0.388

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

The ever-productive interface developers at *Kaspersky* have been at work once more for this version. Personally, I am less of a fan of this latest incarnation than the previous interface, though this is more due to unfamiliarity than any obvious faults. The only oddity noted was in the 'time remaining' bar on the scanning interface, which demonstrated some interesting time dilation and compression phenomena.

On the detection front, however, there were few changes to be seen. Two zipped W32/Heidi samples on access were the sum total of missed files, leaving *Kaspersky* the holder of a VB 100% yet again.

McAfee VirusScan Enterprise 8.00 4400 4669

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

VirusScan was the third and last of the products in this test to require manual activation on archive file scanning during clean set tests. It also showed notable differences between scanning on access and on demand, with several samples of W32/Etap missed on access. No misses were noted on demand, however, and no false positives surfaced either. *McAfee* thus receives a VB 100% award for *VirusScan*'s performance.

MicroWorld eScanWin 8.0.641.1

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

As a rebadged version of the *GDATA* product, *eScanWin* might be expected to show similarities to that product, despite being blue in places rather than yellow.

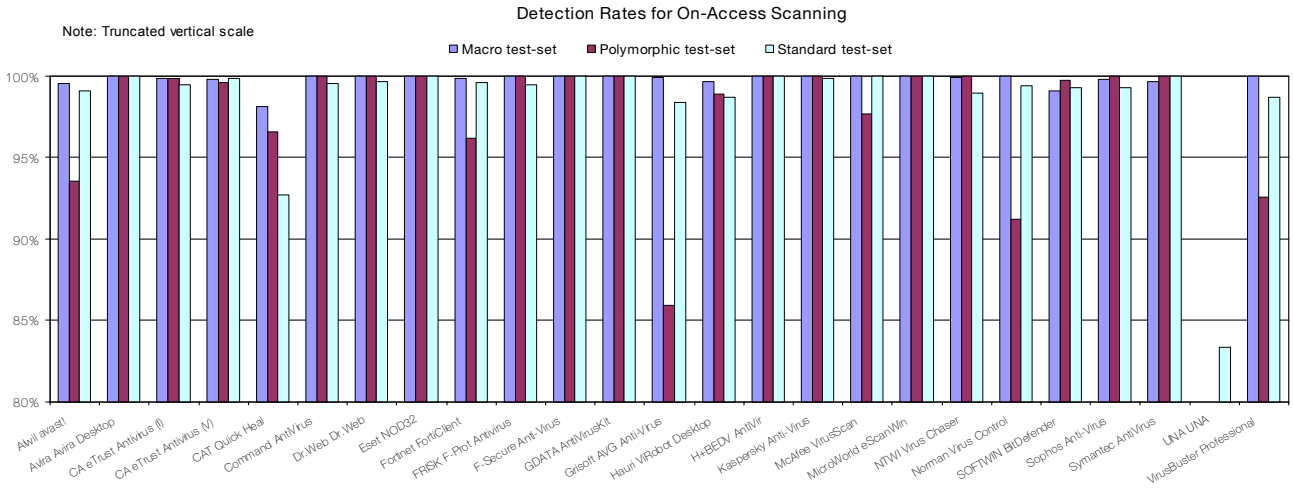
On-access tests	ItW File		ItW Boot		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
Alwil avast!	0	100.00%	0	100.00%	100.00%	18	99.56%	112	93.58%	19	99.07%
Avira Avira Desktop	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
CA eTrust Antivirus (I)	0	100.00%	0	100.00%	100.00%	4	99.90%	1	99.89%	4	99.51%
CA eTrust Antivirus (V)	0	100.00%	0	100.00%	100.00%	12	99.82%	70	99.62%	3	99.84%
CAT Quick Heal	0	100.00%	0	100.00%	100.00%	79	98.11%	241	96.58%	158	92.72%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	5	99.58%
Dr.Web Dr.Web	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.69%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet FortiClient	0	100.00%	0	100.00%	100.00%	4	99.90%	264	96.21%	11	99.62%
FRISK F-Prot Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	6	99.97%	7	99.49%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	1	99.98%
GDATA AntiVirusKit	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG Anti-Virus	0	100.00%	0	100.00%	100.00%	3	99.93%	257	85.97%	30	98.41%
Hauri ViRobot Desktop	0	100.00%	0	100.00%	100.00%	12	99.71%	48	98.91%	23	98.69%
H+BEDV AntiVir	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	2	99.88%
McAfee VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	29	97.67%	0	100.00%
MicroWorld eScanWin	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
NTWI Virus Chaser	2	99.72%	0	100.00%	99.73%	3	99.93%	5	99.98%	13	98.96%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	180	91.24%	12	99.45%
SOFTWIN BitDefender	0	100.00%	0	100.00%	100.00%	35	99.10%	8	99.72%	17	99.27%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	8	99.80%	0	100.00%	15	99.30%
Symantec AntiVirus	3	99.97%	0	100.00%	99.97%	17	99.68%	0	100.00%	0	100.00%
UNA UNA	2	99.85%	0	100.00%	99.85%	1904	54.75%	11991	32.40%	381	83.38%
VirusBuster Professional	1	99.98%	0	100.00%	99.98%	0	100.00%	126	92.58%	29	98.73%

