APRIL 2009

virus BULLETIN

Fighting malware and spam

CONTENTS

ISSN 1749-7027

IN THIS ISSUE

ROGUE TRADERS

Rogue anti-malware applications have been around for several years, conning and causing confusion among users as well as posing problems for anti-malware vendors. Gabor Szappanos takes a look at a piece of anti-virus scamware.

page 9

APPLE CATCHER

Mario Ballano Barcena and Alfredo Pesoli take a detailed look at what appears to be the first real attempt to create a Mac botnet.

page 12

VB100 ON WINDOWS XP

VB's anti-malware testing team put a bumper crop of products through their paces on Windows XP. Find out which products excelled and which have some more work to do.

page 15

vbSpam supplement

This month: anti-spam news and events; and John Levine looks at message authentication using Domain Keys Identified Mail (DKIM).

'An even better solution is to be proactive in the cloud.'

Luis Corrons
Panda Security

FLOODING THE CLOUD

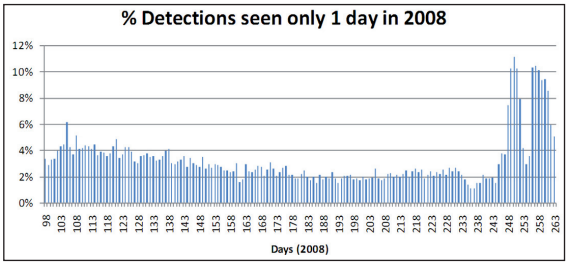
Over the past couple of years we have been hearing the same thing over and again from multiple sources in the AV industry: that the amount of malware in existence is increasing exponentially. In 2008 AV companies were reporting more than 13 million malware samples and in March 2009, *McAfee*'s *AVERT Labs* announced that it had received its 20 millionth malware sample.
The announcement caught my interest so I took a look at our own database and found that we had approximately the same number: on 12 March 2009 it contained 20,064,146 confirmed malware samples.

There are a couple of explanations for this endless flood of new malware samples. First is the large number of point-and-click applications readily available on the black market. Cybercriminals can use these to create custom trojans and pack them without having to write a single line of code. This means that even those without programming skills are able to create new pieces of malware extremely quickly and in large numbers.

The second explanation is server-side polymorphism. When infecting machines in order to steal data, create a botnet, etc., why should the cybercriminals use the same binary code to infect, say, 10,000 computers? Almost every anti-malware solution relies primarily on detection signatures, and an effective way to defeat weak signatures is to make each file slightly different – thus a very efficient infection strategy is to infect every machine with a slightly different sample. This can be achieved with server-side polymorphism: a polymorphic engine resides remotely on a server and distributes mutated variations of malware in large volume. While this strategy won't work against all technologies (for example it is ineffective against HIPS, advanced heuristics, generic detection etc.), it is well worth the effort for its ability to evade signature detection.

I was interested to find out whether these explanations could be verified by our detection data – for example to see for how long each threat was active. I decided to dig into the 2008 data in our database. I ignored MD5s in order to avoid bias caused by minor changes in the malware files, and instead focused purely on detections. I looked at detections over a time period from the 98th day to the 263rd day of 2008.

First, I calculated the total number of detections seen in each 24-hour period – regardless of whether we had seen each threat one time or one million times. Then I calculated the percentage of detections that were seen on that day only – of all the active detections seen each day, an average of 4% were not seen again.



There is no doubt that the volume of malware has been increasing exponentially over the last few years. If 4% of the detected threats are active only for a period of 24 hours, detection must be proactive and as fast as possible. I believe that the fastest way to deliver knowledge is via the cloud, and a number of security companies are adopting this technology.

An even better solution is to be proactive in the cloud – taking advantage of the user community to detect malware that not even the cloud has seen before. This can be done by merging the proactive technologies available at customers' endpoints with the cloud: as soon as a malicious process is detected in a user's PC (whether by system heuristics, emulation, sandboxing or behavioural analysis, etc.), the rest of the users worldwide will automatically benefit from that specific detection. This results in a close-to-real-time detection, not only of initial malware outbreaks but also of targeted attacks whose objective is to infect a very small number of users and stay below the radar. I truly believe that this is the best AV companies can do right now, and I hope more will follow suit. Let's see if we can build a safer world!

NEWS

GHOSTLY GOINGS ON

Last month saw the publication of a research paper reporting on a 10-month investigation of an alleged Chinese spy operation against Tibetan organizations. The investigation not only uncovered evidence of tampering with the Tibetan systems, but also evidence of a more widespread cyber espionage network of over 1,295 infected computers in more than 100 countries – dubbed GhostNet.

The research, conducted by the Information Warfare Monitor, consisted of field-based operations in India, Europe and North America, followed by lab-based data analysis. It was the data analysis phase of the investigation that led to the discovery of insecure, web-based interfaces controlling four servers. The interfaces allow attackers to communicate with compromised computers (sending instructions and receiving data). Further investigation of the servers revealed an extensive network of at least 1,295 compromised computers in 103 countries. Furthermore, the team determined that almost 30% of the infected computers could be considered 'high-value targets' – including those of ministries of foreign affairs, embassies, international organizations, news media, and NGOs.

The researchers are careful to point out that, although circumstantial evidence points to the Chinese state as being the main source of the network, they are unable to reliably ascertain either the motivation or the identity of the attackers/controllers of the network.

The full report can be downloaded at http://www.scribd.com/doc/13731776/Tracking-GhostNet-Investigating-a-Cyber-Espionage-Network. A report by researchers Shishir Nagaraja and Ross Anderson detailing their part of the Tibetan investigation can be read at http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-746.pdf.

INTERNET FRAUD COMPLAINTS RISE

An annual report released by the Internet Crime Complaint Center (IC3) – a non-profit fraud-monitoring organization run by the FBI and the National White Collar Crime Center – shows that the number of complaints of cybercrime it registered in 2008 reached a record 275,284 – a 33.1% increase over the previous year.

The total loss linked to online fraud in 2008 was $265 million – while just four years previously the total dollar loss from all referred cases of fraud had been $68 million. Complaints registered during 2008 covered a range of fraud types, with the non-delivery of merchandise and/or payment accounting for the highest number of complaints (32.9%), followed by Internet auction fraud (25.5%) and credit/debit card fraud (9.0%). The report can be read at http://www.ic3.gov/media/annualreports.aspx.

Prevalence Table – February 2009

Malware	Type	%
NetSky	Worm	23.81%
Mytob	Worm	13.20%
Virut	Virus	9.45%
Iframe	Exploit	5.88%
Downloader-misc	Trojan	5.68%
Agent	Trojan	5.24%
Mydoom	Worm	4.47%
Basine	Trojan	3.77%
Invoice	Trojan	3.35%
PWS-misc	Trojan	2.27%
Bifrose/Pakes	Trojan	2.25%
Autorun	Worm	2.14%
Bagle	Worm	1.74%
Delf	Trojan	1.61%
Suspect packers	Misc	1.53%
Zafi	Worm	1.31%
Inject	Trojan	1.25%
OnlineGames	Trojan	0.88%
Murlo	Trojan	0.73%
Sality	Virus	0.67%
Parite	Worm	0.67%
Small	Trojan	0.66%
Tenga	Worm	0.65%
Cutwail/Pandex/Pushdo	Trojan	0.56%
VB	Worm	0.39%
Grew	Worm	0.38%
Heuristic/generic	Misc	0.36%
Alman	Worm	0.31%
Backdoor-misc	Trojan	0.29%
Nimda	Worm	0.28%
Brontok/Rontokbro	Worm	0.26%
AutoIt	Trojan	0.24%
Fuzen	Rootkit	0.24%
Others[1]		3.48%
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 FIVE

Peter Ferrie
Microsoft, USA

New anti-unpacking tricks continue to be developed as the older ones are constantly being defeated. This series of articles (see also [1–4]) describes some tricks that might become common in the future, along with some countermeasures.

This article will concentrate on anti-debugging tricks that target the *OllyDbg* debugger. All of these techniques were discovered and developed by the author.

1. OllyDbg-specific tricks

OllyDbg is perhaps the most popular of user-mode debuggers. It contains a number of vulnerabilities.

1.1 Malformed files

OllyDbg 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. The file is loaded in *OllyDbg*, but the entry point's break point is not set.

OllyDbg 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 *OllyDbg* will start overwriting memory until a crash occurs.

If the value of the Base Relocation Directory Size field is 0x3FFFFFFE or larger, then *OllyDbg* will write the relocated values to 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 in excess of one gigabyte in size, because *OllyDbg* reads the relocation data directly from the file.

The Export Address Table Entries and Base Relocation Directory Size bugs affect all versions of *OllyDbg*, including 2.00h. The author of *OllyDbg* released version 2.00h almost 60 days after the bugs were reported, but it still contains the bugs. He has not responded to the report.

