JUNE 2009

virus BULLETIN

Fighting malware and spam

CONTENTS

ISSN 1749-7027

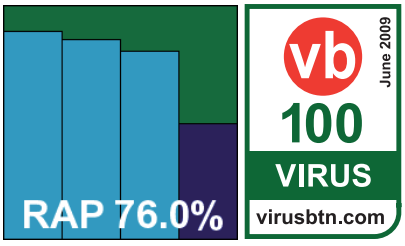
IN THIS ISSUE

WHERE'S WALEDAC?

Win32/Waledac is a trojan that is used to send spam. It also has the ability to download and execute arbitrary files, harvest email addresses from the local machine, perform denial of service attacks, proxy network traffic and sniff passwords. Scott Wu, Terry Zink and Scott Molenkamp take a detailed look at the spambot.

page S1

VB100 ON WINDOWS SERVER 2003



This month's comparative review tackles the 64-bit version of Windows Server 2003, with the platform bringing out quite a number of quirks and oddities in several of the products under test. John Hawes presents a round up of the results including the latest RAP testing data.

page 14

vbSpam supplement

This month: anti-spam news and events and a case study of the Waledac spambot.

'At the rate malware is currently released ... it may be that the specific naming of malware is a dead concept.'

Lysa Myers, West Coast Labs

MALWARE WITHOUT A NAME IS STILL MALWARE

Naming has always been a contentious subject in the anti-malware community, but at the rate malware is currently released, and with the volume of detection automated systems are now adding, it may be that the specific naming of malware is a dead concept.

The original idea behind standardized naming was to allow customers to determine whether a certain virus was detected by their anti-virus product. This was particularly important during the era when the mass media reported on virus outbreaks and AV companies received floods of inquiries about the virus *du jour*. Since the recent surge of financially motivated malware, customers have started to use virus names to find online descriptions so they can assess what damage may have been done.

As researchers and resources are taxed to the limit by this onslaught of malware, online descriptions have suffered. If there is a choice between using researchers' time to add detection or descriptions, it is arguably better to add detection. Remotely controlled and self-updating malware also make it more difficult to create descriptions: how do you create a static description of something which will have been updated by the time you finish writing it? The answer tends to be descriptions that are full of vague phrases such as 'behaviour differs depending on plug-ins installed' and 'differing versions have differing file sizes'.

The alternative for anti-malware vendors is to generate descriptions automatically. This allows more descriptions to be created with a basic level of information.

Automatically generated descriptions can easily detail the files that are added or modified and the network connections that are made by the malware. The downside is that an automated system cannot adapt to malware that requires more specific conditions, whereas a human can finesse a system into prompting additional malicious behaviour from a sample, and better imitate user behaviour.

Anti-malware vendors are already starting to move towards generic naming. A check of the top vendors' malware description sites shows malware names such as 'Troj/Agent', 'TROJ_SMALL', or just 'Generic Trojan'. This trend is likely to continue – if customers didn't complain when it began, they're unlikely to start now.

But the customer still wants to know what to do post infection to ensure their systems are completely cleaned, and what they can do to implement better protection in future. There are a number of options to address this, which boil down to either having someone or something which can forensically examine infected machines, or changing the nature of the 'cleaning' process.

There are many different network-monitoring technologies which provide information about network connections from infected machines. There are also services that examine infected machines forensically, or that offer a highly detailed analysis of captured malware. But, with the economic situation such as it is, it would be difficult to get customers to pay for new technologies or services when they perceive this as a service AV vendors already offer.

The other option is to change the nature of 'cleaning' to mean restoring a machine from a known-clean image or reformatting it entirely. This is certainly a drastic approach, but it is both quick and thorough.

A security representative for a local college used both options together: he would take a snapshot of the machine for forensic and possible data-recovery purposes, and then re-image the machine. He used this approach because he wanted to ensure there were no lingering traces of malware on his machines, and he found this to be the quickest way to get infected users back up and running, while providing detailed forensic data.

As the nature of anti-malware software changes, customers' expectations must be managed accordingly. The AV products of yore dealt with slow-moving threats, and researchers had time to fully examine and document them before they became widespread. Now threats come and go more quickly than any man or machine can adequately handle. Perhaps what is most needed now is a coalition to determine the AV industry's response to this change.

NEWS

OBAMA PLEDGES SECURITY EDUCATION FROM BOARDROOM TO CLASSROOM

US President Barack Obama has announced a new multi-billion-dollar effort designed to bolster the security of the United States' digital infrastructure. The plan involves the creation of a dedicated cybersecurity office at the White House and making cybersecurity 'a national security priority'.

In his speech, Obama referred not only to the need to secure the country's information and communication networks, but also to the 'millions of Americans [who] have been victimized, their privacy violated, their identities stolen, their lives upended, and their wallets emptied'.

Obama pledged to work with key players including state and local governments and the private sector to ensure an organized and unified response to cyber incidents in the future; to strengthen relationships between the public and private sectors and encourage collaboration with industry to find technology solutions; to invest in research and development and to begin a national education campaign to promote cybersecurity awareness and digital literacy 'from our boardrooms to our classrooms'.

With rumours rife of government agencies bringing in the use of 'magic lantern' trojans or 'bundestrojans' to gather intelligence on suspected criminals (see p.13), and corresponding levels of concern over the moral, legal and practical implications of such practices, Obama also made it clear that the stepping up of the country's cybersecurity efforts would not involve the monitoring of private sector networks or Internet traffic, saying: 'we will preserve and protect the personal privacy and civil liberties that we cherish as Americans'.

President Obama also promised an open and transparent process as the new cybersecurity strategy is developed – with the government's 60-day cyberspace policy review available to read from the official website of the White House.

BEWARE OF SEARCHING FOR LYRICS

According to a piece of research undertaken by *McAfee*, the word 'lyrics' is one of the riskiest terms to enter into a search engine. The study looked at more than 2,600 popular keywords and assessed the first five pages of results for each on five major search engines. As defined by *McAfee*, the search term 'lyrics' comes with a maximum risk factor of one in two. Meanwhile, nearly six out of the top 10 search results for 'screensavers' were found to contain malware, and results that contained the word 'free' were found to have a 21.3% chance being malicious. The full report can found at http://us.mcafee.com/en-us/local/docs/most_dangerous_searchterm_us.pdf.

Prevalence Table – April 2009

Malware	Type	%
Autorun	Worm	11.85%
NetSky	Worm	11.32%
Agent	Trojan	11.04%
Dropper-misc	Trojan	9.37%
Virut	Virus	8.02%
Invoice	Trojan	7.84%
Mytob	Worm	7.68%
Suspect packers	Misc	4.83%
OnlineGames	Trojan	4.02%
Mydoom	Worm	3.36%
Iframe	Exploit	3.08%
Basine	Trojan	1.92%
Zbot	Trojan	1.78%
PWS-misc	Trojan	1.08%
LDPinch	Trojan	1.00%
Bagle	Worm	1.00%
Delf	Trojan	1.00%
Zlob/Tibs	Trojan	0.94%
Zafi	Worm	0.87%
VB	Worm	0.72%
Heuristic/generic	Misc	0.60%
Sality	Virus	0.53%
QQPass	Trojan	0.51%
Mabezat	Virus	0.42%
Alman	Worm	0.37%
Backdoor-misc	Trojan	0.34%
Murlo	Trojan	0.31%
Downloader-misc	Trojan	0.29%
Cutwail/Pandex/Pushdo	Trojan	0.28%
Small	Trojan	0.26%
Tenga	Worm	0.23%
Inject	Trojan	0.19%
Brontok/Rontokbro	Worm	0.19%
Others[1]		2.79%
Total		100.00%

[1]Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

TECHNICAL FEATURE

ANTI-UNPACKER TRICKS – PART SEVEN

Peter Ferrie
Microsoft, USA

Unpackers have been around for as long as packers themselves, but anti-unpacking tricks have appeared more recently – and have increased rapidly both in number and, in some cases, complexity.

The final part of this series of articles (see also [1–6]) concentrates on anti-debugging tricks that target a number of popular debuggers, as well as some anti-emulating and anti-intercepting tricks.

All of the techniques described here were discovered and developed by the author.

1. ANTI-DEBUGGING TRICKS

1.1 Immunity Debugger-specific tricks

Immunity Debugger is essentially *OllyDbg* with a Python command-line interface. In fact, large parts of its code are identical, byte for byte, to the *OllyDbg* code. Consequently, it has the same vulnerabilities as *OllyDbg* with respect to both detection and exploitation [5, 6].

1.1.1 Malformed files

Like *OllyDbg*, *Immunity Debugger* does not properly support files whose entry point is zero. Zero is a legal starting value for EXE files and allows execution of the MZ header. Such files are still loaded in *Immunity Debugger*, but in each case the entry point's breakpoint is not set.

Immunity Debugger fails to check the values of the Export Address Table Entries field and the Base Relocation Directory Size field prior to performing some arithmetic on them. This can result in an integer overflow and memory corruption.

If the value of the Export Address Table Entries field is 0x40000000 or larger, then *Immunity Debugger* will start overwriting memory until a crash occurs.

If the value of the Base Relocation Directory Size field is 0x3FFFFFFE or larger, then *Immunity Debugger* will parse relocations from unallocated heap memory. On certain platforms, this can result in the execution of arbitrary code. The mitigating factor for the relocation table problem is the fact that it requires a file size of greater than one gigabyte, because *Immunity Debugger* reads the relocation data directly from the file.

The Export Address Table Entries and Base Relocation Directory Size bugs affect all versions of *Immunity Debugger*, including 1.70. The authors of *Immunity Debugger* released version 1.70 more than 60 days after the report was submitted to them. The authors have not responded to the report.

Despite being based on *OllyDbg*, only four of the *OllyDbg* anti-detection plug-ins have been ported to *Immunity Debugger*: *HideDebugger*, *HideOD, IsDebugPresent* and *PhantOm*. *IsDebugPresent* is a port of an earlier version, which only sets the debuggee's PEB->BeingDebugged to zero. The others are identical to the *OllyDbg* versions, and thus contain the same bugs [5, 6].

1.1.2 FindWindow

Immunity Debugger can be found by calling the user32 FindWindow() function, and then passing 'ID' as the class name to find.

Example code looks like this:

```
    push 0
    push offset l1
    call FindWindowA
    test eax, eax
    jne  being_debugged
    ...
l1: db "ID", 0
```

1.2 Zeta Debugger-specific tricks

Zeta Debugger is a lesser-known user-mode debugger with a graphical user interface. It supports plug-ins, but so far there are none that hide the presence of the debugger. Its code is very good and does not seem to have any obvious vulnerabilities. However, there is a bug that causes it to crash immediately on *Windows 2000*. The bug relates to the use of the kernel32 CreateToolhelp32Snapshot() function on a suspended process. This function was introduced to the *Windows NT*-line in *Windows 2000*, though it existed as far back as *Windows 95* in a separate DLL. On *Windows 2000* and later, it calls into the ntdll RtlQueryProcessDebugInformation() function, which performs the majority of the work. Part of that work includes inserting into the process a thread which gathers information about the process. This has the unintended consequence of resuming the process. Since the debugger has attached to the process, *Windows* also creates another thread that executes a breakpoint on behalf of the debugger. The problem is that when the process wakes up, the debug breakpoint will be executed before the debugger can call WaitForDebugEvent() to intercept it. Typically, the process would crash at this point. However, there are ways to continue execution and the process will not be under the debugger's control.

Windows XP and later attempt to read from the process memory first. This attempt fails for a suspended process because it has not been completely initialized at that time. As a result, *Windows XP* and later do not create a new thread, so they do not demonstrate the problem.

1.2.1 FindWindow

Zeta Debugger can be found by calling the user32 FindWindow() function, and then passing 'Zeta Debugger' as the class name to find.

Example code looks like this:

```
    push 0
    push offset l1
    call FindWindowA
    test eax, eax
    jne  being_debugged
    ...
l1: db "Zeta Debugger", 0
```

1.3 Rock Debugger-specific tricks

Rock Debugger is another less well known user-mode debugger with a graphical user interface. It supports plug-ins, but there are none that hide the presence of the debugger. It does not seem to have any obvious vulnerabilities.

1.3.1 FindWindow

Rock Debugger can be found by calling the user32 FindWindow() function, and then passing 'Rock Debugger' as the window name to find.

Example code looks like this:

```
    push offset l1
    push 0
    call FindWindowA
    test eax, eax
    jne  being_debugged
    ...
l1: db "Rock Debugger", 0
```

1.4 Turbo Debug32-specific tricks

Turbo Debug32 used to be a popular debugger for user-mode applications because of its familiar interface and solid performance. However, there are several problems in its code which leave it vulnerable to denial-of-service attacks, unexpected execution points, and even the execution of arbitrary code.

By far the biggest problem in *Turbo Debug32* is the fact that it makes multiple calls to strcpy() using stack buffers and user-defined copy sizes. These sizes are not checked before the copy is performed, thus it is possible for an attacker to crash the debugger, or potentially to execute arbitrary code.

Turbo Debug32 attempts to read the entire import table from the process in order to find and hook the kernel32 ExitProcess() function. It trusts the Import Table Directory Size field value, and uses it to allocate memory for the import table, regardless of the value that is specified. *Windows* uses the Import Table Directory Size field value as an upper bound value when parsing the import table, not as an allocation size for it. In the case of *Turbo Debug32*, if the size is large enough, the system performance will be impacted severely. Furthermore, since it is possible for the Import Table Directory Size field value to be smaller than the true size of the import table, *Turbo Debug32* might not read enough bytes to parse the import table correctly. As a result, the debugger might attempt to access out-of-bounds memory and crash.

When *Turbo Debug32* is asked to attach to a process that is already running, it assumes that advapi32.dll is already present in memory and available to the kernel32 GetModuleHandle() function. The correct behaviour would be to call the kernel32 LoadLibraryA() function. *Turbo Debug32* calls the kernel32 GetProcAddress() function to retrieve the addresses of two functions from advapi32.dll, and then calls them without checking if those addresses are non-zero.

When *Turbo Debug32* is asked to step over an instruction, it calculates the length of that instruction, then places a breakpoint at the location of the next instruction. However, the debugger calculates the instruction length for the 0xFF15 opcodes ('CALL' instruction, absolute indirect mode) incorrectly. The calculation code is copied directly from the 16-bit product, which checks for x6 for absolute addressing (where x6 represents an instruction encoding where 'x' is any hexadecimal value). However, this is only valid for 16-bit code; in 32-bit code, x5 is absolute addressing.

Turbo Debug32 also has no understanding of SIB mode. As a result, it writes a 0xCC opcode ('INT 3' instruction) at the wrong location. This causes a crash in most cases, but it can allow uncontrolled code execution if the new pointer is somewhat valid, and it could be manipulated by an attacker to produce this effect intentionally. It could also be used as a method to detect *Turbo Debug32*.

Example code looks like this:

```
l1: call d [offset l3]
l2: ...
l3: dd  offset l2
    db  0cch \
    -   (offset $-offset l1) dup (?)
l4: dd  offset being_debugged
```

By stepping over l1, a breakpoint will be placed inside the l1 instruction, instead of at the location of l2. The effect is

to change the 'call d [offset l3]' instruction into a 'call d [offset l4]' instruction.

1.5 Interactive DisAssembler (IDA)-specific tricks

Interactive DisAssembler, or *IDA*, is the most popular disassembler tool available today. It supports plug-ins.