Somewhat disturbingly, however, there was a major difference, in that the on-access scanner crashed during testing. This occurred only once though, so did not seem easily reproducible. Happily, the differences in performance did not extend to detection capabilities and, with 100% detection of ItW viruses and no false positives, *eScanWin* also gains a VB 100%.

New Technology Wave Inc. Virus Chaser 5.0a

ItW Overall	100.00%	**Macro**	99.93%
ItW Overall (o/a)	99.73%	**Standard**	98.96%
ItW File	100.00%	**Polymorphic**	99.98%

Installation of *Virus Chaser* failed initially due to the requirement of a new version of mfc42.dll. Installing the



redistributable C++ libraries on the machine solved this problem.

Somewhat less easily solved was the total lack of control of on-access scanning available within the program. In the end, scanning was performed while locking an appropriate key in a depressed position – scanning in this way taking a little over 24 hours to complete. On demand, scanning progressed more easily, though the logs must have been the creation of either a sadist or a fan of complex logic problems. Overall, there was an impressive degree of user unfriendliness in this product.

Such irritations aside, the product's scanning performance was less than awesome, with a smattering of misses across the test sets. The fact that samples of W32/Yaha.G and W32/Yaha.E were missed on access was sufficient to deny *Virus Chaser* a VB 100% on this occasion.

Norman Virus Control 5.81

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.71%
ItW File	100.00%	**Polymorphic**	91.24%

Minor improvements seem to have been made to the creation of new tasks in *Norman Virus Control* of late, since the process seemed less painful than it has done in the past. Of course, this could merely be due to the fact that I have gained familiarity with the interface, but either way the effect was appreciated.

When the logs were analysed the results were much as expected: some polymorphic and a few other samples were missed, but with no ItW samples missed and no false positives, *NVC* is a VB 100% winner.

SOFTWIN BitDefender Professional Plus 9 9

ItW Overall	100.00%	**Macro**	99.12%
ItW Overall (o/a)	100.00%	**Standard**	99.27%
ItW File	100.00%	**Polymorphic**	99.71%

BitDefender continues to be a solid performer in our tests, with little in the way of comment necessary. It is presumably this solidity which has led to its being the basis of detection in several other products, including *Hauri*'s offering in this test. *SOFTWIN* will be pleased that none of the false positive issues apparent with that derived product were present in *BitDefender*, thus entitling it to a VB 100%.

Sophos Anti-Virus 4.5.8 2.32.6 4.01

ItW Overall	100.00%	**Macro**	99.80%
ItW Overall (o/a)	100.00%	**Standard**	99.43%
ItW File	100.00%	**Polymorphic**	100.00%

Of note in *Sophos*'s clean file scans was the fact that archive scanning is now activated by default. This is a recent and much appreciated configuration change. With both detection and lack of false positives in their usual respectable state, *Sophos* earns itself a VB 100% award. That said, logging functions were not without their niggles, with various unnecessary spaces added to lines which serve no purpose but to make parsing a little more complex. Meanwhile, archives are designated merely by appending \[archivename] directly to the path in which the infected archive is located. This ensures that parsing is made more complex for these entries and would be an ideal place to use the spare spaces just mentioned.