1.2 OutputDebugString

OllyDbg passes user-defined data directly to the msvcrt _vsprintf() function. The data can contain formatting string tokens which can cause the _vsprintf() function to access arbitrary memory via the '%s' token. A number of variations of the attack exist, but three tokens are all that is required. The first two tokens can be of any format ('%c', '%x', etc.), but the third token must be a '%s'. This is because the _vsprintf() function calls the __vprinter() function, and passes a zero as the fourth parameter. The fourth parameter is accessed by the third token if the '%s' is used there.

2. OllyDbg plug-ins

OllyDbg supports plug-ins. A number of packers have been written that are able to detect *OllyDbg*, so plug-ins have been created to attempt to hide *OllyDbg* from those packers. The following is a description of some of those plug-ins, as well as the vulnerabilities that could be used to detect them.

2.1 antiAnti

antiAnti is hard-coded for a particular language version of *Windows XP SP2* because it uses a constant value for the service table index and for the ntdll NtSetInformationThread() and ntdll NtQueryInformationProcess() functions.

The plug-in patches the debuggee's ntdll NtSetInformationThread() and ntdll NtQueryInformationProcess() function codes so that they simply return. This behaviour is a bug, since the return code is never set.

antiAnti injects a thread into the debuggee's process space, which sets the PEB->BeingDebugged, PEB->NtGlobalFlag and PEB->Heap->ForceFlags flags to zero, and sets the PEB->Heap->Flags flags to 'HEAP_GROWABLE'. The problem with this approach is that if the debuggee's process contains a thread local storage callback, the callback will receive notification of the new thread. It can then query the thread for its entry point and see the injected code.

antiAnti also patches the debuggee's user32 EnableWindow() function code so that it simply returns. This behaviour is another bug, since the return code is never set.

The author of *antiAnti* could not be contacted.

2.2 HideDebugger

HideDebugger changes the address of the kernel32 WaitForDebugEvent() and kernel32 ContinueDebugEvent() functions in *OllyDbg*'s import address table. When a debug event occurs, *HideDebugger* sets the debuggee's PEB->BeingDebugged flag to zero. The problem with this approach is that it can be detected by malware that sets the PEB->BeingDebugged flag to a non-zero value, then calls the kernel32 IsDebuggerPresent() function. If an exception or a debug event occurs (for example, stepping

over the kernel32 IsDebuggerPresent() function call), then *HideDebugger* is revealed because the returned value will be zero.

HideDebugger changes the address of the debuggee's ntdll NtOpenProcess() function in the kernel32.dll's import address table to point to a dynamically allocated block of memory. This block contains code that prevents the *OllyDbg* process ID from being opened, which in turn prevents the *OllyDbg* process from being terminated. This redirected import can easily be identified because it does not point into ntdll.dll's image space.

HideDebugger also changes the address of the debuggee's ntdll NtQueryInformationProcess() function in the kernel32.dll's import address table to point to a dynamically allocated block of memory. This block contains code to watch for queries of the ProcessDebugPort class for the debuggee's process. When one is seen, the handle is set to zero, and then the call is allowed to proceed. This invalid handle will cause an error to be returned. The function should never return an error for the current process handle. Such an error is a sure sign that *HideDebugger* is present. The redirected import can easily be identified because it does not point into ntdll.dll's image space.

Similarly, *HideDebugger* changes the address of the debuggee's ntdll RtlRaiseException() function in the kernel32.dll's import address table to point to a dynamically allocated block of memory. This block contains code to check for the most common form of the kernel32 OutputDebugStringA() function exploit, which is a string that begins with a '%'. This does not solve the general problem, though, because the '%s' tokens can appear anywhere within the string and still cause *OllyDbg* to crash. This redirected import can easily be identified because it does not point into ntdll.dll's image space.

Finally, *HideDebugger* changes the address of the user32 SetWindowTextA() and user32 GetWindowTextA() functions in *OllyDbg*'s import address table. The idea is to prevent *OllyDbg* from changing the caption to include the 'OllyDbg' text. Instead, when the user32 SetWindowTextA() function is called for the *OllyDbg* window, the string is copied into a buffer. When the user32 GetWindowTextA() function is called for the *OllyDbg* window, the cached string is returned.

The author of *HideDebugger* could not be contacted.

2.3 HideOD

HideOD sets the debuggee's PEB->BeingDebugged, PEB->NtGlobalFlag and PEB->Heap->ForceFlags flags to zero, and sets the PEB->Heap->Flags flags to 'HEAP_GROWABLE'.

HideOD sets the debuggee's default heap head and tail values to zero, but that change is visible because the heap chunk sizes are not the expected sizes.

HideOD searches within the debuggee's kernel32 UnhandledExceptionFilter() function for the branch that is evaluated after the ProcessDebugPort is queried. There are two branch types that *HideOD* recognizes, allowing *HideOD* to support *Windows 2000* in the first case, and *Windows XP* and later in the second. There is an earlier branch that could have been patched to achieve the same result, and would have allowed for *Windows NT* support. The branch is overwritten to force the FALSE case, meaning that no debug port exists. This patch is recognizable by the '90' opcode ('NOP' instruction) after the '39' opcode ('CMP' instruction).

HideOD saves the debuggee's original ntdll NtSetInformationThread() function code to a dynamically allocated block of memory, then replaces it with the *Windows XP*-style code: MOV EDX, xxxxxxxx / CALL DWORD PTR DS:[EDX]. This change is instantly recognizable in *Windows NT* or *Windows 2000*, since the code is normally: LEA EDX, DWORD PTR SS:[ESP + 4] / INT 2E. The value that is assigned to EDX is a pointer to the dynamically allocated block of memory. This block intercepts attempts to call the ntdll NtSetInformationThread() function with the HideThreadFromDebugger class, and simply returns success in that case. The bug in this code is that if an invalid handle is passed to the function, then an error code should be returned. A successful return is an indication that *HideOD* is running.

HideOD overwrites *OllyDbg*'s kernel32 OutputDebugStringA() handler function with some code that causes it to return immediately. It appears that something more was intended, because space is allocated for a possible error code, and there is a test for the EIP register being in the upper or lower 2Gb memory space.

HideOD patches the debuggee's kernel32 Process32NextW() function code so that it always returns zero.

HideOD changes a conditional branch into a jump in the debuggee's kernel32 CheckRemoteDebuggerPresent() function. The result is that the function always sets to FALSE the value pointed to by the pbDebuggerPresent argument. The correct behaviour would have been to branch to code that returns FALSE only if the function returned successfully, and only if the current process is specified. However, the current process can be specified in ways other than the pseudo-handle that is returned by the kernel32 GetCurrentProcess() function, and that must be taken into account.

HideOD saves the debuggee's original ntdll NtQueryInformationProcess() function code to a dynamically allocated block of memory, then replaces it with the *Windows XP*-style code: MOV EDX, xxxxxxxx / CALL DWORD PTR DS:[EDX]. This change is instantly recognizable in *Windows NT* or *Windows 2000*. The value that is assigned to EDX is a pointer to the dynamically allocated block of memory. This block intercepts attempts to call the ntdll NtQueryInformationProcess() function with the ProcessDebugPort class, and tries to return zero for the port in that case. However, there is a bug in the code, which does not check if the ProcessInformation parameter points to a valid memory address, or that the entire ProcessInformationLength range is writable. If either the ProcessInformation pointer or the ProcessInformationLength is invalid, then *HideOD* will cause an exception. *OllyDbg* will trap the exception, but the debugging session will be interrupted. The correct behaviour would have been to call the original handler then zero the port only if the function returned successfully, and only if the current process is specified. However, the current process can be specified in ways other than the pseudo-handle that is returned by the kernel32 GetCurrentProcess() function, and that must also be taken into account.

The author of *HideOD* could not be contacted.

2.4 IsDebugPresent

IsDebugPresent (also known as *IsDebugExtraHide*) sets the debuggee's PEB->BeingDebugged, PEB->NtGlobalFlag and PEB->Heap->ForceFlags flags to zero. It also runs a thread which can periodically trigger the set after a specified length of time. In the same way as for *HideDebugger*, *IsDebugPresent* can be detected by setting the PEB->BeingDebugged flag to a non-zero value, then calling the IsDebuggerPresent() function after some time. *IsDebugPresent* will be revealed because the returned value will be zero.

The author of *IsDebugPresent* could not be contacted.

2.5 Olly Advanced

Olly Advanced fixes the EXCEPTION_INVALID_HANDLE (0xC0000008) exception 'bug' that occurs when, for example, an invalid handle is passed to the kernel32 CloseHandle() function while a debugger is active. The presence of a debugger causes a debug break to occur, instead of completing the kernel32 CloseHandle() function call. The fix is to patch *OllyDbg* to resume execution and allow the kernel32 CloseHandle() function call to complete as normal.