IDA trusts the value of the Base Relocation Directory Size field, and uses it to allocate memory for the relocation table. However, the relocation table itself may specify a smaller size, because *Windows* uses the Base Relocation Directory Size field value as an upper bound value when parsing the relocation table, not as an allocation size for it. If the Base Relocation Directory Size field value is large enough, the system performance will be impacted severely, and *IDA* might exit unexpectedly.

Recent versions of *IDA* have been extended to include a user-mode debugger. The debugger is implemented as a plug-in for *IDA*. It has a couple of limitations.

The first limitation is during the debugging of files with a PE->ImageBase field value of zero. For such files, the *IDA* debugger will display a message that bears little resemblance to the actual problem. Once the file has loaded, all breakpoints are ignored and attempts to single-step will cause the debugger to resume execution without interruption. This technique has since been disclosed publicly [7].

The second limitation is during the debugging of files which contain multiple relocations pointing to the same memory location. *IDA* will not apply all of the relocation items, leading to an incorrect disassembly. There is no way of producing such a file automatically – manual intervention is required, for example by using a tool. The multiple relocation method can also be combined with the ImageBase zero trick. This combination of techniques is used by the Relock virus.

1.6 IDA plug-ins

A number of packers have been written to detect the *IDA* debugger, so the *IDA Stealth* plug-in was written to attempt to hide the debugger from them. The following is a description of the plug-in, with a list of vulnerabilities that could be used to detect it.

1.6.1 IDA Stealth

IDA Stealth sets the PEB->BeingDebugged and PEB->Heap->ForceFlags flags to zero, and clears all but the HEAP_GROWABLE flag in the PEB->Heap->Flags flags. It clears the FLG_HEAP_ENABLE_TAIL_CHECK, FLG_HEAP_ENABLE_FREE_CHECK and FLG_HEAP_VALIDATE_PARAMETERS bits in the PEB->NtGlobalFlag field. This behaviour is not as bad as setting bits arbitrarily, but it is still incorrect because the value in the PEB->NtGlobalFlag field can be set by a registry key and/or the debuggee [8].

IDA Stealth hooks the debuggee's ntdll NtQuerySystemInformation() function by replacing the first five bytes with a relative jump to an injected DLL. The hook intercepts attempts to call the ntdll NtQuerySystemInformation() function with the SystemKernelDebuggerInformation class. When that occurs, the hook checks if the SystemInformation parameter points to a valid memory address. However, it does not check whether the SystemInformationLength parameter contains a value that is large enough to hold the complete return value. As a result, if the length is too small, then *IDA Stealth* will cause an exception. The *IDA* debugger will trap the exception, but the debugging session will be interrupted.

If the parameters contain valid values, then *IDA Stealth* will store a value that corresponds to the KdDebuggerEnabled flag that has been cleared and the KdDebuggerNotPresent flag that has been set. However, due to an oversight by the author of the plug-in, the hook then calls the original ntdll NtQuerySystemInformation() function, and returns the true value. This fact was probably not noticed by the author of the plug-in because *IDA* is not a kernel-mode debugger, so unless a real kernel debugger was active at the time, the true value would match the fake one.

The hook also checks if the ntdll NtQuerySystemInformation() function has been called with the SystemProcessInformation class. If it has, then the hook calls the original ntdll NtQuerySystemInformation() function. If the call is successful, and the 'hide IDA' option is enabled, then the hook searches within the returned buffer for 'idag.exe' (the graphical version of *IDA*), and then erases all copies of the name that are found. The hook does not search for 'idaw.exe' (the console version of *IDA*), though.

If the 'fake parent' option is enabled, then the hook replaces the process ID of the *IDA* debugger with the process ID of EXPLORER.EXE in the InheritedFromUniqueProcessId field. This could be considered a bug, since the true parent might not be *Explorer*. The proper behaviour would be to use the process ID of *IDA*'s parent. Due to what appears to be another oversight by the author of *IDA Stealth*, this option is mutually exclusive with the 'hide IDA' option.

IDA Stealth hooks the debuggee's ntdll NtQueryInformationProcess() function by replacing

the first five bytes with a relative jump to an injected DLL. The hook intercepts attempts to call the ntdll NtQueryInformationProcess() function with the ProcessDebugPort class. When that occurs, the hook tries to return a zero for the debug port. However, there is a bug in this code, which is that the hook uses a hard-coded buffer length when it calls the original ntdll NtQueryInformationProcess() function. This can allow the function to succeed, even in cases where the ProcessInformationLength is invalid. As a result, passing an invalid length (longer than allowed) will result in a fixed return length (if the ReturnLength has been specified) and a zeroed port instead of an error code, and *IDA Stealth* is revealed.

The hook also checks if the ntdll NtQueryInformationProcess() function has been called with the ProcessBasicInformation class. If it has, then the hook assumes that the caller is requesting information about itself. The hook replaces the parent process ID with that of the shell window in the InheritedFromUniqueProcessId field, without first checking if the requested process ID is that of the current process. This behaviour is incorrect because the debuggee might be inquiring about a different process. This could also be considered a bug, since the true parent might not be the shell. The correct behaviour would be to use the process ID of *IDA*'s parent.

IDA Stealth hooks the debuggee's ntdll NtQueryObject() function by replacing its first five bytes with a relative jump to an injected DLL. The hook intercepts attempts to call the ntdll NtQueryObject() function with the ObjectAllTypesInformation class. When this occurs, the hook calls the original ntdll NtQueryObject() function, then searches within the returned buffer for the 'DebugObject' string. The hook sets the object counts to zero if the DebugObject string is found. There is a minor bug in the method of comparison, which is that it assumes that the name is zero-terminated. While this is currently the case for DebugObject, it is not a requirement, and there are already other objects with names that are not zero-terminated. The correct method would be to use the length field as the number of characters to compare. Of course, the length should be verified first, to avoid false success on substrings or superstrings, depending on which length is chosen for the comparison. A correct implementation is described in [8].

IDA Stealth hooks the debuggee's ntdll NtClose() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it registers a Structured Exception Handler before calling the original ntdll NtClose() function. The idea is to consume any exception that occurs. However, a debug event occurs in the debugger before the exception occurs in the debuggee, and that event cannot be prevented by the debuggee. The result is that *IDA* will always break by default. Furthermore, this method does not take into account that, in *Windows XP* and later, any Vectored Exception Handlers that the debuggee registers will run before the registered Structured Exception Handler in *IDA Stealth*. Thus, the presence of the debugger can still be detected on those platforms.

The plug-in hooks the debuggee's kernel32 OutputDebugStringA() and kernel32 OutputDebugStringW() functions by replacing the first five bytes of each with a relative jump to an injected DLL. When the hook is reached, it calls the original kernel32 OutputDebugString() function, and then sets the error code.

IDA Stealth tries to hook the debuggee's ntdll NtSetInformationThread() function by replacing its first five bytes with a relative jump to an injected DLL. The hook would intercept attempts to call the ntdll NtSetInformationThread() function with the HideThreadFromDebugger class, and if that were to occur, the hook would ignore the request and return successfully. However, there are two bugs in the code. The first is that the author of *IDA Stealth* mistyped the name of the function, so it is never hooked. The second is that if an invalid handle is passed to the function, an error code should be returned – a successful return would be an indication that the plug-in is running.

IDA Stealth hooks the debuggee's kernel32 SuspendThread() function by replacing the first five bytes with a relative jump to an injected DLL. When the hook is reached, it simply returns failure. This behaviour is a bug because no error code is returned if an invalid handle is specified.

IDA Stealth hooks the debuggee's kernel32 GetTickCount() function by replacing the first five bytes with a relative jump to an injected DLL. When the hook is reached, it returns a tick count that is incremented by a constant value each time it is called, regardless of how much time has passed. The value of the constant depends on the option that is enabled, and is either 1 or 966.

The plug-in hooks the debuggee's user32 BlockInput() function by replacing the first five bytes with a relative jump to an injected DLL. When the hook is reached, it simply returns successfully.

IDA Stealth hooks the debuggee's kernel32 OpenProcess() function by replacing its first five bytes with a relative jump to an injected DLL. When the hook is reached, it enumerates the list of processes in order to find the CSRSS.EXE process. If this is found, then its process ID is compared to the requested process ID. If there is a match,

then the hook returns an error code. Otherwise, it calls the original function.

IDA Stealth hooks the debuggee's user32 SwitchDesktop() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it simply returns success. This behaviour is a bug because no error code is returned if an invalid handle is specified.

The plug-in hooks the debugger's ntdll DbgUiConvertStateChangeStructure() function, if it is available, by replacing the first five bytes with a relative jump to the plug-in. When the hook is reached, it checks for the DBG_PRINTEXCEPTION_C (0x40010006) exception, and then simply returns success if it is seen. Otherwise, it calls the original function. This allows the exception to be delivered to the debuggee.

IDA Stealth hooks the debuggee's ntdll KiUserExceptionDispatcher() function by replacing the first five bytes with a relative jump to an injected DLL. When the hook is reached, it saves the values of the debug registers to a private memory block, and then clears them in the context structure, before passing the exception to the debuggee's exception handler. Upon return from the debuggee's exception handler, the hook restores the values of the debug registers, and then resumes execution.

IDA Stealth hooks the debuggee's kernel32 SetThreadContext() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it saves the contents of the debug registers to a private memory location. It clears the bit in the context structure that specifies that the debug registers are present, and then calls the original kernel32 SetThreadContext() function. The effect is to cache the requested changes to the debug registers, but to prevent those changes from occurring.

The plug-in hooks the debuggee's kernel32 GetThreadContext() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it calls the original kernel32 GetThreadContext() function, then merges the cached contents of the debug registers with the true values of the rest of the context. The effect is to simulate the requested changes to the debug registers.

The plug-in hooks the debuggee's ntdll NtYieldExecution() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it calls the original ntdll NtYieldExecution() function, then returns successfully.

IDA Stealth hooks the debuggee's user32 FindWindowA(), user32 FindWindowW(), user32 FindWindowExA() and user32 FindWindowExW() functions by replacing the first five bytes of each with a relative jump to an injected DLL. When the hook is reached, it checks if a class name has been specified. If it has not, then the window name will be used. In either case, the hook converts the name to lower case, and to Unicode if a hook was reached for an ANSI function. The hook then searches for the name within a list carried by the DLL. If a match is found, then the hook returns a failure. Otherwise it calls the original function.

The list of class names is as follows:

idawindow

tnavbox

idaview

tgrzoom

The list of window names is as follows:

ida

graph overview

idc scripts

disassembly

program segmentation

call stack

general registers

breakpoint

structure offsets

database notepad

threads

segment translation

imports

desktopform

function calls

structures

strings window

functions window

no signature

The problem is that the entire requested string is searched for each of the names in the list, which means that windows will be hidden if they contain words that include any of the strings in the lists. This mostly affects the 'ida' string. For example, a window with the title 'Acrobat Reader - [hidan.pdf]' [9] will not be visible.

IDA Stealth hooks the debuggee's user32 EnumWindows() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook

is reached, it replaces the callback function pointer on the stack with a new callback function pointer inside the DLL, and then calls the original function. When the new callback function is reached, it retrieves the window name and searches within the list of window names above. If a match is found, then the callback continues the enumeration. If no match is found it calls the original function.

IDA Stealth hooks the debuggee's ntdll NtTerminateThread() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it simply returns failure. This behaviour is a bug because no error code is returned if an invalid handle is specified. The same happens when the plug-in hooks the debuggee's ntdll NtTerminateProcess() function.

IDA Stealth hooks the debuggee's kernel32 GetVersion() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it returns a constant value that decodes to version 5.1.2600. This corresponds to *Windows XP*.

The plug-in hooks the debuggee's ntdll RtlGetVersion() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it checks if the RTL_OSVERSIONINFOW or RTL_OSVERSIONINFOEXW format has been requested. If the RTL_OSVERSIONINFOW format is requested, the hook returns version 5.1.1 with a platform ID that corresponds to *Windows 9x/Me* and a description of 'Service Pack 3'. This information contains two bugs. The first is that the build number is not '2600'. In fact, the correct build number is assigned, but to the wrong structure member. The second bug is that the platform ID does not correspond to a *Windows NT*-based platform.

If the RTL_OSVERSIONINFOEXW format is requested, the hook returns version 5.1.2600, with a platform ID that corresponds to a *Windows NT*-based platform, and a description of 'Service Pack 3'.

There is a bug in the code if neither format is requested, which is that no error code is returned.

The author of *IDA Stealth* responded to the report very quickly. The bugs were mostly fixed in beta 2. A number of new bugs were introduced in beta 2, but they were fixed in beta 3.

2. ANTI-UNPACKING BY ANTI-EMULATING

An emulator, as referred to within this paper, is a purely software-based environment, most commonly used by anti-malware software. It places the file to execute inside the environment and watches the execution for particular events of interest.

2.1 Software interrupts

2.1.1 Interrupt 4

When an EXCEPTION_INTEGER_OVERFLOW (0xC0000095) exception occurs, the EIP register has already been advanced to the next instruction, so *Windows* tries to rewind the EIP to point to the proper place. The problem is that *Windows* assumes that the exception is caused by a single-byte 'CE' opcode ('INTO' instruction). If the 'CD 04' opcode ('INT 4' instruction) is used to cause the exception, then the EIP will point to the wrong location. The same behaviour can be seen if any prefixes are placed before the 'INTO' instruction. An emulator that does not behave in the same way will be revealed instantly.

2.1.2 Interrupt 0x0D

When a general protection fault (interrupt 0x0D) occurs, *Windows* attempts to determine the cause of the fault in order to supply the appropriate exception code to the handler. The problem is that there are several ways to produce the general protection fault, which can result in very different exception codes.

For example, attempting to execute an instruction that contains too many prefixes yields EXCEPTION_ILLEGAL_INSTRUCTION (0xC000001D). The use of the HLT instruction, any of the descriptor table instructions and certain ports, yields EXCEPTION_PRIVILEGED_INSTRUCTION (0xC0000096). Other instructions and ports yield EXCEPTION_ACCESS_VIOLATION (0xC0000005). As described elsewhere [10], an instruction that contains the value 0xF0 within the first four bytes yields EXCEPTION_INVALID_LOCK_SEQUENCE (0xC000001E).

2.1.3 Interrupt 0x2C

In *Windows NT*, interrupt 0x2C formed one half of an event pair with interrupt 0x2B. A client and a server each controlled one half of the pair, with the server using interrupt 0x2B to pass information to the client, and the client using interrupt 0x2C to pass information to the server.

That functionality was removed in *Windows 2000*. Instead, in *Windows 2000* and *Windows XP*, interrupt 0x2B is the user-mode callback interface for user32.dll, and interrupt 0x2C returns the EXCEPTION_NO_EVENT_PAIR (0xC000014E) in the EAX register. That functionality was changed again in *Windows Server 2003*. Now, in *Windows Server 2003* and *Windows Vista*, interrupt 0x2C is the

DbgRaiseAssertionFailure() macro, and when it is executed *Windows* issues an EXCEPTION_ASSERTION_FAILURE (0xC0000420) via an exception that can be intercepted.

2.2 File-format tricks

Normally, a PE file requires a non-zero section count and corresponding section descriptors to lay out the file in memory. However, as noted in [8], it is possible to have no section table in the file. As a result, it is also possible to specify explicitly that the file contains no sections. That is, to set the PE->NumberOfSections field to zero. Following such a change, it becomes possible to completely remove the section table on all *Windows NT*-based platforms, including *Windows Vista*. As a result of removing the section table, many tools decide that the file is corrupted and not worthy of examination.