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)
Alwil avast!	254.0	2153.3		25	3173.4		84	1897.8	26	2869.5
Avira Avira Desktop	485.0	1127.7		17	4666.7		216	738.0	28	2664.6
CA eTrust Antivirus (I)	252.0	2170.4		15	5288.9		92	1732.8	21	3552.7
CA eTrust Antivirus (V)	250.0	2187.7		15	5288.9		97	1643.5	20	3730.4
CAT Quick Heal	174.0	3143.3		103	770.2		40	3985.4	23	3243.8
Command AntiVirus	220.0	2486.1		16	4958.4		73	2183.8	12	6217.3
Dr.Web Dr.Web	434.0	1260.2		37	2144.2		123	1296.1	22	3391.2
Eset NOD32	132.0	4143.4		17	4666.7		76	2097.6	20	3730.4
Fortinet FortiClient	469.0	1166.2		29	2735.6		173	921.5	18	4144.9
FRISK F-Prot Antivirus	267.0	2048.4		17	4666.7		101	1578.4	15	4973.8
F-Secure Anti-Virus	294.0	1860.3		39	2034.2		119	1339.6	31	2406.7
GDATA AntiVirusKit	631.0	866.8		31	2559.2		220	724.6	38	1963.4
Grisoft AVG Anti-Virus	364.0	1502.6		19	4175.5		72	2214.1	17	4388.7
Hauri ViRobot Desktop	565.0	968.0	6 [1]	47	1688.0		162	984.1	41	1819.7
H+BEDV AntiVir	480.0	1139.4		15	5288.9		215	741.5	28	2664.6
Kaspersky Anti-Virus	461.0	1186.4		52	1525.6		153	1041.9	36	2072.4
McAfee VirusScan	185.0	2956.4		23	3449.3		91	1751.8	24	3108.6
MicroWorld eScanWin	488.0	1120.8		50	1586.7		206	773.9	46	1621.9
NTWI Virus Chaser	285.0	1919.1		28	2833.3		97	1643.5	18	4144.9
Norman Virus Control	603.0	907.0		22	3606.1		228	699.2	17	4388.7
SOFTWIN BitDefender	416.0	1314.7		21	3777.8		145	1099.4	15	4973.8
Sophos Anti-Virus	208.0	2629.5		24	3305.6		72	2214.1	21	3552.7
Symantec AntiVirus	368.0	1486.2		34	2333.3		111	1436.2	30	2486.9
UNA UNA	198.0	2762.3		21	3777.8		87	1832.4	24	3108.6
VirusBuster Professional	326.0	1677.7	[1]	41	1935.0		132	1207.7	32	2331.5

Symantec AntiVirus 9.0.0.338 51.3.0.11

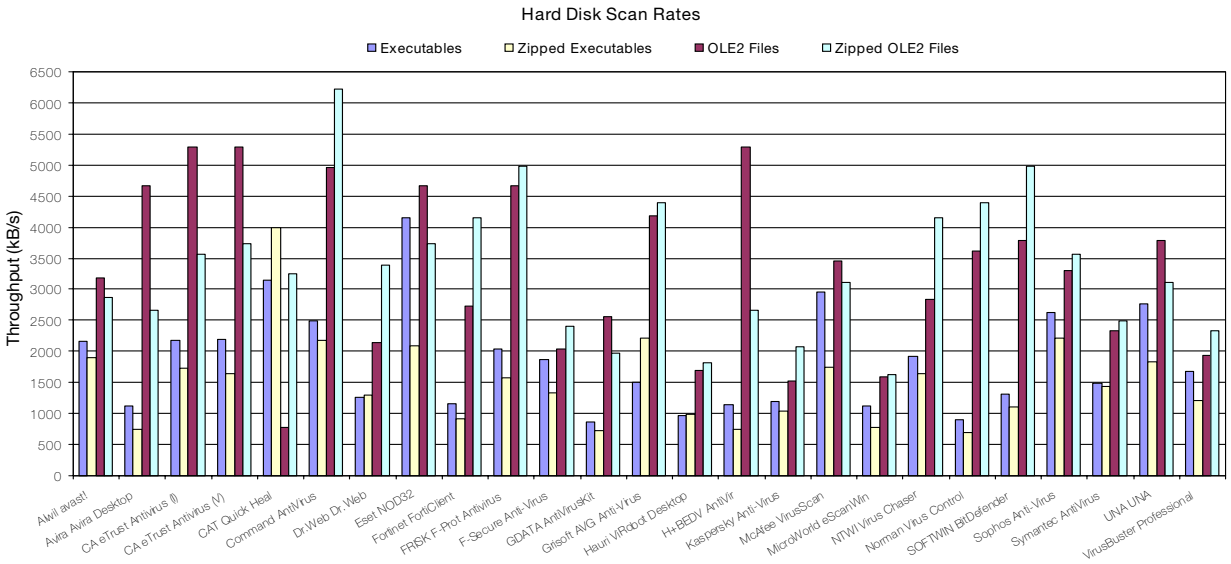
ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	99.97%	**Standard**	100.00%
ItW File	100.00%	**Polymorphic**	100.00%

SAV's scanning speed was far slower with some settings activated than with others. Default scanning settings are not pleasant when large numbers of infected files are present, though acceptable when files are mostly clean.

Unfortunately, it seems that on-access no POT or PPT files were checked in the default mode, thus resulting in samples of O97M/Tristate.C being missing in the ItW test set and no VB 100% being awarded this time.