Olly Advanced forces *OllyDbg* to ignore any failure of the kernel32 TerminateProcess() function.

Olly Advanced hooks the call in *OllyDbg* to the kernel32 CreateProcess() function that creates the process for debugging. When the hook is reached, *Olly Advanced* makes the following changes:

- It searches within the debuggee's ntdll NtQuerySystemInformation() function code for the 'C2' opcode ('RET' instruction) and replaces it with an 'E9' opcode ('JMP' instruction) to point to a dynamically allocated block of memory. There are two problems with this. The first is that the search for the 'C2' opcode is done blindly, so the 'C2' that is seen might be the function index rather than the RET instruction. The second problem is that there might not be five bytes available to replace. If there are fewer than five bytes available, then *Olly Advanced* will destroy part of the following function. This can cause the debuggee to crash randomly.

  The block intercepts attempts to call the ntdll NtQuerySystemInformation() function with the SystemKernelDebuggerInformation class, and tries to return zero for the debugger information in that case. However, there is a bug in that code, which does not check if the SystemInformation parameter points to a valid memory address, or that the entire SystemInformationLength range is writable. If either the SystemInformation pointer or the SystemInformationLength is invalid, then *Olly Advanced* will cause an exception. *OllyDbg* will trap the exception, but the debugging session will be interrupted. The correct behaviour would have been to zero the debugger information only if the function returned successfully. It is also unclear why this function is intercepted, since *OllyDbg* is not a kernel-mode debugger.

- *Olly Advanced* performs the same search for the 'C2' opcode ('RET' instruction) within the debuggee's ntdll NtQueryInformationProcess() function code, and once again replaces it with an 'E9' opcode ('JMP' instruction) to point to a dynamically allocated block of memory. The same problems exist here: the search for the 'C2' opcode is done blindly, so the 'C2' that is seen might be the function index rather than the RET instruction, and if there are fewer than five bytes available, then *Olly Advanced* will destroy part of the following function.

  The block intercepts attempts to call the ntdll NtQueryInformationProcess() function with the ProcessDebugPort class, and tries to return zero for the port in that case. The block also checks if the ntdll NtQueryInformationProcess() function was called with the ProcessBasicInformation class, and tries to

replace the InheritedFromUniqueProcessId with the UniqueProcessId. However, there is a bug in both codes, which do not check if the ProcessInformation parameter points to a valid memory address, or that the entire ProcessInformationLength range is writable. If either the ProcessInformation pointer or the ProcessInformationLength is invalid for any reason, then *Olly Advanced* will cause an exception. *OllyDbg* will trap the exception, but the debugging session will be interrupted.

The correct behaviour would have been to zero the port, or perform the replacement only if the function returned successfully, and only if the current process is specified. However, the current process can be specified in ways other than the pseudo-handle that is returned by the kernel32 GetCurrentProcess() function, and that must be taken into account. Another problem with the replacement is that the process ID should never be identical to the parent process ID. Not even the EXPLORER.EXE process looks like that. Such a result is an obvious sign that *Olly Advanced* is active.

- *Olly Advanced* performs the same search again within the debuggee's ntdll NtQueryObject() function code, replacing the 'C2' opcode with an 'E9' opcode pointing to a dynamically allocated block of memory. The same problems exist here as described above.

  The block intercepts attempts to call the ntdll NtQueryObject() function with the ObjectAllTypesInformation class and tries to zero out the entire returned data. However, there is a bug in the code, which does not check if the ObjectInformation parameter points to a valid memory address, or that the entire ObjectInformationLength range is writable. If either the ObjectInformation pointer or the ObjectInformationLength is invalid, then *Olly Advanced* will cause an exception. *OllyDbg* will trap the exception, but the debugging session will be interrupted.

  It would have been better to have zeroed out the entire returned data only if the function returned successfully, but the correct behaviour would be to parse the returned data to find the DebugObject, if it exists, and then zero out the individual handle counts, but only if the function returned successfully. This arbitrary erasure is an obvious sign that *Olly Advanced* is active.

Olly Advanced hooks the code in *OllyDbg* that is reached when the debuggee's entry point is reached. When the hook is reached, *Olly Advanced* makes the following changes:

- It searches within the debuggee's kernel32 UnhandledExceptionFilter() function code for the code that retrieves the debuggee's PEB pointer. Then it searches for the '0F 84' opcode ('JE' instruction) that follows in order to fix a problem that was introduced in *Windows XP*. The problem is that *Windows XP* doesn't return immediately if the FLG_POOL_ENABLE_TAIL_CHECK (0x100) is set in the debuggee's PEB->NtGlobalFlag flags. *Olly Advanced* patches the branch so that it always returns immediately. However, the result is still that the registered exception handler will not be called because a debugger is present.
- It patches the debuggee's kernel32 Process32NextW(), kernel32 Module32Next(), kernel32 CheckRemoteDebuggerPresent() and kernel32 GetTickCount() function codes, so that they always return zero.
- It hooks the debuggee's ntdll NtSetInformationThread() function by replacing the first five bytes of the function with a relative jump to a dynamically allocated block of memory. That block intercepts attempts to call the ntdll NtSetInformationThread() function with the HideThreadFromDebugger class, and then simply returns. This behaviour is a bug, since the return code is never set.
- It patches the debuggee's kernel32 TerminateProcess() function code to cause it simply to return. This behaviour is a bug, since the return code is never set.
- It sets the debuggee's PEB->BeingDebugged flag to zero.
- It sets the debuggee's PEB->NtGlobalFlag flag to the 'HKLM\System\CurrentControlSet\Control\Session Manager\GlobalFlag' registry value. This is the least incorrect behaviour, and *Olly Advanced* appears to be the only plug-in that does this. However, it is still incomplete.

Olly Advanced hooks the code in *OllyDbg* that is reached when the debuggee's entry point break point is set. When the hook is reached, it searches within *OllyDbg*'s ntdll.dll in-memory image for a particular Thread Local Storage (TLS) callback-specific text string, then searches for the code that references that string. It places a break point in the debuggee's address space at the location after the code that prints the TLS message.

A minor bug exists in the parsing of the MZ header, though, which is that the MZ->lfanew field is assumed to be only 16 bits large. If the PE header is located more than 64KB from the start of the file, then *Olly Advanced* will access an unpredictable memory location while attempting to retrieve the PE->SizeOfImage field value, and then probably crash. However, ntdll.dll is unlikely to have such a large MZ->lfanew field value.

Olly Advanced hooks the code in *OllyDbg* that is reached when the debuggee's PE->BaseOfCode and PE->SizeOfCode fields are cached for later use. When the hook is reached, *Olly Advanced* attempts to calculate the correct sizes for the PE->BaseOfCode and PE->SizeOfCode fields. The MZ->lfanew field size bug is present here.

In this case, the MZ->lfanew field is in the debuggee's executable. We have seen viruses which append a new PE header to the host in order to defeat some heuristic detection methods. Such files can certainly have an MZ->lfanew field value in excess of 64KB. When that happens, *Olly Advanced* will read incorrect data for the PE->COFF->SizeOfOptionalHeader and PE->SizeOfImage fields and the VirtualAddress fields for the first two sections. These unpredictable values could introduce a problem that was not present before.

Olly Advanced forces *OllyDbg* to ignore the debuggee's Export Address Table if the table size appears to be too small or cannot be read completely.

Olly Advanced hooks the code in *OllyDbg* that is reached when *OllyDbg* has finished loading all plug-ins. When the hook is reached, *Olly Advanced* erases the entire Export Address Table and the Export Table Directory. This prevents detection via the kernel32 ReadProcessMemory() function to look for things like the 'ollydbg.exe' DLL name. The MZ->lfanew field size bug is present here, but *OllyDbg* is unlikely to have such a large MZ->lfanew field value.

Olly Advanced forces *OllyDbg* to ignore the data directory/OptionalHeader bug described above.

Olly Advanced hooks the code in *OllyDbg* that is reached when *OllyDbg* is formatting the kernel32 OutputDebugStringA() string. When the hook is reached, *Olly Advanced* checks if the parameter is in readable memory and skips it if not.

Olly Advanced hooks the call in *OllyDbg* to the kernel32 DebugActiveProcess() function. When the hook is reached, *Olly Advanced* touches each page in the first section of the debuggee's ntdll.dll image. The purpose of this is unclear, but it would allow the kernel32 ReadProcessMemory() function to avoid some failures. The MZ->lfanew field size bug is also present here, but ntdll.dll is unlikely to have such a large MZ->lfanew field value.

Olly Advanced sets the PEB->Heap->Flags flags to 'HEAP_GROWABLE', and sets the debuggee's PEB->Heap->ForceFlags flags to zero.