3. ANTI-UNPACKING BY ANTI-INTERCEPTING

3.1 W^X interception

Finally, some unpacking tools work by changing the previously writable-executable page attributes to either writable or executable, but not both. These changes can be detected by using timing attacks, such as a timer query around a local memory write.

Example code looks like this:

```
rdtsc
mov  ebx, edx
xchg ecx, eax
;hidden page fault because page is not writable
mov  b [offset $], 8bh
rdtsc
sub  eax, ecx
sbb  edx, ebx
jne  being_debugged
cmp  eax, 500h
jnbe being_debugged
```

In the example code, the assumption is that the code section is both executable and writable. This is tested by querying a timer (RDTSC), saving the result, attempting to write to the code section, then querying the timer again. In a normal environment, the difference between the two timer values would be small. However, in a W^X environment, the write will cause a page fault because the page attributes have been changed to read-only. The servicing of the page fault will take a long time, and so the difference between the timer values will be large.

CONCLUDING REMARKS

As noted throughout this series, new anti-unpacking techniques continue to be developed as the older ones are constantly being defeated. This series of articles has focused on some of the tricks that might become common in the future, along with some countermeasures.

The text of this article was produced without reference to any *Microsoft* source code or personnel.

REFERENCES

[1] Ferrie, P. Anti-unpacker tricks – part one. Virus Bulletin, December 2008, p.4. http://www.virusbtn.com/pdf/magazine/2008/200812.pdf.

[2] Ferrie, P. Anti-unpacker tricks – part two. Virus Bulletin, January 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200901.pdf.

[3] Ferrie, P. Anti-unpacker tricks – part three. Virus Bulletin, February 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200902.pdf.

[4] Ferrie, P. Anti-unpacker tricks – part four. Virus Bulletin, March 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200903.pdf.

[5] Ferrie, P. Anti-unpacker tricks – part five. Virus Bulletin, April 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200904.pdf.

[6] Ferrie, P. Anti-unpacker tricks – part six. Virus Bulletin, May 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200905.pdf.

[7] Souriz's weblog. #773: bug in IDA-Pro [fails to debug zero-based PE]. http://souriz.wordpress.com/2008/05/14/773-bug-in-ida-pro-fails-to-debug-zero-base-pe/.

[8] Ferrie, P. Anti-unpacker tricks. http://pferrie.tripod.com/papers/unpackers.pdf.

[9] Ferrie, P. Hidan and dangerous. Virus Bulletin, March 2007, p.4. http://www.virusbtn.com/pdf/magazine/2007/200703.pdf.

[10] Ferrie, P. Locked and loaded. http://pferrie.tripod.com/misc/lowlevel1.htm.

CONFERENCE REPORT 1

CARO MIO, AMTSO MON AMOUR

David Harley
ESET

A researcher's lot is not an easy one, with frequent treks to be made both virtually and in reality across time zones in an attempt to keep up with current threat and research trends. Sometimes, though, one comes across a conference or workshop where a happy combination of social networking, the exchange of solid information, great entertainment and a beautiful setting makes it all worthwhile. Last month I was fortunate enough to attend two such events held consecutively in Budapest: the annual CARO workshop, and the most recent AMTSO (Anti-Malware Testing Standards Organization) meeting.

CARO

This year's CARO workshop was focused on the theme of exploits and vulnerabilities. The agenda displayed at http://www.caro2009.com/ gives some idea of the range of sub-topics covered, and this report will cover a few of the highlights. As in previous years, it is likely that some of the presentations will be made available on the website, though the scope and nature of the workshop was such that some of the material may not be released publicly. While it is not appropriate for me to go into detail about technical issues that presenters may not wish to be made public, I hope it is acceptable for me to record some personal impressions of this lively event.

If anyone was going to have problems with a reluctant laptop-to-projector, it was probably just as well that it was Righard Zwienenberg: in his keynote presentation he rose above the problems to give a typically entertaining, yet thought-provoking talk. His description of a call centre conversation about anti-skimming measures was a perfect illustration of a very common problem in security: the culture clash.

The keynote was followed by a consideration of recent vulnerabilities in *Adobe* products (especially *Acrobat/Reader* and *Flash*), an issue to which I've also been paying much attention in recent months. It may not be altogether fair to lay too much emphasis on the sheer number of such issues, but I was slightly shocked to see how many CVEs *Adobe* has notched up in 2008–2009. I would certainly hope in future to see a more coherent and proactive approach to security problems from *Adobe* than I think is currently the case.

Taking a very different angle, the next presentation considered the impact of zero-day vulnerabilities on vendor stock market prices. To my surprise, I found this fascinating, and I will certainly be checking out some of the other research in this area that was cited by the presenter, Anthony Bettini.

The MS08-067 vulnerability is usually associated with Conficker, but it is useful to remember that the Conficker gang is not the only one skinning that particular cat, so Pierre-Marc Bureau followed the road less hyped as well as the established Conficker time line. However, as you might expect, the Conficker connection turned up on several occasions during the course of the workshop.

The afternoon's presentations, including Maksym Schipka's paper on *Office* exploits, maintained the high standards that had been set in the morning, but perhaps the show stopper was Peter Ször's 'Attacking the Cloud', a broadly based consideration of some potential weaknesses in cloud-based anti-malware technology. Controversially, some of the points he made referred to products that are already working in that space, provoking some lively discussion the following day.

For that evening we were all spirited away – that is, transported by bus – to an equestrian display, followed by an excellent dinner.

The next morning, Andreas Marx started off proceedings with a paper entitled 'Testing exploit-prevention mechanisms in anti-malware products'. The presentation drew comparisons with other pain points in the need for new approaches to anti-malware testing and set the tone for (or at least prefigured) the AMTSO meeting that was to follow the next day. Other presentations in the morning looked at PE and other vulnerable formats (AutoIt executables, NSIS installers and SWF files), plus a more specific look at Conficker in the context of vulnerability analysis. Abhijit Kulkarni and Prakash Jagdale followed up on work they had presented at AVAR last year on vulnerabilities in anti-malware scanners executing in 64-bit environments, and Ziv Mador presented a view of the current exploit landscape from *Microsoft*. In the final sessions, Roel Schouwenberg shared some juicy data and Nick FitzGerald talked about web exploit kits and their evolution, bringing to an end a typically exhausting but unmissable two-day brain-dumping session.

AMTSO

The following morning it was back to the same room for a rather different event, though with a considerable overlap in attendees. The AMTSO workshop was very much focused on organizational administration issues and

forthcoming deliverables, and it never ceases to amaze me that such an aggregation of strong-minded individuals are able to reach consensus on so many topics so (relatively) quickly. (I guess that not every horse designed by a committee is a camel.) Again, I should emphasize that these are very much personal impressions.

After a summary of the organization's recent activities, introducing such issues as the recently overhauled website at http://www.amtso.org/ (it looks very good, but anyone with links to the documents hosted there might want to check that they still work), three more documents were discussed exhaustively and eventually accepted in principle by the membership:

- A document outlining the process for dealing with requests for review analyses. This establishes the mechanism by which interested parties can request an analysis of tests and reviews based on how closely they conform to AMTSO guidelines (see the 'Fundamental Principles of Testing' document at http://www.amtso.org/documents.html). I imagine that many will see this as a critical aspect of AMTSO's activities in the near future, and an essential step towards establishing compliance with AMTSO's principles as a 'must-have' for credible testing.
- A document outlining issues with and best practices for the testing of security products that use some form of 'in-the-cloud' distributed processing. Like dynamic testing, I expect this to be a growth area in comparative testing: it will be difficult and resource-intensive for testers to implement these approaches properly, but this document will offer solid guidance on evolving techniques that they will need to address sooner rather than later.
- A document suggesting methods by which samples can be validated. Again, I see this as a topic of crucial importance in testing: inadequate validation has undermined the viability of test after test over the years, and I regard it as one of the major issues that a testing standards body needs to address.

Work also continues on a glossary (yet another vital project, in my view) and on some other papers that are not yet ready for final approval, addressing topics such as sample generation (I can hear you groaning from here) and testing methods that take fully into account the holistic detection abilities of a product that so often get lost in a simple static test. Work has started on some new documentation.

Special thanks and congratulations to Gabor Szappanos and his colleagues for setting everything up for both events, and for looking after us all so well.

CONFERENCE REPORT 2

EICAR 2009 IN A NUTSHELL: ICH BIN EIN EICARER

Eddy Willems
Kaspersky Lab and EICAR, Belgium

The 18th EICAR conference took place last month in Berlin. Situated close to the fabulous Kurfürstendamm shopping street, as well as the famous Gedächtniskirche church, the Steigenberger hotel provided an ideal setting for the conference and the sun shone throughout the week.

The pre-conference programme, which ran for two days prior to the start of the conference, featured a number of workshops including an interesting tutorial about JavaScript and VBScript malware analysis, and a session on the theoretical and practical implications of supervised automation of malware variant generation. A live memory forensics tutorial also proved to be worth the visit.

The real meat of the conference itself began with an opening word from the chairman of EICAR, Rainer Fahs, followed by a keynote address from Professor Dr Fred Cohen. The professor is widely acknowledged as having been the first person to define the term 'computer virus', having included the definition in his 1984 thesis. He is also the author of the Deception Toolkit – well known today in the UNIX/*Linux* world. Prof. Cohen's speech – which was an absolute highlight – gave a nice indication of the differences between commercial and academic views of the malware problem. He concluded that viral computing is here to stay, and that we have to live with it, but that we really must put thought and effort into defending 'our' cyberspace and the very vulnerable infrastructure behind it.

After Prof. Cohen's speech, Ronald Schulze from *BDK* described a project called Webpatrol – an interesting approach to handling Internet emergencies by using feedback forms filled in by ordinary users. Boris Sharov from *Dr. Web* continued the morning's presentations with an excellent overview of some newly detected malware. After this, the conference split into two tracks with a mixture of industry and academic papers – which makes this conference quite unique these days. As always, it was hard to decide which stream to follow.

First, I attended a presentation by Magnus Kalkuhl from *Kaspersky Lab*'s Global Research Team, who described some of the undesirable situations that could potentially arise in the next 10 years. The more people depend on computers and robotics, the stronger the impact that malware will have on their lives – not only in financial terms, but with serious consequences for victims' lives. Magnus looked at some of the ways in which the risk could be reduced, which seemed a bit utopian at first, but that

might have been due to their futuristic nature. This was a real science fiction thriller.

And there was more to come: Babu Nath Giri from *McAfee* presented a paper entitled 'Malware in men'. By combining materials from two studies he demonstrated that implantable medical devices are vulnerable to malicious attacks. He discussed the possibility of such malware arising in the future. I must confess that, after hearing what Babu had to say, I would think twice before having a bionic eye or a hearing aid implant!

Another enjoyable presentation from *McAfee* (by Ramagopal Prashanth, Mohandas Rahul and Thomas Vinoo) was about the rise of autorun-based malware. The paper looked at advancements in this type of malware. Thomas discussed methods that can be used proactively to detect and stop malware that spreads via removable drives, using a combination of traditional anti-virus and cloud computing techniques. Later, Michael Friela's presentation detailing his risk behaviour index gave an insight as to how the use of psychology in the context of security could help create awareness by changing human behaviour. Such a feat is easier said than done, and I have my doubts about its viability, but remain open minded.

That evening, the conference gala dinner provided an opportunity to relax and enjoy a real treat: magician Didi Saxer put on a perfect show with a brilliant mixture of comedy and magic.

The following morning, Professor Dr Nikolaus Forgo presented an overview of the current status of and recent developments in European legislation on data protection and data security. Of course, Prof. Forgo's presentation touched on the topic of the possible German 'BundesTrojan' and the issues that it raises for the security industry. EICAR will continue to monitor legal developments in Europe as they become increasingly important.

For the first time in the history of the EICAR conference, the best paper prize was awarded to an industry paper which brilliantly combined elegant theory with practical applications in critical fields: Sébastien Tricaud and Philippe Saadé's 'Applied parallel coordinates for logs and network traffic analysis'. If you are mathematically minded this paper is a must-read.

One of the specific areas this EICAR conference focused on was anti-malware testing. David Harley and Randy Abrams from *ESET* presented a paper on 'Execution context in anti-malware testing'. They reviewed the most common mainstream anti-malware detection techniques and tried to clarify the terminology most commonly used in this context in relation to the technology it describes. Hopefully the attempts by AMTSO to establish testing standards, and anticipated parallel initiatives from EICAR, will start to break down psychosocial barriers to the popular acceptance of the need for more rigorous testing practices.

Other papers on the subject of testing included an empirical evaluation of whether behavioural anti-virus products are able to detect complex metamorphic malware (Jean-Marie Borello, Ludovic Mé and Eric Filiol from ESIEA); a paper entitled 'Applied evaluation methodology for AV software' (Alexandre Gazet and Jean-Baptiste Bédrune from Sogeti/ESEC); and a study of 'anti-virus response to unknown threats' (Christophe Devine and Nicolas Richaud from *Thales Security Systems*), which gave some insight into problems relating to anti-malware products. My advice to some of the authors is to take a deeper look at the AMTSO documents – however, from a theoretical point of view, the papers were quite interesting.

Andrew Hayter from *ICSA Labs* looked at how the accreditation of testing and certification programmes under the ISO 9001 and 17025 standards could provide assurance both to the anti-malware developers and to the endpoint consumer that test labs meet the rigorous standards set by the International Standards Organization. Meanwhile, Ferenc Leitold from *Veszprog* described a unique and closed testing and certification procedure that could be used for dynamic testing.

A good part of both the commercial and academic anti-malware worlds were represented in a panel session about anti-malware testing, which was another highlight of the conference. This session continued to provide a deeper look at and better understanding of the principles of testing and the complexity of the issue. It was agreed that we need recognized testing standards and some independent body(ies) to regulate testing, all for the benefit of the user. This is also the approach of the AMTSO initiative. In determining these standards and regulations we should include as many organizations, vendors, academics and testing bodies as possible, but we must not forget also to include the end-users.

By the time you read this, or soon after, most of the presentations from this year's conference (including those I have been unable to include in this summary) will be available on the EICAR conference website (http://www.eicar.org/). This year saw a significant increase in both the quality and quantity of papers submitted for the conference, and the event itself was a great success.

The 19th EICAR is due to take place next year in France, at the ESAT facilities (Ecole Supérieure et d' Application des Transmissions) in the heart of Paris from 8 to 11 May 2010. A call for papers as well as more detailed information will be published soon. Mark the dates in your diaries!

COMPARATIVE REVIEW

VB100 ON WINDOWS 2003 SERVER X64

John Hawes

This month's comparative review tackles the 64-bit version of *Windows Server 2003*. Although superseded by *Server 2008* last year, the platform remains the standard server OS in many *Windows* environments, and as such it should be well provided for by anti-malware solutions.

The platform presents a number of issues for developers to overcome, not least the 64-bit environment, whose unexpected quirks and oddities seemed certain to show up in the performance of a few products – especially those not specifically built for the environment. Several potential pitfalls presented by the WOW64 system were highlighted at a recent conference on vulnerabilities, where researchers documented the possibility for numerous products to be deceived by the doctored responses returned by the set-up. Many other issues, particularly with built-in emulation, also seemed likely to crop up.

A slightly larger than anticipated field of competitors entered the fray this month, despite a couple of unexpected absentees. A total of 22 products made the final list, many of them dedicated server products but with a fair share of standard desktop editions as well. A single newcomer bravely took its first stand against the VB100 system on this tough platform, with most of the other entrants familiar through long histories in our tests.