UNA UNA 1.83

ItW Overall	99.85%	**Macro**	54.75%
ItW Overall (o/a)	99.85%	**Standard**	83.38%
ItW File	99.85%	**Polymorphic**	32.40%



The good news for *UNA* is that scanning was fast and no false positives were flagged. The bad news is that there were a multitude of missed detections in every test set. Although only two files were missed In the Wild both on access and on demand, this is ample reason to deny a VB 100%.

VirusBuster Professional 2005 5.001 41

ItW Overall	99.98%	**Macro**	100.00%
ItW Overall (o/a)	99.98%	**Standard**	98.90%
ItW File	99.98%	**Polymorphic**	92.59%

VirusBuster was perhaps the most troubled of all the products. First, it required mfc42.dll to be installed – a hurdle that was easily passed. When scanning on access, however, the scanner failed repeatedly. This failure was silent, with no indication other than the fact that no files were being checked. It seemed reproducible, simply by passing around 6,000 infected files through the on-access scanner. Woes were to continue in the clean sets too, where a suspicious file was noted. Matters on the detection front were no more inspiring. Despite having .EML files flagged for scanning, the .EML version of W32/Nimda.A was missed both on access and on demand. A VB 100% award is thus out of reach for *VirusBuster* this month.

CONCLUSION

The biggest surprise for me in this test was not the products that failed to detect virus samples, but the issues concerning operating system support. *NOD32*, for example, included the Microsoft C++ foundation classes as part of its installation package and asked whether they should be installed. Several products, however, were missing DLLs when installed onto the *Windows NT* platform. This shows a little lack of care for *Windows NT*, even if it is aged and mostly ignorable as far as new installations are concerned.

The instabilities noted with on-access scanning are more worrying, and presumably due to the operating system rather than any basic software flaws, since the same issues have not been noted with these products on other platforms. Essentially, the developers are caught between the most modern and most ancient incarnations of the *NT* operating systems and the desire to produce one package which will install on every variant. With the differences apparent between *Windows XP* and *Windows NT*, this is obviously easier said than done.

While *Microsoft* can drop support for a platform, the same is not true for developers. Without the *Microsoft* monopoly to back them up, anti-virus developers can gain customers by their range of supported platforms, and lose them if they cut back when a customer demands support for machines of more historical than practical interest. One wonders whether *Microsoft*'s entry into anti-virus, currently restricted to *Windows*-only platforms, will be influenced by this in future.

Technical details

Test environment: Three 1.6 GHz Intel Pentium 4 workstations with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy, all running *Windows NT 4 Workstation SP 6*.

Virus test sets: Complete listings of the test sets used can be found at http://www.virusbtn.com/Comparatives/WinNT/2006/test_sets.html.

A complete description of the results calculation protocol can be found at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

END NOTES & NEWS

Techworld is organising a free, half-day seminar on 2 February 2006 in London, UK. The seminar, entitled 'Endpoint Security: how to protect your system from its users', will focus on areas such as threat trends, patch management, securing endpoints, virus protection and configuration management. See http://www.techworld.com/.

RSA Conference 2006 will be held 13–17 February 2006 in San Jose, CA, USA. For more details including the full agenda and online registration see http://2006.rsaconference.com/us/.

The Black Hat Europe 2006 Briefings & Training will be held 28 February to 3 March 2006 in Amsterdam, The Netherlands. For details including online registration see http://www.blackhat.com/.

The 9th annual WEBSEC conference takes place 27–31 March 2006 in London, UK. The event will include live hacking demos, a network and application hacker challenge, more than 40 sessions on topical security issues including a panel debate in which *Virus Bulletin*'s Technical Consultant Matthew Ham will be a panel member. For more details see http://www.mistieurope.com/.

The 2nd Information Security Practice and Experience Conference (ISPEC 2006) will be held 11–14 April 2006 in Hangzhou, China. For details see http://ispec2006.i2r.a-star.edu.sg/.

Infosecurity Europe 2006 takes place 25–27 April 2006 in London, UK. For details or to register interest in the event see http://www.infosec.co.uk/.

The 15th EICAR conference will take place from 29 April to 2 May 2006 in Hamburg, Germany. Authors are invited to submit posters for the conference. The deadlines for submitting poster presentations is 24 February 2006. For more information see http://conference.eicar.org/2006/.

The Seventh National Information Security Conference (NISC 7) will take place from 17–19 May 2006 at St. Andrews Bay Golf Resort & Spa, Scotland. Enquiries may be directed to tina.deighton@sapphire.net or via http://www.nisc.org.uk/ .