Olly Advanced patches the kernel32 SuspendThread() and user32 BlockInput() function codes to cause them simply to return. This behaviour is a bug, since the return code is never set.

The author of *Olly Advanced* responded to the bug report, saying that the *Olly Advanced* source has been available from the 'VIP' area of the Tuts4you site [5] for some time, but so far no one has worked on it. As a result, it seems unlikely that the bugs will be fixed.

2.6 OllyICE

OllyICE is a patched version of *OllyDbg*. One of the patches is reached when formatting the kernel32 OutputDebugStringA() string. The patch attempts to replace all '%' characters with ' ' in the message. However, a bug in the routine causes it to miss the last character in the string. This is probably the source of the code that is used by the plug-ins *Olly Invisible* and *Olly's Shadow*.

OllyICE ignores both the data directory/OptionalHeader bug and the entry point-zero bug described above.

Another patch is in the __fuistq() function, to avoid the floating-point operations error described in [1] by changing the value to 9.2233720368547758e+18. That is, the last three digits are removed to keep the value within bounds. However, this fix applies only to the positive value. The negative value will still crash *OllyICE*.

In the penultimate part of this article series (next month) we will look at anti-debugging tricks that target the *OllyDbg* plug-ins *Olly Invisible*, *PhantOm*, *Stealth64* and *Olly Shadow*.

The text of this paper 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] Tuts4you. http://forum.tuts4you.com/.

MALWARE ANALYSIS 1

YOUR PC IS INFECTED

Gabor Szappanos
VirusBuster, Hungary

In an article in the January issue of *Virus Bulletin* [1] I described the threats faced by an average user. One of the most significant of them was the infamous FakeAlert trojan, which is distributed via spam messages and which causes bogus malware warnings in association with rogue security software.

Meanwhile, rogue anti-virus applications and their related malware also posed one of the most significant problems for *VirusBuster*'s virus lab during the latter half of 2008.

Advanced Cleaner	MacSweeper	SpySpotter
AlfaCleaner	MalCrush 3.7	Spyware Cleaner
AntiSpyCheck 2.1	Malware Bell 3.2	Spyware Quake
AntiSpyStorm	MalwareAlarm	Spyware Stormer
AntiSpyware Shield	MalwareCore	SpywareGuard 2008
AntiSpywareExpert	MS Antivirus	SpywareStrike
AntiSpywareMaster	PAL Spyware Remover	SpyWiper
AntiSpywareSuite	PC Antispy	System anti virus 2008
Antivermins	PC Clean Pro	System Live Protect
Antivirgear	PC SpeedScan Pro	SystemDoctor
Antivirus 2008	PC-Antispyware	TheSpyBot
Antivirus 2009	PCPrivacytool	Total Secure 2009
AntiVirus Gold	PCSecureSystem	TrustedAntivirus
Antivirus Master	Perfect Cleaner	Ultimate Antivirus
Antivirus pro 2009	PersonalAntiSpy Free	UltimateCleaner
Antivirus XP 2008	PestTrap	Virus Isolator
Avatod Antispyware 8.0	PSGuard	Virus Respone Lab 2009
Awola	Rapid AntiVirus	Virus Trigger
BestsellerAntivirus	Registry Great	VirusBurst
Brave Sentry	Saliar	VirusHeat
Cleanator	SecurePCCleaner	VirusProtectPro
ContraVirus	Security toolbar 7.1	VirusRanger
Disk Knight	Smart Antivirus 2008	VirusRemover2008
Doctor Antivirus	Smart Antivirus 2009	Vista Antivirus 2008
DriveCleaner	Spy Away	WinAntiVirus Pro 2006
EasySpywareCleaner	SpyAxe	WinDefender
Errorsafe	SpyCrush	WinFixer
free-viruscan.com	Spydawn	WinSpywareProtect
IE Antivirus	SpyGuarder	WorldAntiSpy
IEDefender	SpyHeal	XP AntiSpyware 2009
InfeStop	Spylocked	XP Antivirus
Internet Antivirus	Spy-Rid	Zinaps AntiSpyware 2008
KVMSecure	SpySheriff	

Figure 1: Plausible-sounding names for scamware – is your anti-virus in this list? [3]

A ROSE BY ANY OTHER NAME

Rogue anti-virus applications, or scamware, have been around for some years now. Early instances have already been discussed in detail [2].

Authors of scamware have been very creative in selecting plausible-sounding names for their fake products. Figure 1 displays a list of some of them. One of these, 'VirusBurst', proved to be especially problematic for *VirusBuster*: many customers confused it with our product, and we had to explain time after time that it was not *our* product that kept popping up asking them for money.

Another well-known example (at least in Hungary) was an application known as 'Furkó Antivirus' (Figure 2). It had the same characteristics as contemporary scamware variants:

- A 'free' version of the application was downloadable (even from popular download repositories).
- It claimed to find a couple of virus samples (on a clean system) in files which did not even exist on the machine.
- It claimed that, for a $19.95 fee, the full version of the program could be downloaded and would remove the threats it had just identified.

It took quite some time for the unsuspecting Hungarian public to realize that there was no real product behind 'Furkó Antivirus'. Our first encounter with the application was in 2005 (although we had heard of its existence prior to that), and only in 2006 did the Hungarian media grasp hold of the fact that it was a scam.

INFILTRATION

The user's first encounter with this threat often comes via an email message which contains a download link to the

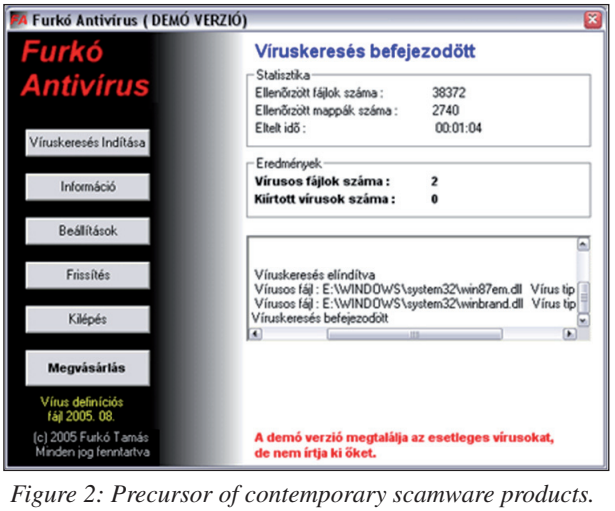


Figure 2: Precursor of contemporary scamware products.

first-stage executable. We have also seen cases of FakeAlert downloader scripts having been injected into non-malicious websites.

In the following sections I will describe the chain of events that occurs following receipt of a FakeAlert variant seeded via email[1].

Initial trigger: spammed email

As discussed in more detail in [1], the initial vector of this trojan arrives in an email message, typically disguised as an advertisement for pornographic content (Figure 3). The message contains a small amount of text and a hyperlink.

Clicking on the link leads to a web page displaying the common codec error message and offering the user the opportunity to download an update (Figure 4). This primitive form of social engineering appears to be sufficient for the trojan's needs – the websites are equipped to host code distribution kits such as Mpack, but it seems it has not proved necessary to bring in the heavy artillery.

On clicking on the 'Continue' button the fake codec is downloaded as view.exe. Once the executable has been downloaded, it deploys an arsenal of psychological warfare to scare the user into paying for the full 'product'.

View.exe – first stage downloader

View.exe is a rather simple downloader. The code is somewhat obfuscated, but its real purpose is to attempt to download three executables from a website (which at the time of writing is no longer live):

- http://79.135.167.18/scan4.exe
- http://79.135.167.18/sl32.exe
- http://79.135.167.18/gpls32.exe

The first, scan4.exe, is the next stage of the FakeAlert scam. Interestingly, sl32.exe is an Exchanger variant (aka Srizbi – see *VB*, November 2007, p.5) – I don't believe it is a coincidence that there is a connection between the spammed FakeAlert variants and one of the world's largest botnets. We have not yet found an active instance of the gpls32.exe file.

Scan4.exe – start them bugging

Scan4.exe is essentially a dropper, which drops and installs the 'nagger' component. This is dropped into the %SYSTEM% folder as 'braviax.exe' (occasionally as 'buritos.exe'), and registers for startup under:

[1] *The sequence of events may vary from case to case; steps may differ as the contents of download links change.*

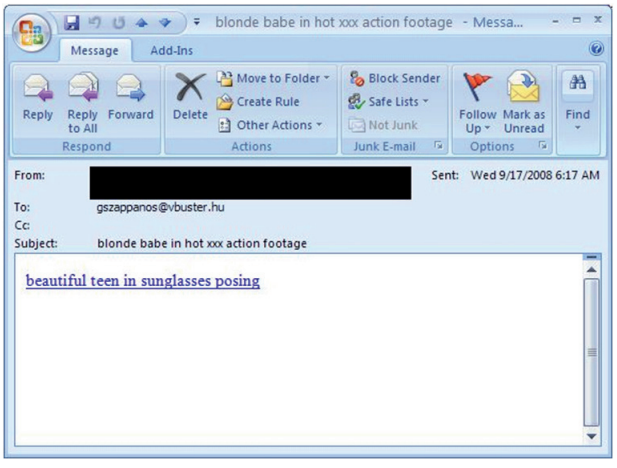


Figure 3: The bait.