PLATFORM AND TEST SETS

Initial set-up and configuration of the operating system is not too complex or demanding a task, particularly as our requirements were for little more than a basic fileserver system – the main aim of our test is to measure the abilities of the products to protect both the local system and other systems accessing files stored on it, and the more complex side of server administration – running web, mail and database servers and so on – was outside of our remit. Beyond installing the OS, overlaying the latest service pack and applying some network drivers required to activate the network cards, little additional manipulation was required to get the systems set up to our liking.

With snapshots of the test systems taken, test sets were copied to shares on each machine. This month's test set deadline was 17 April – rather earlier than usual to accommodate the new RAP set-up and a slew of important conferences taking place around the start of May, and this unfortunately meant missing the release of an updated WildList by a matter of days. As usual, we went with the

most up-to-date list available at the time of the deadline: the February 2009 list, released in late March. This meant that there would have been plenty of time for labs to ensure full coverage, and it also meant that only fairly minor changes to our core certification set needed to be made. Additions consisted mainly of the standard autorun worms and online gaming trojans which have been dominating the list for some time, with a sprinkling of new W32/Conficker variants as the main item of interest. One of the most highly anticipated additions, a new strain of the complex W32/Virut polymorphic file infector, did not quite make it onto this list – making its debut in the March list (so likely to appear in our WildList set in the next VB100 review) – but as samples were rife in our feeds in the month prior to the test we were able to include a large batch in our polymorphic set.

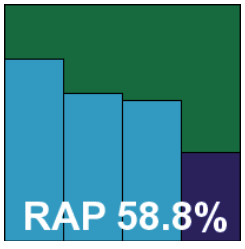
The size of this batch was considerably enhanced by an automated virus replication tool which has been under development in the lab for some time. After having reached a reliable state, the tool has been churning out large numbers of new samples throughout the last few months. This has enabled us to refresh and enlarge several of our polymorphic test sets, with several of the more virulent W32/Virut strains now represented by several thousand samples. With the latest strain well represented here, we were promised some insight into how well labs have dealt with this tricky, highly prevalent and now officially in-the-wild threat.

Elsewhere, the RAP and trojan test sets were made up of recent items arriving from our various sample sources, with the RAP samples gathered in the three weeks prior to the product deadline and the week after it, and the trojan set built from items appearing in the month or so prior to that. We had hoped to find time to rebuild and refresh our set of worms and bots, and did put together a semi-validated set for this purpose, but regrettably we were unable to perform the necessary steps to complete the integration; the VB100 review on *Vista* (due for publication in August) should see this set stocked with fresh items from the same period as the trojans set.

The clean set saw a fairly standard-sized update, with the bulk of new additions consisting of drivers and firmware for network devices and tools. With everything ready, all systems matching and sets synchronized, we got down to finding out how the products would fare.

Agnitum Outpost Security Suite Pro 2009 6.5.4.2525.381.0687

ItW	100.00%	**Polymorphic**	88.58%
ItW (o/a)	100.00%	**Trojans**	80.78%
Worms & bots	99.91%	**False positives**	0

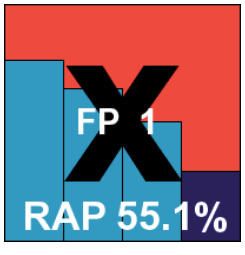


Agnitum's *Outpost* suite is essentially a desktop product, but should provide ample protection for a server platform. The installation process includes options for the numerous components, including web and spam filtering and behavioural monitoring as well as the famed client firewall. Installation took quite some time thanks to various network scans, attempts to update (foiled of course by the isolated nature of our lab), and finished with a recommendation to reboot to ensure full efficacy.

Once ready to use, the interface impressed the lab team with its simple, uncluttered layout, but it seemed somewhat lacking in the fine-tuning options likely to be required by most server admins to ensure best fit with their specific requirements. Scanning speeds were no more than fairly good, and on-access lags were somewhat above average, but a caching system should provide better speeds once the product has familiarized itself fully with its environment (something which we hope to be able to test more accurately in the near future). Our tests didn't cover the behavioural and other aspects of protection provided, but the detection rates recorded represent a fair measure of the product's ability to protect fileshares from infiltration. These rates proved fairly decent in general, with a steady decline in detection of the RAP sets as time to product freezing drew closer, as expected. In the polymorphic set, a fair number of samples of the latest Virut variant were missed, suggesting that some more work may be needed to make the grade next time around, but with no issues in the current WildList set, no false positives and no other problems, *Agnitum* starts this month's comparative off well by winning a VB100 award.

AhnLab V3NET for Windows Servers 7.0.2.2

ItW	100.00%	**Polymorphic**	98.92%
ItW (o/a)	100.00%	**Trojans**	75.38%
Worms & bots	99.86%	**False positives**	1



AhnLab's dedicated server product proved much simpler to install, with the option of a pre-install scan to ensure the system is clean before getting under way. The install offers an optional 'anti-hacking' feature alongside the standard choices, and is up and running with no reboot required. The interface,

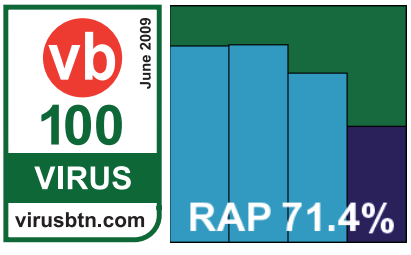
closely mirroring the desktop product, is clean and simple with most of the basics easy to find, but once again, the more in-depth configuration which seems appropriate for server products was absent.

Speeds were somewhat slow on demand, but not bad at all on access. Detection rates also seemed fairly impressive, with again a pretty steady decline across the RAP sets as expected, and mediocre coverage of the latest Virut variant.

In the WildList set, things seemed fine on demand but less so on access, where a small selection of samples were not blocked immediately. Probing this issue, it seemed that the product continues the somewhat outmoded path of separating 'virus' and 'spyware' detection, to the extent of requiring separate filesystem scans to check for each type of malware. Both types of detection are active on access, and some WildList samples were being detected by the anti-spyware portion of the product. Despite appearing to be configured to deny access on detection, the anti-spyware module seemed not to do this as well as the anti-virus module, which was blocked from scanning the files as they had already been alerted on by the anti-spyware component. Although this seems like a rather nasty situation, logging of detection is all that the VB100 rules demand and thus the product is credited with full coverage of the WildList. In the clean sets, a false positive emerged on a fairly obscure browser product, relieving us of the pressure of making a tricky call on the WildList behaviour, and *AhnLab* does not quite make the grade for a VB100 award.

Alwil avast! Server Edition 4.8.1087

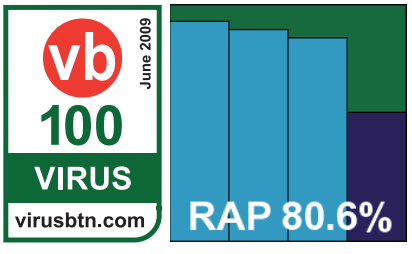
ItW	100.00%	**Polymorphic**	99.22%
ItW (o/a)	100.00%	**Trojans**	94.20%
Worms & bots	99.91%	**False positives**	0



Alwil's server version provides a speedy and straightforward installation process, at the end of which a reboot is not forced, but those choosing not to do so are warned that 'system failure' may result. The interface closely resembles the desktop edition, with the advanced version providing a wide range of controls and options but proving rather cluttered and difficult to navigate. While a simple version is also available, the default settings provided are fairly basic and likely to be inadequate for most admins. Scanning speeds were fairly mid-range, and detection rates a fraction below the outstanding levels expected – but were nevertheless impressive, with no problems having been encountered in detecting the large numbers of new W32/Virut samples in the polymorphic set. The WildList was likewise covered cleanly, and with the clean set presenting no serious problems either, a VB100 is awarded to *Alwil*.

AVG Internet Security Network Edition 8.5.322

ItW	100.00%	**Polymorphic**	98.96%
ItW (o/a)	100.00%	**Trojans**	96.15%
Worms & bots	99.95%	**False positives**	0



AVG's product again provides a slick and fast install, with no reboot necessary, and a 'first run wizard' provides configuration for things like updating, scheduled scans, trusted networks and so on. The interface seems identical to the standard desktop version – rather busy, with icons for numerous components and modules leading to more advanced configuration in tree format, which can also become a little tricky to navigate in its rather small default window. Some options that would be of relevance to server admins, such as processing of archive files on access, seemed to be absent, but could merely have been overlooked in the confusion.

Speeds were in the medium range, and detection rates continued their recent upward climb, with once again no problems with either the WildList or the new Virut strain expected to join it next time around. The clean sets were ably handled too, and *AVG*'s superb performance earns a VB100 award.

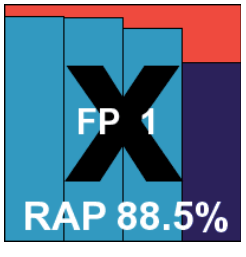
Avira AntiVir Server 9.00.00.23

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	96.98%
Worms & bots	100.00%	**False positives**	1

Avira also impressed with the speed of its installation process, despite the need to set up some Visual C++ components on the system, and again no reboot was required. This is a proper server edition, with an MMC-based console to control configuration – which appeared to be provided in considerable depth. The neatly laid out tree structure proved simple to navigate and easy to use, and overall the design was declared excellent by the lab

On-access tests	WildList viruses		Worms & bots		Polymorphic viruses		Trojans		Clean sets	
	Missed	%	Missed	%	Missed	%	Missed	%	FP	Susp.
Agnitum Outpost	0	100.00%	2	99.91%	442	88.58%	1931	77.34%	0	0
AhnLab V3Net	0	100.00%	3	99.86%	246	98.92%	2251	72.47%	1	0
Alwil avast!	0	100.00%	2	99.91%	13	99.22%	538	93.92%	0	0
AVG Internet Security	0	100.00%	1	99.95%	21	98.96%	394	94.56%	0	0
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	241	96.86%	1	1
BitDefender Security	0	100.00%	0	100.00%	0	100.00%	911	87.62%	1	0
CA eTrust	0	100.00%	0	100.00%	1049	92.03%	6743	32.03%	0	0
ESET NOD32	0	100.00%	0	100.00%	0	100.00%	1405	86.70%	0	0
Fortinet FortiClient	0	100.00%	0	100.00%	202	99.15%	8872	5.66%	0	0
Frisk F-PROT	0	100.00%	0	100.00%	165	98.93%	2420	67.74%	1	0
F-Secure Anti-Virus	0	100.00%	0	100.00%	1	100.00%	3184	75.76%	0	2
K7 Total Security	0	100.00%	134	93.72%	760	86.09%	4265	61.82%	0	0
Kaspersky Anti-Virus	0	100.00%	0	100.00%	1	100.00%	3226	74.65%	0	0
McAfee VirusScan	0	100.00%	0	100.00%	1	100.00%	903	87.95%	0	0
MWTI eScan	0	100.00%	6	99.72%	0	100.00%	837	88.71%	0	0
Netgate Spy Emergency	143	69.96%	484	77.33%	9963	1.77%	8163	14.61%	13	0
Norman Virus Control	0	100.00%	0	100.00%	726	81.34%	2677	70.95%	0	0
Quick Heal Anti-Virus	0	100.00%	8	99.63%	178	95.69%	2738	68.28%	1	0
Sophos Anti-Virus	0	100.00%	0	100.00%	4	99.97%	857	88.88%	0	6
Symantec Endpoint Protection	0	100.00%	0	100.00%	1	100.00%	478	93.59%	0	0
TrustPort Antivirus	0	100.00%	0	100.00%	131	98.82%	1441	88.19%	0	0
VirusBuster Professional	0	100.00%	2	99.91%	442	88.58%	2044	78.29%	0	0

team, although the default settings on access were once again fairly basic. Running through the test quickly and easily, we noted that on-demand speeds, normally extremely fast, were not as far ahead of the pack as usual, although on-access overheads were as excellent as ever.



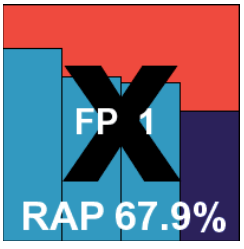
Detection rates were similarly superb across the board, with some truly remarkable figures in the RAP sets and no problems handling the expanded polymorphic sets. Sadly, however, a single false alert on a fairly minor item in the clean sets scuppered *Avira*'s hopes of earning a VB100 this month.

BitDefender Security for Windows File Servers 3.1.70

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	88.36%
Worms & bots	100.00%	**False positives**	1

Another full server edition, *BitDefender* offers admins control over the number of scanning processes implemented, and during installation does some probing to estimate an optimal default level. Also included with the otherwise fairly standard install process is a request for permission to send crash information back to the developers to smooth out any wrinkles in the product's stability, and at the end a reboot is required to finalize the

install. The interface again uses the MMC system and a tree of configuration and option controls, which the team found clear and well laid out. It also provides lots of statistical information on its own performance, which many server admins may find useful, and provides a wealth of other server-oriented extras such as importing schedule settings from a file.

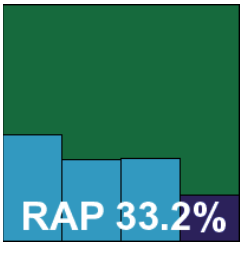


An initial run over the test sets found no problems on access, but the on-demand tests were held up while we tried to persuade the scanner to run. A batch of scheduled scans were set up to run over a weekend but failed to activate, and attempts to kick-start the same jobs manually also proved fruitless. Little information seemed available and it was not even clear whether scans were in fact running in the background and simply snagged somewhere, or not running at all. Reinstalling the product on a fresh system fixed all this however, with no repeat of the odd issues, and all tests were completed without further upset.

Scanning speeds were not excellent, with some rather heavy overheads on access, but detection was very good across all sets, with a gentle decline through the RAP sets but little missed elsewhere, including full coverage of the WildList and polymorphic sets. In the clean test sets, logs confused us for a while with their tendency to include password-protected files in the 'virus' category, and a single item, a component of the popular open-source graphics tool the *Gimp*, was mislabelled as a trojan. *BitDefender* thus also misses out on a VB100 award despite a strong showing.

CA eTrust Anti-Virus 8.1.637.0

ItW	100.00%	**Polymorphic**	92.03%
ItW (o/a)	100.00%	**Trojans**	32.03%
Worms & bots	100.00%	**False positives**	0

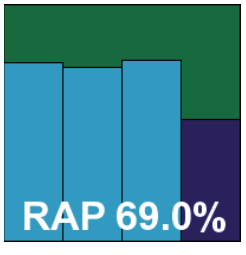


CA's *eTrust* is a corporate-focused product and has remained unchanged for several years, although the anti-malware side of the giant company has gone through a major evolution lately and we hope to see a significant overhaul of the product in the near future. Installation was somewhat arduous, with a number of lengthy EULAs which had to be scrolled through to the end to simulate reading them, and a form requiring plenty of personal information. A reboot is required to finalize the process. The interface has never been the most popular with the lab team, but worked better than usual on a server platform, presenting fewer of the slowdowns noted on some desktop tests. Testing ran through at a rapid rate, aided by the product's remarkable scanning speeds. On-access overheads were similarly feather-light, but completing the testing process was somewhat hampered by the product's horribly unfriendly logging format, which required some fairly crude hacking into shape before any useful data could be extracted.

Results were much along the lines of recent experience: fairly mediocre in the trojans and RAP test sets and with some work to do in the polymorphic set too – a fair number of samples of the new strain of W32/Virut were missed. In the WildList set there were no problems however, and with no false positives either *CA* earns another VB100 award.