The 2006 IEEE Symposium on Security and Privacy will be held 21–24 May 2006 in Oakland, CA, USA. For details see http://www.ieee-security.org/TC/SP2006/oakland06.html.

AusCERT 2006 takes place 21–25 May 2006 in Gold Coast, Australia. A programme overview, providing a list of confirmed speakers, can be found at http://conference.auscert.org.au/.

The Fourth International Workshop on Security in Information Systems, WOSIS-2006, will be held 23–24 May 2006 in Paphos, Cyprus. For details see http://www.iceis.org/.

CSI NetSec '06 takes place 12–14 June 2006 in Scottsdale, AZ, USA. Topics to be covered at the event include: wireless, remote access, attacks and countermeasures, intrusion prevention, forensics and current trends. For more details see http://www.gocsi.com/.

Black Hat USA 2006 will be held 29 July to 3 August 2006 in Las Vegas, NV, USA. The call for papers opened on 2 February and online registration for the event will be available from 15 March. See http://www.blackhat.com/.

The 15th USENIX Security Symposium takes place 31 July – 4 August 2006 in Vancouver, B.C., Canada. A training programme will be followed by a technical programme, which will include refereed papers, invited talks, work-in-progress reports, panel discussions and birds-of-a-feather sessions. A workshop, entitled Hot Topics in Security (HotSec '06), will also be held in conjunction with the main conference. For more details see http://www.usenix.org/.

HITBSecConf2006 will take place 16–19 September 2006 in Kuala Lumpur. More details and a call for papers will be announced in due course at http://www.hackinthebox.org/.

The 16th Virus Bulletin International Conference, VB2006, will take place 11–13 October 2006 in Montréal, Canada. *Virus Bulletin* is currently seeking submissions from those interested in presenting papers at the conference (see p.3). For details of sponsorship opportunities, please email vb2006@virusbtn.com. Online registration and further details will be available soon at http://www.virusbtn.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

- Single user: $175
- Corporate (turnover < $10 million): $500
- Corporate (turnover < $100 million): $1,000
- Corporate (turnover > $100 million): $2,000

See http://www.virusbtn.com/virusbulletin/subscriptions/ for subscription terms and conditions.

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 Web: http://www.virusbtn.com/

vbSpam supplement

CONTENTS

NEWS & EVENTS

HONG KONG PROPOSES ANTI-SPAM BILL

Hong Kong's Commerce, Industry & Technology Bureau has revealed its proposals for anti-spam legislation and launched a two-month public consultation.

The proposed Unsolicited Electronic Messages Bill adopts a 'technology-neutral' approach, in a bid to accommodate any new forms of electronic message that may appear in the future as well as the forms of messaging that are affected by spam at the present time. The proposed regulation adopts an opt-out strategy, requiring senders of commercial electronic messages to stop sending further such messages if the recipient asks them to. This, according to Hong Kong's Secretary for Commerce, Industry and Technology will 'provide companies with room to promote their products, and in turn facilitate development of small and medium sized enterprises. It also provides opportunities for recipients to browse through promotion information before deciding whether to receive further messages.'

Under the proposed Bill, convicted spammers would be liable to a maximum fine of 100,000 HK dollars (approx. US $12,000), and 1,000 HK dollars (approx. US $129) per day for repeated offences. The period of public consultation continues until 20 March.

SPAMMER TO PAY AOL OVER $5 MILLION

AOL is waiting to receive $5.6 million this month after winning its case against 25-year-old Minnesota spammer Christopher William Smith. Smith, whom *AOL* says it has been pursuing for three years, sent billions of spam messages via the company's email service and has now been ordered to pay $25,000 for every day he sent out spam emails, plus $287,059 to cover the company's legal fees.

Smith, who is reported to have been living it up in a $1.1 million house until his arrest, now resides in prison, awaiting an October trial in Minneapolis on federal charges that he operated an illegal online pharmacy during 2004 and 2005.

EVENTS

The 6th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will take place 28 February to 2 March 2006 in San Francisco, CA, USA. Members and non-members are welcome. Two further general meetings will also take place this year: 27–29 June 2006 in Brussels, Belgium, and 24–26 October 2006 in Boston, MA, USA. For details see http://www.maawg.org/.

The 2006 Spam Conference will be held 28 March 2006 at MIT, Cambridge, MA, USA. For details see http://www.spamconference.org/.

The Authentication Summit II takes place on 19 April 2006 in Chicago, IL, USA. The conference will cover the latest advances in email authentication, including Sender ID Framework (SIDF) and DomainKeys Identified Mail (DKIM), with a focus on real-life results and prescriptive information. For full details see http://emailauthentication.org/.