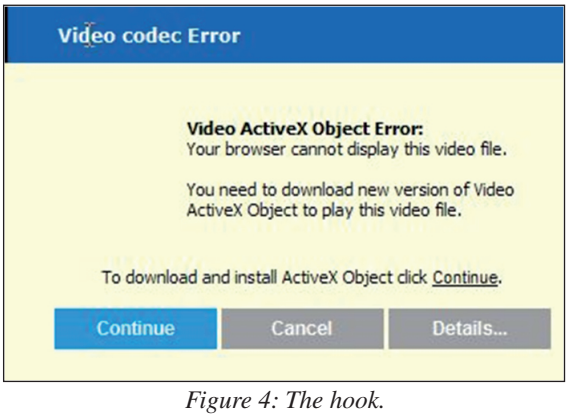


Figure 4: The hook.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run "" = C:\WINDOWS\System32\braviax.exe
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run "" = C:\WINDOWS\System32\braviax.exe

Braviax.exe opens a notify bubble on the taskbar which claims that the system is infected (Figure 5). If the user clicks on the bubble, the trojan will download the next component into %system%winivstr.exe and execute it. It uses the following URL list to download this component:

http://virus-quick-scan.com/?wmid=1062&l=12&it=2&s=1

http://antispyware-quick-scan.com/?wmid=1062&l=12&it=2&s=1

http://spyware-quickscan-2008.com/?wmid=1062&l=12&it=2&s=1

http://virus-quickscan-2008.com/?wmid=1062&l=12&it=2&s=1

http://spyware-quickscan-2009.com/?wmid=1062&l=12&it=2&s=1

Figure 5: First warning: bubble.

http://virus-quickscan-2009.com/?wmid=1062&l=12&it=2&s=1

http://antivirus-quick-scan.com/?wmid=1062&l=12&it=2&s=1

Additionally, a rootkit component is dropped onto the system as %SYSTEM%\drivers\figaro.sys (with copies named 'beep.sys' in the same folder and in the dllcache folder). This is responsible for terminating a large list of security-product-related processes, and rehooking its components (braviax.exe and karina.dat) in the registry; the first in HKLM\Software\Microsoft\Windows\CurrentVersion\Run and the second one as an Appinit_DLL, making it difficult to remove the complex. Interestingly, if buritos.exe is executed, the Run key is removed (but not the Appinit_DLL).

Scan.exe: third stage disturber

This is the point at which the real psychological warfare begins.

The downloaded third executable stage changes the desktop background to display a large malware warning dialog similar to that shown in Figure 6.

If the user falls for the scam at this point and chooses to download the full product the URL accessed will be similar to the following:

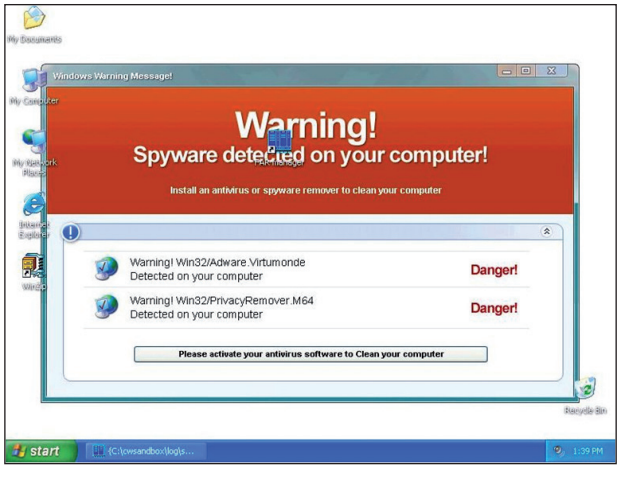


Figure 6: Second warning: your desktop is infected.

http://av-xp2008.com/images/*/3a35c64942d7aa9dec056277e50741da/*.gif

(* represents random values)

However, some users will ignore this warning, so in addition, the screensaver is set to the 'blue screen' screensaver from *Sysinternals*, which also mimics a reboot process (Figure 7).

Our support department received numerous calls from customers complaining that their computer had landed in a continuous crash-reboot stage. In reality, it was this screensaver fooling them.

A VBS script is also dropped and executed, which removes all the system restore points. This disables the trivial removal method, which would be to revert to a known safe restore point before the infection occurred.

Winivstr.exe: buy me or you are doomed

Generally, the malware warning and the blue screen prove more than sufficient to break down the user's resistance and persuade them to download the recommended application ('VirusRemover2008' at the time of collecting the material for this article, but the name of the product is subject to frequent change). Once downloaded, the application claims to find a handful of spyware and trojan instances on the system.

Needless to say, this is also the case when run on a clean, freshly installed virtual machine without Internet connectivity. So unless we believe that *Microsoft* ships trojans with its *Windows XP* installer packs (we don't), the claims are false.

In some of the scam AV product installations, legitimate anti-virus data files are downloaded from legitimate locations. Matters took an interesting turn when a company,

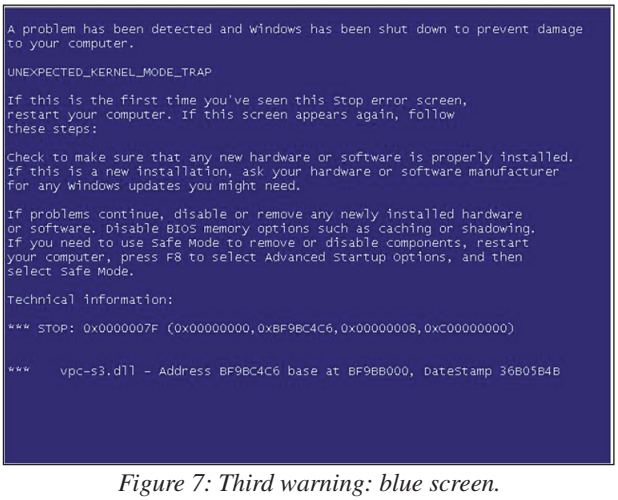


Figure 7: Third warning: blue screen.

Figure 8: Infection is found on the system.

having earlier been identified as being responsible for one of these scam attempts, approached *VirusBuster*, making enquiries about licensing our scan engine. Of course, we refused them.

From a support point of view, we receive numerous complaints as a result of the scamware. First, it 'detects' infected files on the system that our product has not detected (because they do not really exist on the system). Then, because the users get infected over and over again with the 'same' scamware (what the average user is not expected to realize is that, despite having the same application name, the executables behind the scamware change frequently). Finally, when it comes to removing the malicious files, the protecting rootkit component makes it more difficult than usual.

Fake security products and all the malicious components that go along with them to run the schemes have been appearing at a steady pace over recent months. They may not appear in the top half of our prevalence lists, but they are forever burnt into the memories of our user base.

Overall, nothing extremely malicious has happened during the process (if we ignore the fact that the infected PC has been connected to the Srizbi botnet), only a little nuisance. Nothing personal, just business.

REFERENCES

[1] Szappanos, G. A day in the life of an average user. Virus Bulletin, January 2009, p.10. http://www.virusbtn.com/pdf/magazine/2009/200901.pdf.

[2] Schouwenberg, R. The (correct) detection of light grey software. Proceedings of the Virus Bulletin International Conference, 2006.

[3] http://en.wikipedia.org/wiki/Rogue_software.

MALWARE ANALYSIS 2

THE NEW iBOTNET

Mario Ballano Barcena, Alfredo Pesoli
Symantec, Ireland

Recent weeks have seen the discovery of a new piece of malware affecting the *Apple* operating system. This article will take a detailed look at what appears to be the first real attempt to create a *Mac* botnet.

The malware variants are OSX.Iservice and OSX.Iservice.B; they are both Mach-O format universal executables designed to run on *Apple* operating systems.

The main binaries are almost identical, but what differs between them are the ways in which they are distributed and installed on the victim system.

The variants have been found inside bogus copies of *iWork '09* and *Adobe Photoshop CS4* which were shared on the popular p2p torrent network. The author of the malware downloaded the original/trial versions of each program and introduced a copy of the malicious binary into the packages. Users who then downloaded and installed the applications from the torrent download would have been infected (see Figure 1). It is estimated that thousands of people have downloaded the infected torrent files.