ESET NOD32 Antivirus Business Edition 4.0.424.0

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	82.89%
Worms & bots	100.00%	**False positives**	0



ESET's *NOD32* has an excellent history in VB100 testing, with excellence in both detection rates and speeds, but in recent years has lost some ground in the speed area. We were interested to see if the release of version 4 would have any impact on this trend, and initial impressions during installation were fairly promising. There was a brief lag during the 'preparing to install' stage, but otherwise it was a very fast and highly user-friendly set-up process, not needing a reboot to get full protection up and running.

Running through the speed tests first, on-demand settings were pretty thorough by default and throughput seemed fairly sluggish, although it is perhaps unfair to judge against sky-high expectations and in fact it proved to be among the faster products under test, while on-access overheads were barely noticeable. A detailed and well-designed interface appeared well stocked, but a notable omission was the ability to scan archives by default – an option some admins may find useful and one which would have enabled the full running of our speed comparisons.

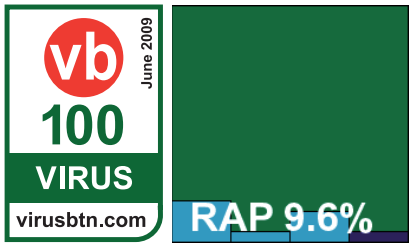
On-demand tests	WildList viruses		Worms & bots		Polymorphic viruses		Trojans		Clean sets		RAP
	Missed	%	Missed	%	Missed	%	Missed	%	FP	Susp.	
Agnitum Outpost	0	100.00%	2	99.91%	442	88.58%	1685	80.78%	0	0	58.8%
AhnLab V3Net	0	100.00%	3	99.86%	246	98.92%	1945	75.38%	1	0	55.1%
Alwil avast!	0	100.00%	2	99.91%	13	99.22%	520	94.20%	0	0	71.4%
AVG Internet Security	0	100.00%	1	99.95%	21	98.96%	290	96.15%	0	0	80.6%
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	231	96.98%	1	1	88.5%
BitDefender Security	0	100.00%	0	100.00%	0	100.00%	861	88.36%	1	0	67.9%
CA eTrust	0	100.00%	0	100.00%	1049	92.03%	6743	32.03%	0	0	33.2%
ESET NOD32	0	100.00%	0	100.00%	0	100.00%	3384	82.89%	0	0	69.0%
Fortinet FortiClient	0	100.00%	0	100.00%	202	99.15%	8823	6.46%	0	0	9.6%
Frisk F-PROT	0	100.00%	0	100.00%	165	98.93%	2370	68.49%	1	0	48.0%
F-Secure Anti-Virus	0	100.00%	0	100.00%	1	100.00%	3182	75.82%	0	2	69.8%
K7 Total Security	0	100.00%	1	99.95%	1535	75.93%	4111	64.24%	0	0	43.2%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	1	100.00%	2985	78.66%	0	0	69.3%
McAfee VirusScan	0	100.00%	0	100.00%	1	100.00%	893	88.05%	0	0	66.1%
MWTI eScan	0	100.00%	0	100.00%	0	100.00%	839	88.67%	0	0	68.5%
Netgate Spy Emergency	143	69.96%	484	77.33%	9963	1.77%	8166	14.56%	13	0	10.7%
Norman Virus Control	0	100.00%	0	100.00%	507	83.19%	2604	71.99%	0	0	48.4%
Quick Heal Anti-Virus	0	100.00%	5	99.77%	178	95.69%	899	87.95%	1	0	61.9%
Sophos Anti-Virus	0	100.00%	0	100.00%	4	99.97%	857	88.90%	0	6	81.8%
Symantec Endpoint Protection	0	100.00%	0	100.00%	1	100.00%	478	93.59%	0	0	76.0%
TrustPort Antivirus	0	100.00%	0	100.00%	131	98.82%	1705	83.63%	0	0	80.7%
VirusBuster Professional	0	100.00%	2	99.91%	442	88.58%	1734	80.25%	0	0	57.0%

Moving on to the infected sets, things went a little less smoothly. Some quirks in the operation of the on-access scanner meant having to run parts of the test by copying the sets to the test system across the network to activate detection, but this seemed reasonable in a test of fileshare protection. There were also a few occasions when the product seemed overwhelmed by the high stress it was put under, with the interface freezing up for long periods and on one occasion a reboot being needed to get things moving along.

Detection rates in the expanded polymorphic sets were impeccable, and fairly reasonable in the trojan and RAP sets, although perhaps a fraction below the excellent standards we have come to expect. This was thanks in part to a quirk which seemed to cause the on-demand scanner to ignore a fairly large number of items alerted on on-access – as these broadly fell into several clusters of near-identical files, counted as single items when calculating percentages, this impacted more heavily on the raw numbers than the percentage scores, but does seem somewhat worrying. With the WildList covered with no difficulties, however, and no false positives or other issues, *ESET* earns a VB100.

Fortinet FortiClient 4.0.1.54

ItW	100.00%	**Polymorphic**	99.15%
ItW (o/a)	100.00%	**Trojans**	6.46%
Worms & bots	100.00%	**False positives**	0



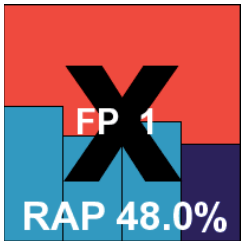
Fortinet's installation process was interrupted by several rather worrying alerts from *Windows* warning that some components were not approved by *Microsoft* and may threaten the stability of the system. Ignoring strong recommendations to abort the install, it continued fairly smoothly, but spent several minutes apparently 'optimizing performance' before the set-up process was complete. Once up and running, a revamped interface presented a smooth and colourful outlook, much more cheery than the previous effort which, while thorough and businesslike, lacked a little charm. It also seems somewhat less cluttered than the old version, while still providing a very good level of configuration and range of fine-tuning options. The defaults, set to thorough and secure, provided a stark contrast with many of the other products looked at so far, which seemed to err on the lax rather than cautious side.

With this thoroughness in evidence in some rather slow scanning speeds, one area where the defaults seemed bizarrely lacking was in the detection capabilities. The standard settings, while capable of handling the WildList quite happily and scoring reasonably well in the other older sets, showed fairly limited coverage of the new Virut strain and miserably low scores across the trojans and RAP sets. Having diagnosed this issue in previous tests, we re-ran scans after activating some additional options. With 'extended databases' enabled, as well as greyware detection and heuristics, detection rates shot up to impressive levels, with a huge leap to over 80% in the trojans set and similar levels achieved across the RAP sets, dropping fairly sharply in the 'Week+1' set.

Admins would be best advised to enable full detection capabilities, but under the VB100 rules defaults must be used (however bizarre they may seem), and the figures reported in our tables thus do not include the additional detections. Activation of the full range seemed to have little impact on the clean sets, with a few additional files labelled as suspicious, and with the default settings not raising any issues at all here a VB100 is duly awarded.

Frisk F-PROT Antivirus 6.0.9.1

ItW	100.00%	**Polymorphic**	98.93%
ItW (o/a)	100.00%	**Trojans**	68.49%
Worms & bots	100.00%	**False positives**	1

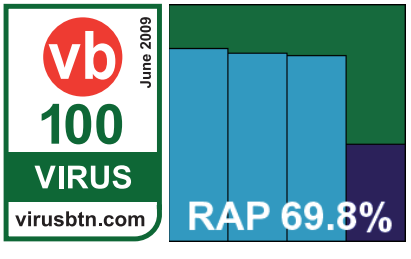


Frisk's *F-PROT* product is a pretty pared-down, bare-bones kind of affair, providing straightforward malware protection for the filesystem, with a little extra in the form of web and mail scanning. The installation process is therefore fairly simple, but seemed a little sluggish at times and needed a reboot to complete. With only basic configuration available, we relied on the defaults to see us through and got the test battery over with fairly quickly, with very good scanning speeds and minimal overheads on access.

Detection rates were a little below expectations in the RAP sets, but much better in the slightly older trojans set and splendid elsewhere, handling all the new Virut samples with aplomb. The WildList proved no problem, but in the clean set a handful of files included with some UPS management software from a major vendor were alerted on by heuristics, which was enough to count as a false alarm under our rules, thus disqualifying *Frisk* from a VB100 award this month.

F-Secure Anti-Virus for Windows Servers 8.00.14130

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	75.82%
Worms & bots	100.00%	**False positives**	0



F-Secure's server edition seems fairly similar to its standard desktop range, although the normal installation process also includes options for centralized or local management policies. Configuration in the simple, sensible interface is available in great depth, although the scheduler seemed to lack sophistication, allowing only a single job with a single target to be specified.

Scanning speeds were, as usual, on the slow side, and on-access overheads pretty hefty, but detection was generally solid, if not quite up to the expected high standards in the RAP and trojan sets. In the polymorphic set, a single instance of the latest Virut variant was not detected, but the WildList set was handled thoroughly. In the clean set, a couple of suspicious alerts were no barrier to *F-Secure* achieving a VB100 award this month.

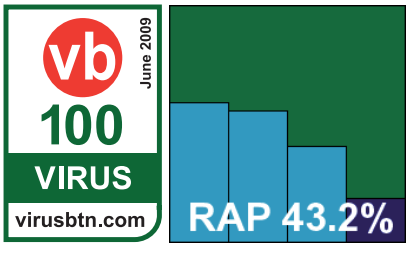
On-demand throughput (MB/s)	Archive files				Binaries and system files				Media and documents				Other file types			
	Default settings		All files		Default settings		All files		Default settings		All files		Default settings		All files	
	Time (s)	Thr. put (MB/s)	Time (s)	Thr. put (MB/s)	Time (s)	Thr. put (MB/s)	Time (s)	Thr. put (MB/s)	Time (s)	Thr. put (MB/s)	Time (s)	Thr. put (MB/s)	Time (s)	Thr. put (MB/s)	Time (s)	Thr. put (MB/s)
Agnitum Outpost	954	3.15	954	3.15	217	11.93	217	11.93	127	16.25	127	16.25	90	10.42	90	10.42
AhnLab V3Net	588	5.12	588	5.12	1303	1.99	1303	1.99	170	12.14	170	12.14	903	1.04	903	1.04
Alwil avast!	34	88.50	473	6.36	216	11.99	225	11.51	85	24.28	123	16.78	85	11.04	105	8.93
AVG Internet Security	1806	1.67	1806	1.67	278	9.31	278	9.31	161	12.82	196	10.53	35	26.80	148	6.34
Avira AntiVir	422	7.13	422	7.13	180	14.39	180	14.39	122	16.92	122	16.92	104	9.02	104	9.02
BitDefender Security	1350	2.23	1350	2.23	341	7.59	341	7.59	95	21.73	95	21.73	96	9.77	96	9.77
CA eTrust	262	11.48	262	11.48	50	51.79	50	51.79	43	48.00	43	48.00	30	31.27	30	31.27
ESET NOD32	1363	2.21	1363	2.21	376	6.89	376	6.89	55	37.53	55	37.53	55	17.06	55	17.06
Fortinet FortiClient	304	9.90	304	9.90	345	7.51	345	7.51	56	36.86	56	36.86	68	13.80	68	13.80
Frisk F-PROT	295	10.20	295	10.20	350	7.40	350	7.40	47	43.91	47	43.91	40	23.45	40	23.45
F-Secure Anti-Virus	1504	2.00	1999	1.51	425	6.09	421	6.15	94	21.96	198	10.42	64	14.66	231	4.06
K7 Total Security	136	22.13	NA	NA	213	12.16	213	12.16	31	66.58	31	66.58	34	27.59	34	27.59
Kaspersky Anti-Virus	1850	1.63	1850	1.63	363	7.13	363	7.13	189	10.92	189	10.92	203	4.62	203	4.62
McAfee VirusScan	61	49.33	689	4.37	1070	2.42	1445	1.79	100	20.64	99	20.85	112	8.38	119	7.88
MWTI eScan	521	5.78	521	5.78	1402	1.85	1402	1.85	1142	1.81	1142	1.81	873	1.07	873	1.07
Netgate Spy Emergency	31	97.07	NA	NA	105	24.66	105	24.66	99	20.85	99	20.85	67	14.00	67	14.00
Norman Virus Control	599	5.02	599	5.02	1615	1.60	1615	1.60	58	35.59	58	35.59	136	6.90	136	6.90
Quick Heal Anti-Virus	207	14.54	422	7.13	76	34.07	75	34.52	80	25.80	90	22.93	52	18.04	66	14.21
Sophos Anti-Virus	54	55.72	1636	1.84	399	6.49	523	4.95	69	29.91	165	12.51	35	26.80	213	4.40
Symantec Endpoint Protection	470	6.40	NA	NA	293	8.84	293	8.84	208	9.92	208	9.92	185	5.07	185	5.07
TrustPort Antivirus	925	3.25	925	3.25	339	7.64	339	7.64	118	17.49	118	17.49	120	7.82	120	7.82
VirusBuster Professional	365	8.24	641	4.69	169	15.32	170	15.23	66	31.27	114	18.10	21	44.67	58	16.17

K7 Total Security 9.7.0173

ItW	100.00%	**Polymorphic**	75.93%
ItW (o/a)	100.00%	**Trojans**	64.24%
Worms & bots	99.95%	**False positives**	0

K7's main market is in Japan, but the English version of the product seems pretty smooth and solid. The installation process had a slightly boxy feel, but ran through quickly, with a pause to gather some user information and a reboot at the end. It seemed to make startup slightly slower than expected, but once up and running provided a straightforward and responsive interface with a reasonable level of configuration available. Something that may prove problematic for server admins is the apparent inability to scan more than one level deep into archives, even in thorough on-demand scans.

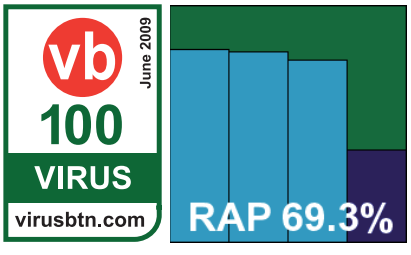
Perhaps thanks in part to these fairly minimal settings, scanning speeds were through the roof, and on-access overheads very low indeed, but detection rates



were medium at best, with a fairly steep week-on-week decline in the RAP sets and large swathes of the new Virut samples not covered. The WildList was handled without any difficulty, and the clean sets likewise, so *K7* also meets the requirements for a VB100 award.

Kaspersky Anti-Virus 6.0 for Windows Servers Enterprise Edition 6.0.2.555

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	78.66%
Worms & bots	100.00%	**False positives**	0

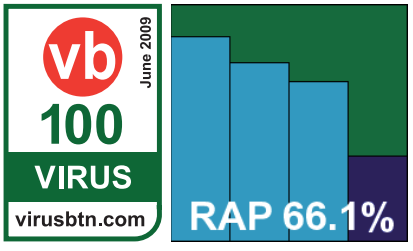


Kaspersky's server edition is a quite separate beast from the company's desktop range, with a long and complex installation process tripping through a large number of options screens. Once the process is complete many admins will also require the administration component, which rather than being an option to the main installer is in fact its own standalone module with a separate set-up process. Once everything is ready, an MMC interface provides a long and complex tree of configuration, monitoring and reporting options. This proved generally fairly simple to navigate, although there were a few moments of confusion thanks to unexpected behaviours and surprising placement of controls. A few times setting changes were rejected, and for a time some error messages appeared to say that the product had lost connection to itself. Most disturbingly, the on-demand scan settings seemed to constantly revert to defaults when changing views from one tab to another, leading to several frustrating runs through the tests as samples were trashed against our instructions. This could be a fairly serious issue in enterprise environments, where experienced admins will want to know exactly what has been found on their networks – with physical copies of files so they can be analysed and any potential breach of data privacy recorded.