INBOX 2006 will be held 31 May to 1 June 2006 in San Jose, CA, USA. The event will cover all aspects of email including topics such as 'has CAN-SPAM failed us?', 'what can ISPs do to fix spam?', 'how not to be a spammer' and 'new directions in identifying spam'. For more information see http://www.inboxevent.com/2006/.

The third Conference on Email and Anti-Spam, CEAS 2006, will be held 27–28 July 2006 in Mountain View, CA, USA. The conference encompasses a broad range of issues relating to email and Internet communication. The conference format includes short and long presentations selected by peer review, as well as invited addresses. Those wishing to present long or short papers are invited to submit their proposals before 23 March 2006. Full details can be found at http://www.ceas.cc/.

The Text Retrieval Conference (TREC) 2006 will be held 14–17 November 2006 at NIST in Gaithersburg, MD, USA. More information about the TREC 2006 spam track can be found at: http://plg.uwaterloo.ca/~gvcormac/spam/.

FEATURE

DOES BAYESIAN POISONING EXIST?

John Graham-Cumming
The POPFile Project, France

A common criticism of statistical spam filters (such as *SpamBayes*, *DSPAM* and *POPFile*) is that they can be 'poisoned' by inserting random words into spam messages.

Postini, for example, makes the claim in one of its white papers [1] that the addition of legitimate-seeming words to spam messages can cause them to slip through spam filters. Worse, *Postini* claims, it can also cause false positives when users correct the spam filter's error by assigning a spammy probability to a good word.

Cloudmark's chief scientist Ved Prakash made the following claim in a news article [2]: 'The automata will just keep selecting random words from the legit dictionary ... When it reaches a Bayesian filtering system, [the filtering system] looks at these legitimate words and the probability that these words are associated with a spam message is really low. And the program will classify this as legitimate mail.'

Meanwhile, *Process Software* claims [3] that Bayesian poisoning has little effect: 'Even though the following tricks to poison Bayesian databases have little (if any) success, they still appear very frequently in spam messages.'

In December 2002, Paul Graham (who is arguably the father of Bayesian spam filtering) addressed the question of poisoning [4]. He said: 'To outweigh incriminating words, the spammers would need to dilute their emails with especially innocent words, i.e. those that are not merely neutral but occur disproportionately often in the user's legitimate email. But these words (the names of one's friends and family, terms one uses in one's work) are different for each recipient, and the spammers have no way of figuring out what they are.'

Confused? The opinions appear to depend on the nature of the product of those expressing them: *Postini* and *Cloudmark* sell non-Bayesian systems and claim that poisoning works; *Process Software*'s system and Paul Graham's system are both Bayesian and claim the opposite.

In this article I will review the published data on poisoning Bayesian spam filters to answer the question: does Bayesian poisoning exist? I will then show a novel way of poisoning a Bayesian spam filter and measure its effectiveness.

PUBLISHED RESULTS

At the Spam Conference held at MIT in 2004 I presented two possible attacks on *POPFile*'s Bayesian engine [5]. One was unsuccessful and the other worked, but was impractical. In doing this I identified two types of poisoning attack: passive (where words are added without any feedback to the spammer) and active (where the spammer gets feedback after the spam has been received).

The passive method of adding random words to a small spam was ineffective as a method of attack: only 0.04% of the modified spam messages were delivered. The active attack involved adding random words to a small spam and using a web bug to determine whether the spam was received. If it was, another Bayesian system was trained using the same poison words. After sending 10,000 spams to a single user I determined a small set of words that could be used to get a spam through.

Of course, the simple countermeasure of disabling remote images (web bugs) in emails eliminates this problem.

At the CEAS conference in 2004, Wittel and Wu presented a paper [6] in which they showed that the passive addition of random words to spam was ineffective against *CRM-114*, but effective against *SpamBayes* with 100 words added per spam.

They also showed that a smarter passive attack, adding common English words, was still ineffective against *CRM-114*, but was even more effective against *SpamBayes*. They needed to add only 50 words to a spam to get it past *SpamBayes*.

However, Wittel and Wu's testing has been criticized due to the minimal header information that was present in the emails they were using; most Bayesian spam filters make extensive use of header information and other message metadata in determining the likelihood that a message is spam. A discussion of the *SpamBayes* results and some counter evidence can be found in the *SpamBayes* mailing list archive: http://mail.python.org/pipermail/spambayes-dev/2004-September/thread.html#3065.

All of these attacks are what I refer to as type I attacks: attacks that attempt to get spam delivered. A type II attack attempts to cause false positives by turning previously innocent words into spammy words in the Bayesian database.