INSTALLATION AND DISTRIBUTION

The two variants use different techniques to obtain the user's password, which is needed in order to execute the malware with full system privileges. Note that the malware will not run if it does not have root access.

The first variant, OSX.Iservice, was bundled within the rogue *iWork '09* installer. The malware author modified the mkpg package to include the 'iWorkServices' package (the malicious executable). In this case the malware gets an authenticated session through the installer itself.

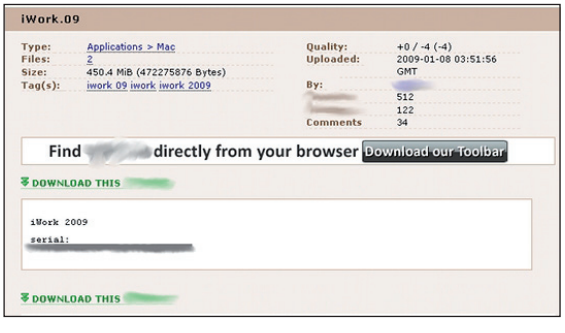


Figure 1: Users who downloaded and installed iWork '09 and Adobe Photoshop CS4 from the torrent download were probably infected by the OSX.Iservice malware family.

In the second variant, distributed through the *Adobe Photoshop CS4* torrent file, the trick is a little different. In this case the application package has not been altered. Instead, it is the crack for the application that contains the malicious binary, and once executed it will prompt the user for a password and install the trojan.

The binary that is executed by the OSX.Iservice.B variant is a dropper which performs the following operations:

1. Drops the main malware binary and executes it.
2. Asks for the user password.
3. Opens a dmg containing the real *Adobe Photoshop CS4* crack.

The main malware binary is embedded in the __data section of the dropper.

We found the trojan's second installation step very interesting: we discovered that it uses some of the internal functions of *Mac* OS that relate to the Authorization Services.

The Authorization Services APIs are used by applications in cases where certain functionalities are provided only to specific users with the relevant access rights on the system. The security agent is responsible for prompting the user for their password. It is possible to use the operating system's authentication facilities to authorize specific applications (see Figure 2).

Think about the Lock feature in System Preferences: whenever the user wants to make a change he has to provide his password. Every application has access to the Authorization Services API in order to use the OS itself for authentication. The dialog box can also be customized with an icon and a message that will be displayed before the standard text. In this case, the malware author has not customized the dialog box, and the trojan just prompts the user for the password (which may seem a suspicious request). However, with malware authors showing an

```
mov     [esp+0C48h+var_C3C], eax ; authorization - Re
                                 ; session
mov     [esp+0C48h+var_C40], 0 ; flags - kAuthorizati
mov     [esp+0C48h+var_C44], 0 ; environment - 10.3 c
                                 ; 10.4 or later you can pass
                                 ; to authorize with no user i
                                 ; 0 == kAuthorizationFlagDefa
mov     [esp+0C48h+var_C48], 0 ; rights - Pointer to
                                 ; NULL if no rights needed
call    _AuthorizationCreate ; Creating the authoriza

test    eax, eax
jnz     jmpOnFail

mov     [esp+0C48h+var_C38], 0 ; FILE **communicatior
mov     [esp+0C48h+var_C3C], 0 ; char *const *argumer
mov     [esp+0C48h+var_C40], 0 ; AuthorizationFlags c
mov     [esp+0C48h+var_C44], esi ; const char *pathTo
mov     eax, [ebp+var_1C]
mov     [esp+0C48h+var_C48], eax ; AuthorizationRef a
call    _AuthorizationExecuteWithPrivileges ; Execute
```

Figure 2: It is possible to use the OS authentication facilities to authorize specific applications.

Figure 3: The dialog box requests the user's password.

increasing interest in the *Mac* platform, we believe that more advanced UI spoofing tricks may be seen in the future.

Of course, the whole authentication part could also have been spoofed [1].

STARTUP AND ENCRYPTION CAPABILITIES

When the malware starts, it checks for the presence of its own configuration file: /var/root/.iWorkServices (or /var/root/.DivX for OSX.Iservice.B).

The malware author wasn't careful enough to remove all of the debug symbols, so some interesting strings such as '/Users/jason/diarrhea/aes/aes_modes.c' are still visible in the code. Although it is not highly sensitive information, this gives us a clue as to the possible username of the creator, the name of the project and the use of the AES algorithm for encryption capabilities.

The configuration files are encrypted using AES with a static key for encryption and decryption. The same key is also used for encrypting and decrypting network traffic. Wrappers around the 'recv' and 'send' functions allow for the specification of whether or not the traffic needs to be encrypted/decrypted (Figure 4). Some of the malware's functionalities, such as the peer-to-peer engine, make use of encryption, while others, like the http download routines, use plain text communications (for obvious reasons).

Although the use of AES makes the malware harder to analyse and will probably annoy network administrators who won't be able to identify what kind of protocol it is, it is not at all secure. We believe it is possible to create network signatures for this threat since the encryption key is fixed and some invariant packets are sent and received by the trojan.

If the malware does not detect the presence of the configuration file, a new one will be generated.

After having parsed the configuration file the malware will attempt to contact the following hosts:

- 69.92.[censored].[censored]:59201
- qwfojzlk.[censored].com:1024

```
push    ebp
mov     ebp, esp
push    edi
push    esi
push    ebx
sub     esp, 1Ch
mov     esi, [ebp+arg_0]
mov     edi, [ebp+arg_4]
mov     [esp+28h+var_1C], 0
mov     eax, [ebp+arg_8]
mov     [esp+28h+var_20], eax
mov     [esp+28h+var_24], edi
mov     eax, [esi]
mov     [esp+28h+var_28], eax
call    _recv

mov     ebx, eax
cmp     byte ptr [esi+22Ch], 0 ; check AES er
jz      short loc_4EE1

test    eax, eax
jle     short loc_4EE1

mov     [esp+28h+var_1C], eax
mov     [esp+28h+var_20], edi
mov     [esp+28h+var_24], edi
lea     eax, [esi+118h]
mov     [esp+28h+var_28], eax
call    AES_Decrypt
```

Figure 4: The malware has wrappers around the 'recv' and 'send' functions.

These hosts were probably the main core of the p2p network – the malware expects their response in order to start the p2p engine.

BOTNET AND THE P2P ENGINE

The botnet is based in a p2p network for communication, and messages exchange between all the infected nodes.

This method of communication is becoming increasingly popular with malware authors, since it provides a more secure channel to control the infected hosts than the usual 'call to home'. It also allows the network to be kept alive even if the main servers are down – and makes it more difficult to track the controllers and the infected hosts.

Unfortunately, at the time of writing this article the main servers of the botnet were down, so it was not possible to join it in order to capture and analyse the traffic between the servers and other clients.

The p2p engine is built with LUA [2]. The malware author embedded a LUA interpreter and implemented some of the malware functions with it, so it is possible to script over them, and this gives the trojan good extensibility.

Every p2p command is initially registered in a table which holds a command name and a function pointer associated with the command. These functions are registered as LUA functions, thus all the botnet commands are LUA-registered functions (Figure 5).

P2P COMMANDS

Once all the commands are registered they can be used as if they were LUA functions.

```
push    ebp
mov     ebp, esp
push    ebx
sub     esp, 14h
mov     ebx, [ebp+arg_0] ; Holds a copy of LuaState st
mov     dword ptr [esp+8], 0
mov     dword ptr [esp+4], offset p2p_cmd_socks
mov     [esp], ebx
call    reg_p2p_command_ptr

mov     dword ptr [esp+8], offset aSocks ; "socks"
mov     dword ptr [esp+4], 0FFFFD8EEh
mov     [esp], ebx
call    reg_p2p_command_name

mov     dword ptr [esp+8], 0
mov     dword ptr [esp+4], offset p2p_cmd_system
mov     [esp], ebx
call    reg_p2p_command_ptr

mov     dword ptr [esp+8], offset aSystem ; "system"
mov     dword ptr [esp+4], 0FFFFD8EEh
mov     [esp], ebx
call    reg_p2p_command_name

mov     dword ptr [esp+8], 0
mov     dword ptr [esp+4], offset p2p_cmd_httpget
mov     [esp], ebx
call    reg_p2p_command_ptr

mov     [esp+18h+var_10], offset aHttpget ; "httpget"
mov     dword ptr [esp+4], 0FFFFD8EEh
mov     [esp], ebx
call    reg_p2p_command_name
```

Figure 5: All the botnet commands are LUA-registered functions.