Finally gathering the required data for the infected sets, detection rates proved good, but not as excellent as usual, with a sharp drop in the 'Week+1' RAP set contrasting sharply with the desktop product's performance in the last comparative (see *VB*, April 2009, p.15). Nevertheless, scores were still commendable, the WildList was covered without difficulty, and with no false positives, a VB100 award is duly granted.

McAfee VirusScan Enterprise 8.7.0i

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	88.05%
Worms & bots	100.00%	**False positives**	0

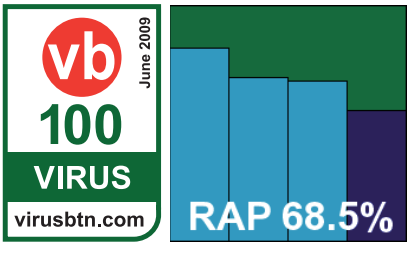


While most competitors have evolved their installers and interfaces into more shiny, colourful and cuddly versions, *McAfee*'s set-up remains sober, sensible and grey. The GUI is simplicity itself, but all the options an admin could desire are neatly tucked away in its easy-access corners. Not everything is same-old, same-old though: the new 'Artemis' in-the-cloud detection layer which has been attracting much attention in recent months, is apparently rolled into this version, as shown by a button offering additional online heuristic data. As this was disabled by default, its input did not count towards detection scores under the VB100 rules.

On-demand speeds were reasonable, on-access overheads a little heavy, with executable files particularly slow to process, and detection rates proved pretty solid, with a gradual decline across the RAP weeks to a fairly steep drop in the 'Week+1' set. The new Virut strain was not quite fully covered with a single item missed, but the WildList presented no problems and with the clean sets free from upset too, *McAfee* takes away another VB100 award.

MWTI eScan Internet Security for Windows 10.0.977.411

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	88.67%
Worms & bots	100.00%	**False positives**	0



MicroWorld's *eScan* has a rather cuddly, cartoony feel to it in places but retains an air of solidity and thoroughness nevertheless. Set-up is a breeze, but once finalized the main interface did seem rather reluctant to show itself, on occasion taking as long as 20 seconds from click to full display. There were a few similarly long lags accessing logs at times too, mostly thanks to their large

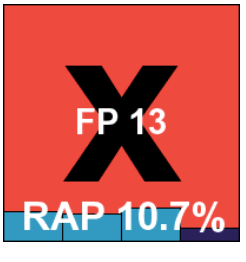
File access lag time (s/MB)	Archive files				Binaries and system files				Media and documents				Other file types			
	Default settings		All files		Default settings		All files		Default settings		All files		Default settings		All files	
	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)
Agnitum Outpost	67	0.02	NA	NA	464	0.17	464	0.17	163	0.07	163	0.07	107	0.09	107	0.09
AhnLab V3Net	82	0.03	NA	NA	222	0.08	222	0.08	113	0.04	113	0.04	100	0.09	100	0.09
Alwil avast!	142	0.05	562	0.19	272	0.10	292	0.11	194	0.08	219	0.09	165	0.16	171	0.16
AVG Internet Security	228	0.07	238	0.08	387	0.14	388	0.14	108	0.04	133	0.05	33	0.01	65	0.05
Avira AntiVir	44	0.01	172	0.06	194	0.07	194	0.07	110	0.04	150	0.06	55	0.04	146	0.14
BitDefender Security	581	0.19	1469	0.49	320	0.12	342	0.13	105	0.04	119	0.05	105	0.09	109	0.10
CA eTrust	27	0.01	NA	NA	66	0.02	66	0.02	65	0.02	65	0.02	43	0.03	43	0.03
ESET NOD32	12	0.00	NA	NA	61	0.02	61	0.02	75	0.03	75	0.03	56	0.04	56	0.04
Fortinet FortiClient	277	0.09	277	0.09	350	0.13	350	0.13	63	0.02	63	0.02	74	0.06	74	NA
Frisk F-PROT	71	0.02	NA	NA	323	0.12	323	0.12	52	0.01	52	0.01	43	0.03	43	0.03
F-Secure Anti-Virus	52	0.02	1670	0.55	370	0.14	423	0.16	150	0.06	223	0.10	148	0.14	224	0.22
K7 Total Security	76	0.02	NA	NA	250	0.09	250	0.09	57	0.02	57	0.02	52	0.04	52	0.04
Kaspersky Anti-Virus	376	0.12	1427	0.47	350	0.13	376	0.14	186	0.08	211	0.09	159	0.15	181	0.17
McAfee VirusScan	41	0.01	497	0.16	488	0.18	814	0.31	101	0.04	114	0.04	108	0.10	118	0.11
MWTI eScan	358	0.12	496	0.16	232	0.08	232	0.08	61	0.02	72	0.02	55	0.04	86	0.07
Netgate Spy Emergency	48	0.01	NA	NA	108	0.04	NA	NA	105	0.04	NA	NA	41	0.02	NA	NA
Norman Virus Control	44	0.01	NA	NA	207	0.07	207	0.07	94	0.03	94	0.03	103	0.09	103	0.09
Quick Heal Anti-Virus	15	0.00	NA	NA	66	0.02	NA	NA	65	0.02	NA	NA	29	0.01	NA	NA
Sophos Anti-Virus	63	0.02	1124	0.37	463	0.17	483	0.18	141	0.06	182	0.08	160	0.15	190	0.18
Symantec Endpoint Protection	37	0.01	NA	NA	228	0.08	228	0.08	163	0.07	163	0.07	142	0.13	142	0.13
TrustPort Antivirus	301	0.10	NA	NA	593	0.22	593	0.22	194	0.08	194	0.08	188	0.18	188	0.18
VirusBuster Professional	24	0.01	29	0.01	177	0.06	175	0.06	47	0.01	94	0.03	30	0.01	64	0.05

size after scanning large infected sets, but otherwise things were smooth and reliable. On-demand scanning speeds were very slow, but on access speeds were around the middle of the field. Detection rates were pretty good, with a very slow decline in the RAP sets and an excellent showing in the 'Week+1' set, as well as flawless coverage of the polymorphic sets. With no untoward issues in the WildList or clean sets, *eScan* comfortably wins a VB100 award.

Netgate Spy Emergency 2009 6.0.305.0

ItW	69.96%	**Polymorphic**	1.77%
ItW (o/a)	69.96%	**Trojans**	14.56%
Worms & bots	77.33%	**False positives**	13

A newcomer to the VB100 this month, *Netgate*'s *Spy Emergency* suffers from a rather improbable name with more than a hint of the rogue product about it. The product itself provides a very slick and professional installation and set-up process however, dented in seriousness only by the option to select the GUI skin colour at the end. The interface itself is also attractive and well designed, with only a minimum level of configuration, but what controls there are proved responsive. Logging proved a little less reliable, possibly thanks to inept user interaction, but nevertheless



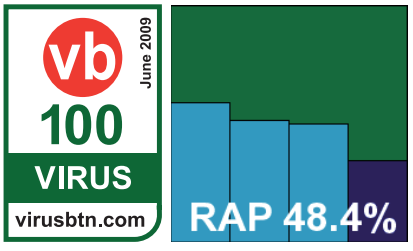
Archive scanning		ACE	CAB	EXE	JAR	LZH	RAR
Agnitum Outpost	OD	2	√	√	√	X	√
	OA	X	X	X	X	X	X
AhnLab V3Net	OD	9	9	9	9	9	9
	OA	X	X	X	X	X	X
Alwil avast!	OD	X/√	X/√	√	X/√	X/√	X/√
	OA	X/√	X/√	√	X/√	X/√	X/√
Avira AntiVir	OD	√	√	√	√	√	√
	OA	X	X/√	X/√	X/√	X/√	X/√
AVG Internet Security	OD	X	√	√	√	√	√
	OA	X	X	X	X	X	X
BitDefender Security	OD	√	√	8	√	√	√
	OA	X/8	X/8	X/4	8	X/8	X/8
CA eTrust	OD	X	√	√	√	√	√
	OA	X	X	X	1	X	X
ESET NOD32	OD	√	√	√	√	√	√
	OA	X	X	X	X	X	X
Fortinet FortiClient	OD	X	√	√	√	√	√
	OA	X	√	√	√	√	√
Frisk F-PROT	OD	1	√	√	√	√	√
	OA	1	X	2	2	X	X
F-Secure Anti-Virus	OD	X/√	5	5	5	5	5
	OA	X/√	X/5	X/5	X/5	X/5	X/5
K7 Total Security	OD	X	1	X	1	1	1
	OA	X	X	X	X	X	X
Kaspersky Anti-Virus	OD	√	√	√	√	√	√
	OA	X/√	X/√	√	X/√	X/√	X/√
McAfee VirusScan	OD	X/2	X/√	X/√	X/√	X/√	X/√
	OA	X/2	X/√	X/√	X/√	X/√	X/√
MWTI eScan	OD	√	√	8	√	√	√
	OA	X/√	X/√	X/8	√	X/√	X/√
Netgate Spy Emergency	OD	X	X	X	X	X	X
	OA	X	X	X	X	X	X
Norman Virus Control	OD	X	X	√	√	√	√
	OA	X	X	X	X	X	X
Quick Heal Anti-Virus	OD	X/2	X/5	X	X/5	X	X/5
	OA	X	X	X	X	X	X
Sophos Anti-Virus	OD	X	X/5	X/5	X/5	X/5	X/5
	OA	X	X/5	X/5	X/5	X/5	X/5
Symantec Endpoint Protection	OD	X	3/√	3/√	3/√	3/√	3/√
	OA	X	X	X	X	X	X
TrustPort Antivirus	OD	X	√	√	√	√	√
	OA	X	X	X	X	X	X
VirusBuster Professional	OD	2	√	√	X	X	√
	OA	X	X	X	X	X	X

Key:
X - Archive not scanned
√ - Archives scanned to depth of 10 or more levels
*Executable file with randomly chosen extension
X/√ - Default settings/thorough settings
[1-9] - Archives scanned to limited depth

numerous pop-up alerts failed to be recorded in initial attempts. When full detection data was finally gleaned, coverage of the sets was fairly poor, with large numbers missed in the WildList set. False positives were also an issue, with handfuls of false alarms in several of the sets, most notably a selection of samples taken from clean *Windows 98* installs, including notepad.exe, calc.exe and explorer.exe. Polymorphic detection was also fairly poor, with very few samples detected at all and no single variant fully covered. There is clearly a good deal of work to be done here before the product is ready for VB100 certification, but it seems like a decent start has been made and those hints of roguishness implied by the unfortunate title should soon be dispelled.

Norman Virus Control 5.99

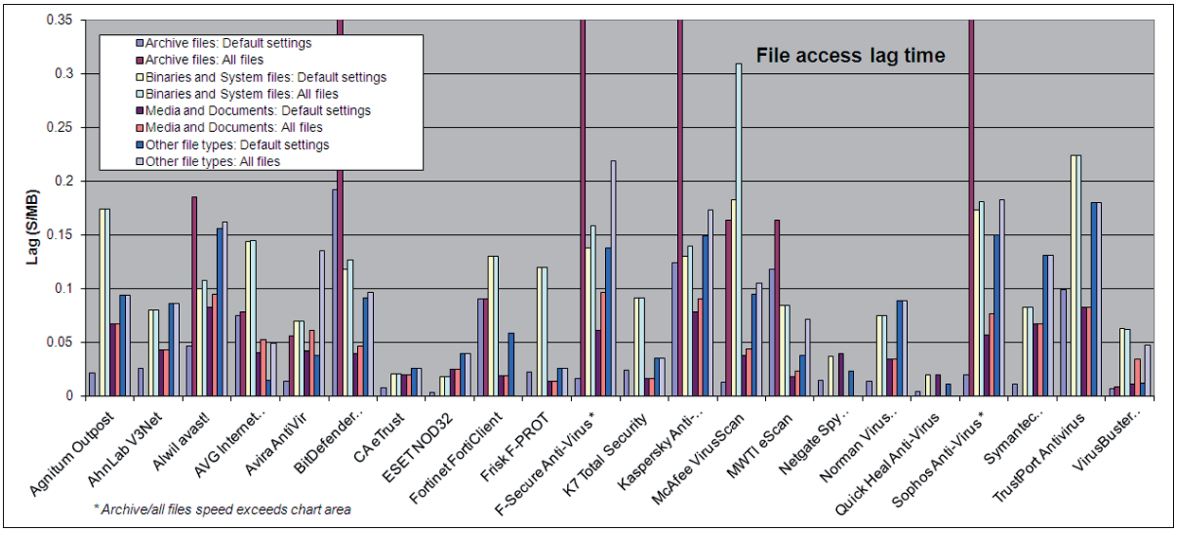
ItW	100.00%
ItW (o/a)	100.00%
Worms & bots	100.00%
Polymorphic	83.19%
Trojans	71.99%
False positives	0



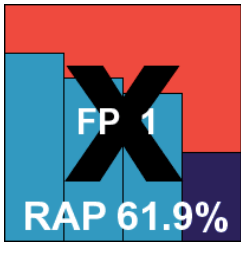
Norman's installation is simpler than most: a bare *InstallShield*-style process trips through the standard steps and ends, after suggesting that a reboot may be required, with no call for one. The *VC* control system is a rather fiddly, multi-interface system which requires several different windows to design and initiate a scan – however, with the benefit of some familiarity, it presented no serious problems. A few irritations included the absence of some options that would have been useful, some options not seeming to work, and despite explicitly setting all actions to log only, numerous samples were removed or disinfected in the various scans run. Scanning speeds and overheads were mostly fairly good, although the executable speed test set took quite some time on demand, and results were fair to middling across the various sets, with some issues apparent over the new Virut samples. The Wildlist presented no such problems however, and with no false positives either *Norman* earns a VB100 award.

Quick Heal Anti-Virus Lite 2009 10.00

ItW	100.00%	**Polymorphic**	95.69%
ItW (o/a)	100.00%	**Trojans**	87.95%
Worms & bots	99.77%	**False positives**	1



Quick Heal continues to live up to its name, providing a rapid and simple installation to go with its fast, uncomplicated product. The latest version of the interface has a crisp, clean glow about it that is very easy on the eye, and the layout remains basic but highly usable. An absence of in-depth options may put off more demanding admins, and some other issues emerged, including an apparent inability to save on-access logs and a tendency to ignore instructions not to interfere with any infections discovered. Also rather frustrating was a lengthy delay accessing browse windows when selecting targets for on-demand scans, sometimes taking over half a minute to display the filesystem. With this hurdle overcome, scanning speeds and overheads were most impressive.

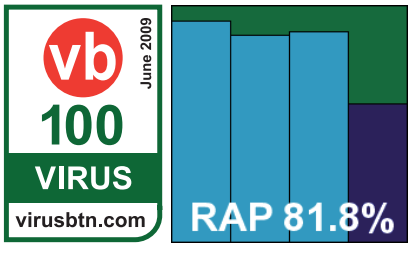


Detection was also very good, with an excellent showing in the trojans set, full coverage of all our Virut samples, and a decent performance in the RAP set-up too. With the WildList presenting no problems, only the clean sets remained an obstacle to VB100 certification, and here sadly the same browser product which tripped up another product earlier was alerted on, using the same identification – suggesting some contamination of shared sample sets somewhere – and *Quick Heal* also misses out on a VB100 award by a whisker.