Also in 2004 Stern, Mason and Shepherd wrote a technical report at Dalhousie University [7], in which they detailed a passive type II attack. They added common English words to spam messages used for training and testing a spam filter.

In two tests they showed that these common words decreased the spam filter's precision (the percentage of messages classified as spam that really are spam) from 84% to 67% and from 94% to 84%. Examining their data shows that the poisoned filter was biased towards believing messages were more likely to be spam than ham, thus increasing the false positive rate.

They proposed two countermeasures: ignoring common words when performing classification, and smoothing probabilities based on the trustworthiness of a word. A word has a trustworthy probability if an attacker is unlikely to be able to guess whether it is part of an individual's vocabulary. Thus common words are untrustworthy and their probability would be smoothed to 0.5 (making them neutral).

At the 2005 CEAS conference Lowd and Meek presented a paper [8] in which they demonstrated that passive attacks adding random or common words to spam were ineffective against a naïve Bayesian filter. (In fact, they showed, as I demonstrated back in 2004, that adding random words improves the spam filtering accuracy.)

They demonstrated that adding 'hammy' words – words that are more likely to appear in ham than spam – was effective against a naïve Bayesian filter, and enabled spam to slip through. They went on to detail two active attacks (attacks that require feedback to the spammer) that were very effective against the spam filters. Of course, preventing any feedback to spammers (such as non-delivery reports, SMTP level errors or web bugs) defeats an active attack trivially.

They also showed that retraining the filter was effective at preventing all the attack types, even when the retraining data had been poisoned.

WHAT WE KNOW TODAY

The published research shows that adding random words to spam messages is ineffective as a form of attack, but that active attacks are very effective and that adding carefully chosen words can work in some cases. To defend against these attacks it is vital that no feedback is received by spammers and that statistical filters are retrained regularly.

The research also shows that continuing to investigate attacks on statistical filters is worthwhile. Working attacks have been demonstrated and countermeasures are required to ensure that statistical filters remain accurate.

A NEW APPROACH: ATTACKING THE SPAM PROBABILITY

All of the attacks described above either add innocent words to a spam message to get it past a Bayesian filter, or add words that will increase the filter's false positive rate when the user reports a delivered spam.

Another way to get spam messages past the filter is to reduce the probability of any spammy word. If it were possible to fill the Bayesian database with many unique spammy words, then the probability of any individual spammy word would be reduced, and when words are combined to score an individual message the spam score would be forced lower. The spammers' aim would be that the spam probability drops enough to cause spam messages to appear to be ham and consequently be delivered.

Bayesian spam filters typically calculate the spam probability for a word as either the number of messages in which the word appears divided by the total number of spam messages in the training set, or as the total number of times the word appears divided by the total number of words in the training set. By adding more unique spam words, each in individual messages, the spam probability for every spam word can be lowered in either case.

To test this theory I took a standard naïve Bayesian spam filter implementation written in C and trained it using spams and hams from the *SpamAssassin* public corpus (see http://spamassassin.apache.org/publiccorpus/). The filter was configured to run in train-on-everything mode (i.e. once the filter had determined the classification of a message it retrained itself automatically on the message, believing its own classification).

A separate corpus of 10,075 email messages drawn from my own mail (6,658 ham and 3,417 spam) was used to test the filter. Running through the messages in a consistent, yet random, order yielded a ham strike rate (also referred to as the false positive rate: the percentage of ham messages incorrectly identified) of 0.06% and spam hit rate (the percentage of spam messages correctly identified) of 99.80%.

To test the theory that delivering spam messages containing a single word could bias the filter's spam probabilities so that more spam would slip through, I randomly inserted spam messages containing a single randomly generated eight-letter word. The header was from a real spam message in the corpus (which caused the message always to be identified as spam and then trained on) and the body consisted of the eight-letter word.

The eight letters were chosen randomly from the alphabet to create a large number of unique words (a sample word might be HLAHEJGE). As the one-word spams were delivered, the unique eight-letter words would be added to the database (by the train-on-everything mode), thus lowering the probability for every spam word in the database.

I ran 11 tests: the first test had no one-word spam messages added and acted as a baseline for the effectiveness of the filter; the second test added 10 randomly generated one-word spam messages per spam message in the corpus (i.e. with 3,417 spams in the corpus, 34,170 one-word spams were inserted); the third test added 20 random one-word spams and so on up to 100 one-word spams per real spam message (for a total of 341,700 one-word spams).