The following is a list of all the commands that the bot supports. Some of them are part of the peer-to-peer engine itself and have not yet been fully analysed:

Command	No. of parameters	Description
socks	1	p2p protocol-related
system	1	Executes a system command
httpget	1/2	Downloads a file
httpgeted	1/2/3	Threaded http download and executes
rand	1/2	Returns a pseudo-random number given a seed
sleep	1	Waits for an interval of time
banadd	1	p2p protocol-related
banclear	0	p2p protocol-related
p2plock	0	p2p protocol-related
p2punlock	0	p2p protocol-related
nodes	0	p2p protocol-related
leafs	0	p2p protocol-related
p2pport	0	p2p protocol-related
p2pmode	0	p2p protocol-related
p2ppeer	0	p2p protocol-related
p2ppeerport	0	p2p protocol-related
p2ppeertype	0	p2p protocol-related
set	2	Sets a parameter in the bot configuration file
get	1	Gets the value of the specified parameter from the bot configuration file
clear	1	Removes the specified parameter from the bot configuration file
p2pihistsize	0	p2p protocol-related
p2pihist	0	p2p protocol-related
platform	0	At this time returns 'OSX'

script	1	Executes a LUA script
sendlogs	2	Sends logs stored in a file named 'ff'
uptime	0	Returns the bot uptime
uid	0	Returns the bot unique identifier
shell	1	Binds a shell on the specified port
rshell	2	Connects back shell to the specified host:port

BOTNET STARTUP AND CONFIGURATION FILE STRUCTURE

The configuration file has a very simple structure:

startup\x00shell('31337')\x00p2pport\x0056620\x00

It expects a sequence of NULL-terminated byte tokens.

The startup token is used to specify a startup LUA code in the configuration file.

Only a few of the malware's internal functions are bound to LUA and, in fact, the embedded interpreter does not even support the basic functions from the standard LUA libraries. Therefore, this startup code must only be composed of botnet-registered commands which will be executed in the malware loading process.

Another reason why we suspect that the malware author is an experienced programmer, and which also makes the malware harder to analyse and debug, is that almost every part of it is multi-threaded.

CONCLUSION

OSX.Iservice is an interesting piece of malware – not only does it make use of *Mac* OS internals, but it is also the first *Mac* botnet that we are aware of. The botnet was reported to have been performing a DDoS attack through a PHP script running on the infected machines [3].

We guess that the person who wrote the malware is not the same as the person who actually 'used' it. The code indicates that, wherever possible, the author tried to use the most flexible and extendible approach when creating it – and therefore we would not be surprised to see a new, modified variant in the near future.

REFERENCES

[1] Mac OS X Dialog Box Spoofing. https://forums.symantec.com/t5/blogs/blogarticlepage/blog-id/malicious_code/article-id/240.

[2] LUA programming language. http://www.lua.org/.

[3] Hit by an OS X exploit. http://notahat.com/posts/28/.

COMPARATIVE REVIEW

WINDOWS XP SP3

John Hawes

The VB100 returns to the evergreen *Windows XP* platform this month – all but guaranteed to provide the setting for the biggest and busiest comparative of the year.

Although expectations of a large field of competitors were not disappointed, our fears of numbers potentially pushing a close to unmanageable 50 products were not realized as submissions from a number of semi-regular entrants were not forthcoming. Despite these absences, an impressive range of 39 products from 34 different vendors made the cut for the 24 February deadline, with the regular well-known brands accompanied by an interesting set of less well-known names and a handful of newcomers. Some of the newcomers hovered on the edge of meeting the requirements for qualification. In particular, the rules regarding a product's on-access functionality insist (for logistical purposes) on the ability to detect files on open or write rather than on full execution. It was decided that any product that could not be coaxed into responding to our test methodology would be excluded from the test.

With such a large and diverse field of products to test in a very limited time frame, the issue of multiple entries from single vendors posed some problems, and it became clear that it may be necessary in future to impose a small charge for vendors who wish to submit several versions of a product to the same test. This would enable us to invest in additional hardware – and potentially manpower – to cope with the testing of an ever-increasing number of products without compromising the essential free-to-all nature of the VB100 (entry of the first product would remain free of charge for every vendor). Details of any decisions we make in this direction will be made clear as part of the official VB100 procedures published on www.virusbtn.com.

This month also saw the first major set of results from our new RAP tests, which were introduced with a much smaller field of competition in the recent *Linux* test (see *VB*, February 2009, p.15). The data from this much larger set of products promised to provide some fascinating insights into many aspects of performance across the board.

PLATFORM AND TEST SETS

More than two years since the release of its successor, *Windows Vista*, more than seven years since its own first appearance, and just a few months since its official retirement from the market, *Windows XP* remains the dominant platform for computer users across the globe.

Anecdotal evidence from users in home, academic and corporate environments is backed up by usage statistics gathered from browser data on machines surfing the Internet, which show that *XP* continues to run on around 70% of desktop systems. *Vista*'s market penetration continues to increase slowly, with the platform now estimated to run on around 20% of systems. It remains to be seen if the advent of *Windows 7*, based on *Vista*'s innovations but with some considerable upgrades, will finally shake users' long-standing attachment to *XP* and herald a new era of computing.

The continued popularity of *XP* reflects its stability, simplicity and familiarity, and preparation of the test systems was a pretty straightforward task. Images used in the last test were adjusted slightly to cooperate with some minor changes in the test network, but were essentially left much as they stood. As per our standard procedures, no further updates beyond the Service Pack 3 level were added, which promised to give us some interesting results from the vulnerability detection features included in a selection of the latest generation of security suites. Otherwise, beyond tweaking the appearance and settings to fit our personal tastes, adding drivers to support the test hardware, and connecting to the lab servers to access sample and log storage, the test machines ran basic, bare and default *XP* setups.

The management of this month's test sets made for rather more work. The WildList deadline for the test was 20 February, a Friday fairly close to both the product deadline (24 February) and the usual release date of new WildLists. This caused some disquiet amongst developers anticipating a very short space of time in which to test their products against new samples added to the list. However, as it turned out, the January issue of the WildList emerged on 19 February, giving developers a little more time to make their checks.

The January WildList continued to be dominated by online gaming password stealers, and a large number of retirements from the list meant that the bulk of the items commonly seen of late, including W32/Mytob and the wide selection of network worms and bots, disappeared from the list.

Most notable among the new additions were a handful of samples representing the Conficker (aka Downadup) worm that is currently making waves around the world (see *VB*, March 2009, p.7). Breaking the monotony of simple static items was a single instance of W32/Fujacks (best known for the 'Panda burning Joss-sticks' icon that accompanied early versions). The inclusion of a file-infecting virus in the WildList set promised to provide a little extra challenge for labs, checking that they are still properly protecting against true viruses as well as the glut of more static malware.

The other test sets saw a little maintenance work as usual, with the polymorphic set having a few new items added to make up for some older items having been retired, while the trojan set was once again built from scratch using a few thousand new items gathered in the three months prior to testing. Work on the set of replicating worms and bots, which we had hoped to refresh completely in a similar manner to the trojan set, was put on the back-burner due to other priorities, but the set did undergo some expansion; we hope to find time to build a full replacement set for the next comparative.

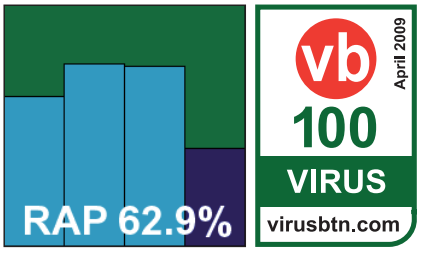
Most of the time set aside for the preparation of the test sets was devoted to building the sets for the RAP testing, with weekly sets built in the three weeks prior to the 24 February deadline and an additional set put together in the week after product updates were frozen ('week +1'). Once again we saw considerable fluctuation in the number of samples gathered in each week, but after classification and validation efforts we managed to build sets which we hoped would be suitably representative of the most prevalent malware as well as large enough to provide a good reflection of real-world performance against both known and unknown malware.

The clean test set also saw a fairly significant expansion, with updates to tracked software and a selection of new packages added. With the strict no-false-positives rule of the VB100 scheme, we endeavour to keep the clean test set as relevant as possible. However, it seems that fairly obscure false alerts – unlikely to impact many regular users – are increasingly becoming a major cause of products' failure to qualify for certification. We are investigating several options that would improve matters in this area, with one of the most important steps being the classification of clean samples according to prevalence and significance. It also seems that false positives are spreading more quickly between products these days, as automation plays a greater part in adding new detections and the samples shared between labs become polluted with clean samples. To circumvent the possibility of unscrupulous vendors exploiting this situation (by passing files known to be in our clean collection to their rivals in such a manner), we have removed from our sets several samples which have been alerted on in the past, thus ensuring that the contents of our sets remain unknown.

With everything prepared and in place a week after the product deadline, it was finally time to make a start on testing.