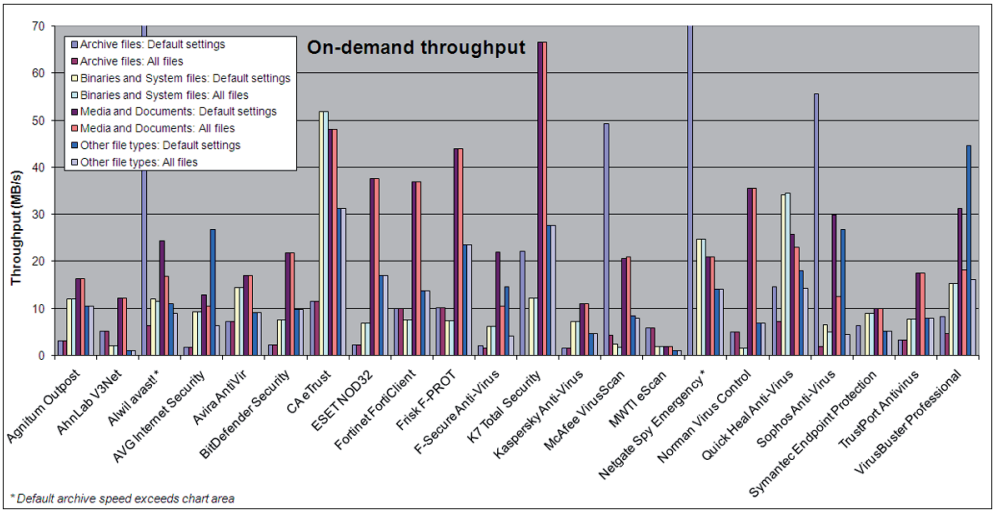
Sophos Anti-Virus 7.6.6

ItW	100.00%	**Polymorphic**	99.97%
ItW (o/a)	100.00%	**Trojans**	88.90%
Worms & bots	100.00%	**False positives**	0

Sophos's set-up procedure starts with a simple unzipping and leads through the standard stages, via an offer to remove competitors' software and a couple of command prompt windows which flicker up briefly, to full activation in short order, with no reboot required. The interface looks much as it has done for some time: a fairly plain and bare look with a splendidly complete range of configuration options available beneath the surface, including a highly advanced area where interference without expert guidance is strongly discouraged.

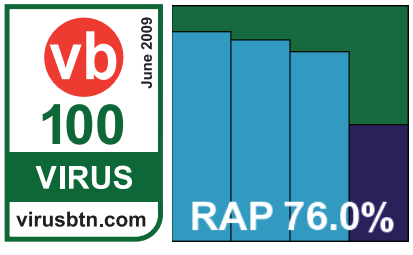


Initial attempts at the speed tests found that on-demand scans invariably included additional scanning for rootkits and suspicious files in standard areas. This added several minutes to each scan, even over a small handful of files, so tests were redone using the right-click option to more closely approximate the standards set elsewhere. The progress bar remains worse than useless, invariably shooting to 80% in the first few minutes of a scan and lingering there for most of the remainder, be that five minutes or 90, but several other products also had some issues in this area. In the final reckoning, scanning speeds were very good, on-access overheads a little heavy, but detection rates were really quite excellent across the board, with a commendably stable set of figures across the trojans and first three weeks of the RAP sets. The WildList was handled easily, and while a sprinkling of items in the clean sets were labelled vaguely suspicious, this is permissible within the VB100 rules and *Sophos* wins another VB100 award.



Symantec Endpoint Protection 11.0.4010.19

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	93.59%
Worms & bots	100.00%	**False positives**	0

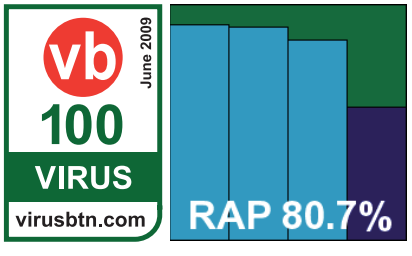


Symantec's corporate product provides options for central or local management to kick off its installation. We opted for local controls, and the rest of the set-up followed the usual path, although when it reached the end and suggested it would require 'several minutes' to tidy up after itself, it was something of a surprise to find that it actually meant it. A reboot was then required, after an attempt to update. The latest interface is a bright and shiny thing, not unpleasant to look at and providing a fair degree of configuration options in its more advanced regions (although some items we looked for were not available). The product includes some additional 'proactive' protection mechanisms, but these were disabled by default.

The system for designing and running on-demand scans proved pleasingly simple and quick to respond, and the bulk of the tests were handled with ease, producing somewhat below-par speeds in both modes but decent detection rates in most sets. These last figures were obtained only with great patience, as scanning large numbers of infected files takes some time – fortunately not a situation most admins would expect to encounter. Logging also proved rather fiddly, with the product taking an enormous amount of time to display and export logs, which in some cases seemed incomplete. Once data was finally accessed, a single sample of the latest Virut strain proved not to have been detected but the WildList was covered with ease. With no false positives either, *Symantec* earns another VB100 award.

TrustPort Antivirus 2009 2.8.0.3014

ItW	100.00%	**Polymorphic**	98.82%
ItW (o/a)	100.00%	**Trojans**	83.63%
Worms & bots	100.00%	**False positives**	0

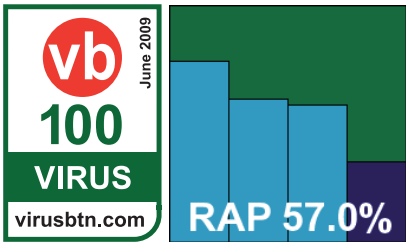


TrustPort is another product to have had something of a facelift of late, with a curvaceous new company logo, some new fonts and a new colour scheme enlivening what is essentially a very similar layout to earlier versions. The installation process includes a strongly worded warning about installing on machines running other security products, and has a post-install configuration scheme including options to control the order in which the two engines included are applied. The interface is available as a highly simplified version, or as a more advanced one. This does indeed provide an advanced level of configuration, although once again some options were clearly absent, and indeed one – the choice to scan compressed files on access – seemed to have little effect when activated. A few other small worries were encountered, most notably some slow startup times for the on-access protection, which seemed still not to be working long after the newly booted machine was responding to commands. On one occasion a scan came to a halt with the stark message that an API error had occurred. Scanning speeds were rather slow, as one would expect from

a multi-engine product, but detection rates were generally very good, although a worryingly large number of new Virut samples were not flagged. There were no problems in the WildList or elsewhere, and *TrustPort* thus also earns a VB100 award.

VirusBuster Professional for Windows Servers (x64) 6.1.130

ItW	100.00%	**Polymorphic**	88.58%
ItW (o/a)	100.00%	**Trojans**	80.25%
Worms & bots	99.91%	**False positives**	0



Bringing up the rear of the alphabetical product list as usual, *VirusBuster*'s server edition presents a rather confusing mix of the desktop and server approach. The installation process is fairly simple, and when up and running an interface can be accessed from the system tray and looks very similar to the standard desktop GUI. A brief browse through it, however, revealed that several standard options, and indeed sets of options, are not available here. To find them, one must turn to a second, MMC-based console, for which a shortcut is dropped onto the desktop. This made for some slightly odd flipping between the two as different tweaks needed to be made in various places. Occasionally some slow response times also frustrated, particularly when adjusting the targets of a scan, with long pauses after each stage of the set-up process. Finally, an issue which has been noted here several times before: the option to enable on-access scanning of archives is provided but appears entirely ineffectual.

Despite these minor irritants, scanning speeds were excellent and detection rates not bad at all, although as with so many other products this month, some work may need to be done on the latest W32/Virut strain. For now, however, the WildList set presented no issues, and without false positives either *VirusBuster* earns another VB100 award.

CONCLUSIONS

As expected, the 64-bit platform brought out quite a number of quirks and oddities in several of the products under test. While last month's comparative suffered from a rash of severe stability issues, with systems freezing and crashing all over the place, this kind of problem was less evident this time, although not completely absent. This is to be expected, as server products do generally need to be more resilient, and crashing a server system is a big sin for any software. However, this month's batch of products showed some more insidious problems, with logging inaccuracies, settings seeming to readjust themselves in some products, while in others they were simply ignored. These are also pretty big crimes in a server system, where admins expect their security software to conform to their requirements and not go off doing its own thing. We have emphasized the availability (or otherwise) of configuration options and fine-tuning controls throughout this month's review, as this is an important aspect of products in a server setting – some of the products proved somewhat lacking in this area.

Detection also seemed a little uneven in some products, with oddly differing behaviour in different modes. Some products did not perform as well as previous experience led us to expect, much of which can be put down to the complexities of the platform and the fact that many developers seem to put more effort into desktop and home-user solutions than into server products. Thanks to this, the RAP results have yet to settle down and show any steady patterns across the board, but after three outings some top performers are starting to emerge, while the rest jostle for position below them.

Since the last test we have been doing some filtering of our clean sets to ensure the most obscure and improbable items are removed. Many of these, including several previously alerted on as false alarms, have been kept handy in a side-set and monitored during testing. This has shown an increasing trend of false alarms spreading across the industry, as clean items make their way into sample collections and are blindly added to detection databases by automated systems. This is perhaps an inevitable side effect of the increased use of such automation, but is a danger labs need to be alert to and should mitigate as best they can.

Of the products failing this month, most were fairly clear false positive issues, of which a few seemed to be shared between products; some products were unlucky with fairly minor false alarms, while the lone newcomer, with a more sizeable clutch of false positives, was expected to have some teething issues and will doubtless improve rapidly. The WildList was handled fairly easily, but next time it should present a much tougher challenge, with the latest W32/Virut strain almost certain to stay in the list long enough to make the next test set and still proving to cause difficulties several months after it was first observed.

Technical details

All products were tested on identical systems with *AMD Athlon64 X2* Dual Core 5200+ processors, 2 GB RAM, dual 80GB and 400GB hard drives, running *Microsoft Windows 2003 Server R2 SP2*, x64 edition.

END NOTES & NEWS

RSA Japan takes place 10–12 June 2009 in Tokyo, Japan. Full details can be found at http://www.cmptech.jp/dcw/rsa/.

The Conference on Cyber Warfare will be held 17–19 June 2009 in Tallinn, Estonia. The conference will cover big-picture perspectives such as concepts, policy, and doctrine, as well as topics of a more technical nature. See http://www.ccdcoe.org/cyberwarfare/.

The 21st annual FIRST conference will be held 28 June to 3 July 2009 in Kyoto, Japan. The conference will focus on issues relevant to incident response and security teams. For more details see http://conference.first.org/.

A Mastering Computer Forensics masterclass will take place 22–23 July 2009 in Jakarta, Indonesia. For details see http://www.machtvantage.com/computerforensics.html.

Black Hat USA 2009 will take place 25–30 July 2009 in Las Vegas, NV, USA. Training will take place 25–28 July, with the briefings on 29 and 30 July. For details see http://www.blackhat.com/.

The 18th USENIX Security Symposium will take place 12–14 August 2009 in Montreal, Canada. The 4th USENIX Workshop on Hot Topics in Security (HotSec '09) will be co-located with USENIX Security '09, taking place on 11 August. For more information see http://www.usenix.org/events/sec09/.

The International Cyber Conflict Legal & Policy Conference 2009 will take place 9–10 September 2009 in Tallinn, Estonia. The conference will focus on the legal and policy aspects of cyber conflict. For details see http://www.ccdcoe.org/126.html.

IMF 2009, the 5th International Conference on IT Security Incident Management & IT Forensics takes place 15–17 September 2009 in Stuttgart, Germany. Experts will present and discuss recent technical and methodical advances in the fields of IT security incident response and management and IT forensics. For more information see http://www.imf-conference.org/.

SOURCE Barcelona will take place 21–22 September 2009 in Barcelona, Spain. The conference will be run in two tracks: Security and Technology, covering security software, application security, secure coding practices, engineering, new tool releases and technology demonstrations; and Business of Security, covering critical decision-making, entrepreneurship, issues of compliance, regulation, privacy laws, disclosure and economics. For full details and registration see http://www.sourceconference.com/.

Hacker Halted 2009 takes place in Miami, FL, USA, 23–24 September 2009. See http://www.hackerhalted.com/.

VB2009 will take place 23–25 September 2009 in Geneva, Switzerland. Early bird registration rates apply until 15 June 2009. For the full conference programme including abstracts for all papers and online registration, see http://www.virusbtn.com/conference/vb2009/.

The third APWG eCrime Researchers Summit will be held 13 October 2009 in Tacoma, WA, USA in conjunction with the 2009 APWG General Meeting. eCrime '09 will bring together academic researchers, security practitioners and law enforcement to discuss all aspects of electronic crime and ways to combat it. For more details see http://www.ecrimeresearch.org/.

Malware 2009, the 4th International Conference on Malicious and Unwanted Software, will take place 13–14 October 2009 in Montreal, Quebec, Canada. For more information see http://www.malware2009.org/.

The SecureLondon Workshop on Information Security Audits, Assessments and Compliance will be held on 13 October 2009 in London, UK. See http://www.isc2.org/EventDetails.aspx?id=3812.

RSA Europe will take place 20–22 October 2009 in London, UK. For full details see http://www.rsaconference.com/2009/europe/.

AVAR2009 will be held 4–6 November 2009 in Kyoto, Japan. More information will be announced in due course at http://www.aavar.org/.

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)1865 543153
Email: editorial@virusbtn.com Web: http://www.virusbtn.com/

vb Spam supplement

CONTENTS

NEWS & EVENTS

PHISHER GETS 8.5 YEARS

A Romanian man living in the United States has been sentenced to eight and a half years in prison for phishing scams that netted him approximately $700,000 from 7,000 individuals.

Through various phishing techniques, Sergiu D. Popa stole names and addresses, bank account numbers, PINs, credit card details and social security information and used these details to siphon money away from his victims' accounts. He also offered phishing toolkits for sale (for the bargain price of $1,500), complete with step-by-step instructions on to how to use them to maximum effect. 23-year-old Popa is reported to have begun his phishing schemes as a youngster as long ago as 2000. On handing out his sentence, Judge John Tunheim, said: 'There needs to be a deterrent to others who are trying similar crimes over the Internet.'

EVENTS

The 16th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will be held in Amsterdam, The Netherlands, 9–11 June 2009. The 17th general meeting will be held 26–28 October 2009 in Philadelphia, PA, USA. Meetings are open to members and invited participants only. See http://www.maawg.org/.

Inbox/Outbox 2009 takes place 16–17 June 2009 in London, UK. See http://www.inbox-outbox.com/.

The sixth Conference on Email and Anti-Spam (CEAS) will be held 16–17 July 2009 in Mountain View, CA, USA. See http://www.ceas.cc/.

The 7th German Anti-Spam Summit takes place 14–16 September 2009 in Wiesbaden, Germany (the event will be held in English). See http://www.eco.de/veranstaltungen/7dask.htm.

SPAMBOT CASE STUDY

WHERE IS WALEDAC?

Scott Wu, Terry Zink, Scott Molenkamp
Microsoft, USA

Win32/Waledac [1] is a trojan that is used to send spam. It also has the ability to download and execute arbitrary files, harvest email addresses from the local machine, perform denial of service attacks, proxy network traffic and sniff passwords.