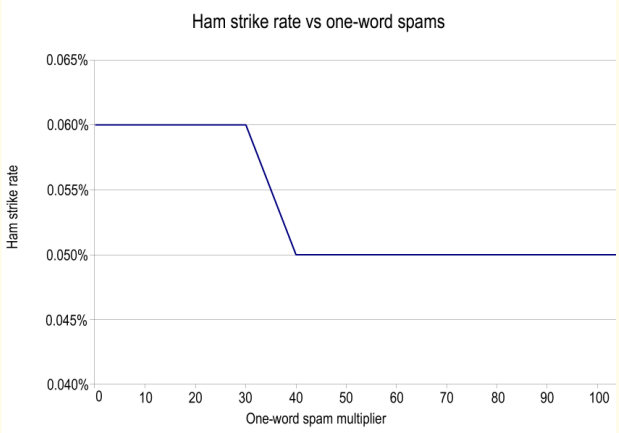


Figure 1: Ham strike rate vs one-word spams.

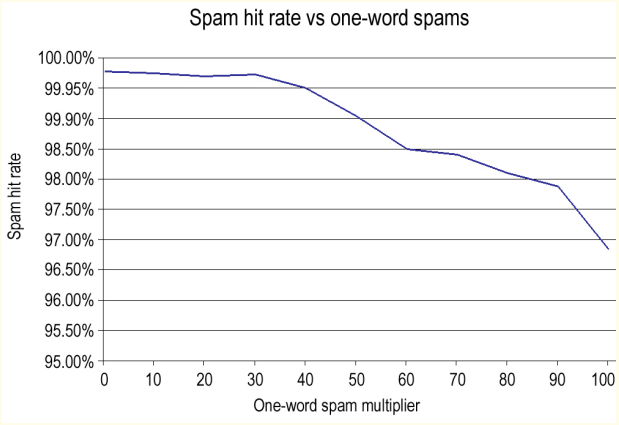


Figure 2: Spam hit rate vs one-word spams.

The existing spam and ham messages were run through the filter in the same order each time and the effectiveness of the filter measured without counting the additional one-word spams.

Figure 1 shows the effect of these one-word spams on the ham strike rate. With 40 one-word spams per real spam (for a total of 136,680 one word spams) the ham strike rate improves, indicating that the message that was previously incorrectly identified as spam has shifted (correctly) to be a ham. That's the first indication that the spam probabilities are lowering, causing the filter to bias towards messages being classified as ham.

Figure 2 shows the effect of the same one-word spam messages on the spam hit rate. The filter's accuracy at spotting spam deteriorates as the number of one-word spams is increased. With 347,100 one-word spams added the filter has dropped from 99.80% accurate at spotting spam to under 97%.

However, to put these figures into perspective, this attack required 341,700 additional spams to get 101 out of 3,417 spam messages delivered and at the same time the attack improved the filter's ability to identify ham messages correctly.

Although the attack is impractical, and easily defeated by purging rarely seen words from the database or by not using the train-on-everything methodology, it *does* work. And for spam filters that calculate the probability of a word based on the total word count (and not the total message count), the attack could be made more effective by including many random words in each message, thus requiring fewer messages to be delivered before having an effect.

CONCLUSION

The evidence suggests that Bayesian poisoning is real, but either impractical or defeatable. At the same time the number of published attack methods indicates that Bayesian poisoning should not be dismissed and that further research is needed to ensure that successful attacks and countermeasures are discovered before spammers discover the same ways around statistical spam filtering.

REFERENCES

[1] Postini white paper, 'The shifting tactics of spammers'. See http://www.spamwash.com/whitepapers/WP10-01-0406_Postini_Connections.pdf.

[2] 'Constant struggle: how spammers keep ahead of technology', *Messaging Pipeline*, 2005. See http://www.messagingpipeline.com/news/57702892.

[3] Process Software white paper, 'Common spammer tricks'. See http://www.process.com/techsupport/spamtricks.html.

[4] Graham, Paul; 'Will filters kill spam?'. See http://www.paulgraham.com/wfks.htm.

[5] Graham-Cumming, John; 'How to beat an adaptive spam filter', MIT Spam Conference 2004. See http://www.jgc.org/SpamConference011604.pps.

[6] Wittel, Greg and Wu, S. Felix; 'On attacking statistical spam filters', CEAS 2004. See http://www.ceas.cc/papers-2004/slides/170.pdf.

[7] Stern, Henry; Mason, Justin and Shepherd, Michael; 'A linguistics-based attack on personalised statistical email classifiers'. See http://www.cs.dal.ca/research/techreports/2004/CS-2004-06.shtml.

[8] Lowd, Daniel and Meek, Christopher; 'Good word attacks on statistical spam filters', CEAS 2005. See http://www.ceas.cc/papers-2005/125.pdf.