Waledac first drew significant attention in December 2008 via a Christmas-themed postcard lure. In the six months since, many users have been the recipient of various other eye-catching lures sent by Waledac. From the perennial holiday-themed lures to the more recent 'Reuters Terror Attack' or 'SMS Spy' themes, downloading a variant of Waledac is only a single, socially engineered step away.

When it was unleashed in December, Win32/Waledac was by no means an under-developed piece of malware. The authors had been testing and developing the capabilities for at least a year prior to its release. The earliest known binary we were able to find in the wild was from 25 December 2007. The developmental progression of Win32/Waledac can be traced by its internal version numbers. In this case, the version was '0'.

A major point in development came with the release of version 15 in the last week of November 2008. This was the first version to support 'labels'. The label would essentially provide a mechanism to identify and segment drones and the tasks designated to them. The labels appear to be used as affiliate identifiers.

Whilst the major distribution vector for Waledac appears to be through the use of spam campaigns and web hosting on compromised machines, the trojan may also be installed via a custom downloader. These custom downloaders are easily recognized as members of the Waledac family because they employ the same downloading technique as the main component. The technique is to decode an encrypted binary appended to a legitimate JPG. The encryption and the marker separating the JPG from the encrypted data are the same for the downloader and the main component.

We observed that the filename of the JPG retrieved was equivalent to the label contained within the binary itself. Some of the labels observed in samples in the wild have the appearance of a 'handle'. For example:

alekseyb	mirabella_site
birdie2	prado
dekadent	semgold
dmitriy777	shmel
ftpfire	twist
gorini4	ub
lynx	zlv
mirabella_exp	59xx39

Searching on the Internet for these labels produces some circumstantial evidence to support this theory. In some cases, where the number of results yielded is low, there is a bias towards Russian-hosted websites.

The authors of Waledac appear to have established a relationship of some description with other malware authors. The most notable demonstration of this is by a variant of Win32/Conficker [2]. This particular variant was able to download an encrypted copy of Waledac. The Conficker binary used a private key to decrypt the file from the host 'goodnewsdigital.com'.

This suggests a level of co-operation, as the Waledac authors would be required to encrypt a binary to an affiliate's specifications. An alternative scenario is that affiliates have the privilege to 'publish' binaries to the distributed hosting network. Therefore, any additional cryptographic transformations could be performed independently.

In addition to Conficker, trojan downloaders such as Win32/Bredolab [3] have also been observed to retrieve Waledac binaries hosted at 'goodnewsdigital.com'. The label of the Waledac variant downloaded by Conficker was 'twist'. The label of the binary downloaded by Bredolab was 'dmitriy777'.

Figure 1: Win32/FakeSpypro – the fact that Win32/Waledac has installed rogue security applications demonstrates that there is money to be made from affected users.

Waledac has the ability to update itself by downloading and executing a newer version from the Internet. This downloading capability is also leveraged to install other malware such as Win32/Rugzip, though perhaps the most interesting piece of malware downloaded recently is Win32/FakeSpypro [4]. The fact that Waledac has installed rogue security applications demonstrates that there is money to be made from affected users.

THE TELEMETRY

Now let's take a look at the *MSRT* (*Malicious Software Removal Tool*) [5] telemetry from April, the month in which Waledac was added to the *MSRT*. Waledac was the

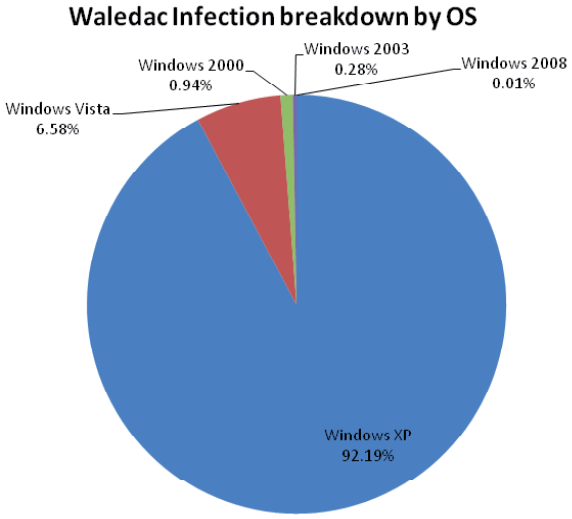


Figure 2: Waledac infection breakdown by OS.

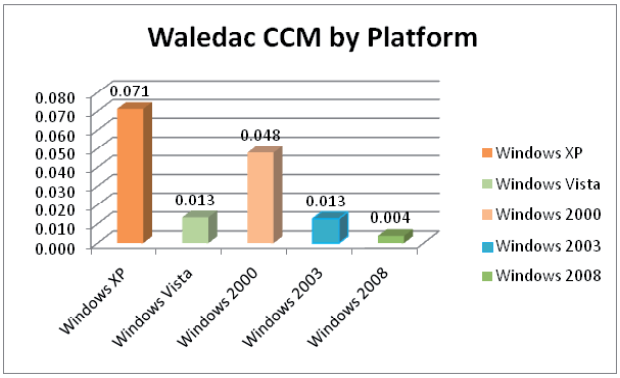


Figure 3: Waledac computers cleaned per thousand (CCM) by platform.

twenty-fourth-most prevalent family during this month. More than 24,000 distinct machines were reported with a Waledac infection worldwide. Waledac is deployed mostly on *Windows XP* (see Figure 2). Note this is not normalized. As of today, the *MSRT* installation base on *Vista* is about 37% the size of that on *Windows XP*.

If we take another step to normalize the infection rate by OS, factoring in the *MSRT* install base, Figure 3 shows that *Windows XP* has the largest number of computers cleaned per thousand *MSRT* executions (CCM). Here, CCM is a metric for infection rate based on the *MSRT* data widely used in the *Microsoft Security Intelligence Report* [6].

Breaking down the reports by country and performing the same normalization with the *MSRT* install base, we derive the following table for infection rate. The table presents the top 25 most 'infected' countries, ranked by CCM. Turkey has the highest infection rate, followed by Hungary, Russia and the United States:

Country/Region	Infected machines	MSRT executions	CCM
Turkey	931	5,903,320	0.158
Hungary	233	1,895,020	0.123
Russia	615	5,554,600	0.111
United States	13,739	124,595,720	0.110
Poland	453	6,390,100	0.071
Norway	198	2,810,480	0.070
Greece	127	1,808,840	0.070
Netherlands	495	8,443,520	0.059
Sweden	269	4,626,080	0.058
Czech Republic	158	2,893,520	0.055
Finland	126	2,382,400	0.053
Portugal	148	2,918,880	0.051
France	963	20,042,000	0.048
Spain	498	11,281,800	0.044
Australia	334	7,612,860	0.044
Denmark	136	3,362,960	0.040
United Kingdom	863	23,238,480	0.037
Belgium	118	3,618,320	0.033
Brazil	399	13,736,700	0.029
Canada	399	14,682,640	0.027
Mexico	176	7,065,520	0.025
Korea	353	14,182,700	0.025
Italy	288	13,001,040	0.022
Japan	707	34,302,520	0.021
Germany	384	26,684,400	0.014

Waledac is highly polymorphic. From over 24,000 infected machines there were 2,452 unique Waledac binaries. The following table shows the top 10 reported Waledac hashes. The top six files reported are internal version 34, which was the most recent at the time of the April *MSRT* release.

MD5	Infected machines	Internal file version	Binary label
02782ddfbd851ce17c68dce078dde190	2,454	34	dmitriy777
82008273fc6eff975e0cf3bfc0e2396f	2,344	34	mirabella
fdd5c061cda0e205e00a849a8e8e6f7a	1,693	34	dmitriy777
10868273a15688d11ccb584653542833	1,132	34	birdie2
223111097b81773822a45b73bac1370a	858	34	ub
55cd9f80b39b1b566d9bbde5815c0969	788	34	dmitriy777
cdee7ff3d373ec38f8b67accdfc1ffe4	540	22	59xx39
dd3de6413bfe3e442d85fdef82297c84	497	31	mirabella
b7db1a54faa4d7b9800393407c0f4dfe	450	33	dmitriy777
4ada90839a8ac31d4f828e9229dfa24f	440	34	ub

THE SPAM DATA

Over the period 16–21 April 2009, *Forefront Online Security for Exchange* (*FOSE*) tracked data on Waledac-related spam. In the study, the following domains were tracked:

bestgoodnews.com
breakinggoodnews.com
bchinamobilesms.com
bsmspianeta.com
bfreeservesms.com
bmiosmsclub.com
bsmsclubnet.com

By observing *FOSE* customers' incoming mail containing these links, it was possible to capture all of the IPs that sent this mail. These IPs were analysed and the sum total of *all* mail sent from these IPs was calculated (not just the mail containing the Waledac spam links). Next, a geographical distribution was sketched showing the allocation of the IPs according to their sending source.

One of the characteristics of the Waledac botnet is that it sends a high proportion of mail with an empty MAIL FROM < > field. Empty senders are not included in either the total spam count or the total mail count, but they are included in the average number of mails sent per IP. Empty sender mail could be spam (such as that occurring in Waledac spam) or it could be backscatter mail. This distinction is not made in the statistics below.

Region	Total spam	Total mail	Empty sender mail	Distinct IPs	Avg. mail/IP
North America	25,786,958	72,756,248	4,220,617	1,801	42,741
Europe	3,976,965	9,491,166	4,013,400	1,561	8,651
Asia	838,969	1,661,167	1,417,824	3,079	1,000
Oceania	58,338	329,307	104,024	477	908
South America	88,794	267,936	60,187	156	2,103
Central America	3,226	13,292	2,035	25	613
Africa	9,554	10,323	897	4	2,805
Total	**30,762,804**	**84,529,439**	**9,818,984**	**7,103**	**13,283**

As a proportion of total overall mail, showing the percentages:

Region	Total spam	Total mail	Empty sender mail	Distinct IPs
North America	83.83%	86.07%	42.98%	25.36%
Europe	12.93%	11.23%	40.87%	21.98%
Asia	2.73%	1.97%	14.44%	43.35%
Oceania	0.19%	0.39%	1.06%	6.72%
South America	0.29%	0.32%	0.61%	2.20%
Central America	0.01%	0.02%	0.02%	0.35%
Africa	0.03%	0.01%	0.01%	0.06%

From the above tables, observe that total spam is only a small proportion of the total mail. Slightly more than a third of North America's mail is marked as spam, and the numbers are not dissimilar for the other regions. This implies that the Waledac botnet is spread very widely on machines that do not typically send high volumes of spam. In other words, the sending machines are compromised, but the amount of mail sent per bot is sufficiently small so as to hide it within a larger, overall good mail stream.

The next table shows the IP distribution per country, sorted by the total amount of empty sender mail. Manual inspection of a number of Waledac-related spam messages confirmed that much of the spam was sent with empty MAIL FROMs. The average mail/IP includes the empty sender count.

Country	Total spam	Total mail	Empty sender mail	Distinct IPs	Avg. mail/IP
United States	25,365,150	71,436,463	4,051,357	1,704	44,300
Great Britain	1,011,802	2,675,004	1,348,016	195	20,631
France	1,468,165	2,853,418	1,222,272	74	55,077
Japan	616,498	1,128,727	754,919	229	8,226
Austria	10,306	102,285	411,946	34	15,124
Sweden	265,132	831,033	353,551	20	59,229
Germany	517,055	1,234,721	281,833	108	14,042
Canada	329,430	1,188,341	164,631	81	16,703
Australia	55,625	320,178	102,928	137	3,088
Italy	78,813	167,939	95,768	137	1,925
China	16,272	47,370	81,395	1,306	99
Switzerland	48,594	94,724	72,574	23	7,274
Singapore	44,113	166,315	68,674	37	6,351
United Arab Emirates	35,473	186,411	47,622	14	16,717
The Netherlands	52,613	347,000	47,094	77	5,118
Spain	114,743	134,229	32,941	103	1,623

Argentina	35,942	63,445	28,202	132	694
Czech Republic	6,481	137,183	27,111	74	2,220
Brazil	23,694	161,893	24,380	231	806
Norway	10,577	286,029	24,363	15	20,693
Ireland	5,403	37,722	16,643	24	2,265
Mexico	92,378	131,444	4,629	16	8,505
Chile	28,179	37,434	966	23	1,670
Belarus	8,930	36,362	380	1	36,742
Slovakia	301,530	354,581	354	10	35,494
All others	579,784	889,650	154,412	1,358	769

As a proportion of relative totals:

Country	Total spam	Total mail	Empty sender mail	Distinct IPs
United States	82.45%	84.51%	41.26%	23.98%
Great Britain	3.29%	3.16%	13.73%	2.74%
France	4.77%	3.38%	12.45%	1.04%
Japan	2.00%	1.34%	7.69%	3.22%
Austria	0.03%	0.12%	4.20%	0.48%
Sweden	0.86%	0.98%	3.60%	0.28%
Germany	1.68%	1.46%	2.87%	1.52%
Canada	1.07%	1.41%	1.68%	1.14%
Australia	0.18%	0.38%	1.05%	1.93%
Italy	0.26%	0.20%	0.98%	1.93%
China	0.05%	0.06%	0.83%	18.38%
Switzerland	0.16%	0.11%	0.74%	0.32%
Singapore	0.14%	0.20%	0.70%	0.52%
United Arab Emirates	0.12%	0.22%	0.48%	0.20%
The Netherlands	0.17%	0.41%	0.48%	1.08%
Spain	0.37%	0.16%	0.34%	1.45%
Argentina	0.12%	0.08%	0.29%	1.86%
Czech Republic	0.02%	0.16%	0.28%	1.04%
Brazil	0.08%	0.19%	0.25%	3.25%
Norway	0.03%	0.34%	0.25%	0.21%
Ireland	0.02%	0.04%	0.17%	0.34%
Mexico	0.30%	0.16%	0.05%	0.23%
Chile	0.09%	0.04%	0.01%	0.32%
Belarus	0.03%	0.04%	0.00%	0.01%
Slovakia	0.98%	0.42%	0.00%	0.14%
All 85 others	0.71%	0.44%	5.65%	32.37%

The United States is first in this list and it appears to send a disproportionate amount of spam compared to the number of distinct IPs associated with it, but if we compare it to the others like France, Sweden and Belarus, it is not the worst offender. One surprise finding in this list is China, which ranks eleventh in the list. Even though it accounts for nearly one fifth of all the IPs found in the botnet, it accounts for less than 1% of the spam sent. In fact, looking at both sets of data, by continent and by country, Waledac is more likely to be found in the western hemisphere than in the eastern hemisphere.

If we compare North America to Europe, we see that substantially more mail comes from North America than from Europe if we exclude empty sender mail. Yet, if we isolate only that particular type of mail, then the two regions are very similar to each other.

REFERENCES

[1] Waledac MMPC encyclopedia. http://www.microsoft.com/security/portal/Entry.aspx?Name=Win32%2fWaledac.

[2] Conficker MMPC encyclopedia. http://www.microsoft.com/security/portal/Entry.aspx?Name=Win32%2fConficker.

[3] Bredolab MMPC encyclopedia. http://www.microsoft.com/security/portal/Entry.aspx?Name=TrojanDownloader%3aWin32%2fBredolab.

[4] FakeSpypro MMPC encyclopedia. http://www.microsoft.com/security/portal/Entry.aspx?Name=Trojan%3aWin32%2fFakeSpypro.

[5] The Microsoft Windows Malicious Software Removal Tool Knowledgebase 890830. http://support.microsoft.com/?kbid=890830.

[6] Microsoft Security Intelligence Report. http://www.microsoft.com/security/portal/sir.aspx.