Agnitum Outpost Security Suite Pro 6.5.2514.381.0685

ItW	100.00%	**Polymorphic**	88.85%
ItW (o/a)	100.00%	**Trojans**	69.93%
Worms & bots	99.90%	**False positives**	0



Agnitum's *Outpost* suite has performed pretty well in our tests over the past few years, and has proved popular with the test team with its simple and clear design and stable performance. Installation took rather a long time, a particularly slow part of the process being the installation of *Microsoft* C++ libraries, but the product is a fairly complete suite including a very highly regarded firewall, so this is perhaps not too surprising. A reboot was required to complete the installation process.

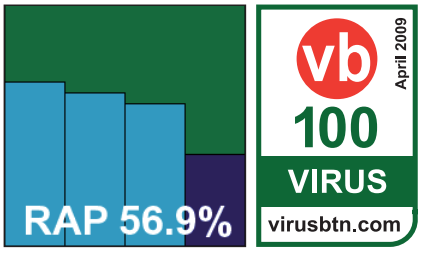
The product's interface remains unchanged, well laid out and easy to navigate. Configuration for the anti-malware component is pretty limited, but the defaults seem sensible and a decent level of protection is provided without adjustments, the on-demand scanner proving to scan much more deeply into archive types etc. than the on-access scanner. Running through the tests proved unproblematic, and results were fairly decent. Scanning speeds and overheads were mid-range, and detection rates were on the better side of average. A few polymorphic viruses were missed, and a steady if rather unimpressive catch rate was achieved across the trojan and RAP test sets, with an obvious drop in the 'week +1' set as expected. It should be noted that the product includes a plethora of additional protection measures that were not tested under our procedures – notably, the combination of firewall and HIPS protection, which would provide a better level of security than simple static detection.

The WildList presented no problems for the product, and without any false positives in the clean set *Agnitum* achieves the first VB100 award of this month's comparative.

AhnLab V3 Internet Security 7 Platinum 7.6.4.1 b.849

ItW	100.00%	**Polymorphic**	99.63%
ItW (o/a)	100.00%	**Trojans**	71.43%
Worms & bots	99.85%	**False positives**	0

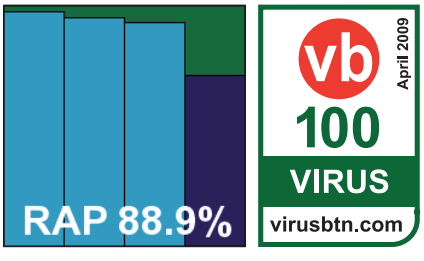
AhnLab's product offers a similar range of functionality but installed much more quickly, with fewer options to deal with and no reboot required. The interface is again clean and simple, with the emphasis firmly on the standard anti-malware side of things and the additional functions positioned less prominently. The layout was generally fairly sensible, with a few options tucked away in unexpected places, and again configuration was somewhat minimal.



Scanning speeds were not the quickest, but on-access overheads were fairly low. Detection rates were pretty average, not hugely impressive in the trojan or RAP sets and with a rather marked decrease in the unseen 'week +1' samples. However, the product has firewall and intrusion-prevention technologies (untested here) which would supplement the protection offered in a real-world situation. There were no false positives, although all *Microsoft Office* documents with macros attached were alerted on, with the product offering the option to remove the macros. The WildList was also covered without difficulty, and a VB100 is thus awarded.

Alwil avast! 4.8 Professional 4.8.1338

ItW	100.00%	**Polymorphic**	99.40%
ItW (o/a)	100.00%	**Trojans**	97.22%
Worms & bots	99.90%	**False positives**	0



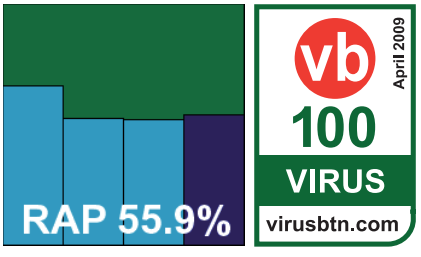
Alwil's product has been achieving some scorching detection rates in recent tests – both our own and those of other independent testing organizations – and we looked forward to seeing if these high standards could be maintained. The product's design has changed little over several years of tests, and the installation process is fairly quick and easy, but does require a reboot to complete. Although the layout has always seemed a little awkward and ungainly, the advanced version of the interface provides ample configuration options and testing ran through smoothly without incident.

Detection rates did indeed prove to be exceptional, with high levels across all our standard sets and over 90% in the first three weeks of the RAP sets. The drop in the 'week +1' test set was noticeable, but a respectable tally was achieved, and the pattern across the four weeks' worth of RAP sets was exactly what we would expect: a gradual decrease over the first three sets followed by a sharper decline as products venture into unknown territory. Scanning speeds were lightning fast, although on-access overheads were in the middle of the field. The product had no problems meeting the requirements for VB100 certification, which is duly awarded.

On-demand detection	WildList viruses		Worms & bots		Polymorphic viruses		Trojans		Clean Sets	
	Missed	%	Missed	%	Missed	%	Missed	%	FP	Susp.
Agnitum Outpost	0	100.00%	2	99.90%	191	88.85%	1925	69.93%	0	0
AhnLab V3	0	100.00%	3	99.85%	24	99.63%	1829	71.43%	0	0
Alwil avast!	0	100.00%	2	99.90%	7	99.40%	178	97.22%	0	0
Authentium Command	0	100.00%	0	100.00%	167	98.75%	1962	69.35%	0	1
AVG	0	100.00%	1	99.95%	22	99.31%	272	95.75%	0	0
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	59	99.08%	0	0
BitDefender	0	100.00%	0	100.00%	0	100.00%	383	94.02%	0	0
BullGuard	0	100.00%	0	100.00%	0	100.00%	298	95.34%	0	0
CA AV	0	100.00%	0	100.00%	860	93.83%	3206	49.91%	0	0
CA eTrust	0	100.00%	0	100.00%	860	93.83%	3216	49.76%	0	0
Check Point Zone Alarm	0	100.00%	0	100.00%	0	100.00%	444	93.06%	0	0
eEye Blink	11	99.55%	0	100.00%	205	84.22%	1198	81.28%	0	0
ESET NOD32	0	100.00%	0	100.00%	0	100.00%	302	95.28%	0	0
Filseclab Twister	53	86.85%	342	83.44%	4131	30.25%	2127	66.77%	21	4
Finport Simple	266	36.72%	732	64.55%	5099	16.47%	4814	24.79%	12	0
Fortinet FortiClient	0	100.00%	0	100.00%	4	99.66%	5989	6.44%	0	0
Frisk F-PROT	0	100.00%	0	100.00%	164	98.90%	1967	69.27%	0	0
F-Secure	0	100.00%	0	100.00%	0	100.00%	435	93.20%	0	0
G DATA	0	100.00%	0	100.00%	0	100.00%	20	99.69%	0	0
K7 Total Security	0	100.00%	4	99.81%	1404	74.94%	558	91.28%	2	0
Kaspersky	0	100.00%	0	100.00%	0	100.00%	251	96.08%	0	0
Kingsoft (Standard)	0	100.00%	15	99.27%	2814	48.30%	5635	11.97%	0	0
Kingsoft (Advanced)	0	100.00%	24	98.84%	2579	52.00%	1661	74.05%	0	0
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	611	90.45%	0	0
Microsoft Forefront	0	100.00%	0	100.00%	575	95.09%	973	84.80%	0	0
Microsoft OneCare	0	100.00%	0	100.00%	575	95.09%	1066	83.35%	0	0
MWTI eScan	4	99.01%	0	100.00%	0	100.00%	313	95.11%	0	2
Norman Security Suite	10	99.79%	0	100.00%	273	83.21%	1207	81.14%	0	0
PC Tools AV	12	99.75%	4	99.81%	3838	18.55%	4972	22.32%	0	0
PC Tools IS	12	99.75%	3	99.85%	3838	18.55%	4942	22.79%	0	0
PC Tools SD	12	99.75%	3	99.85%	3838	18.55%	4942	22.79%	0	0
Quick Heal	0	100.00%	14	99.32%	201	95.09%	857	86.61%	0	0
Redstone RedProtect	0	100.00%	0	100.00%	0	100.00%	438	93.16%	0	0
Rising IS	1	99.75%	17	99.18%	1130	70.02%	2771	56.71%	10	0
Sophos Endpoint	0	100.00%	0	100.00%	762	89.25%	1057	83.49%	0	4
Symantec Endpoint	0	100.00%	0	100.00%	5	99.96%	545	91.49%	0	0
Trustport	10	99.79%	0	100.00%	27	98.56%	352	94.50%	0	0
VirusBuster	0	100.00%	3	99.92%	191	88.85	1939	69.71%	0	0
Webroot	0	100.00%	0	100.00%	775	89.16%	1120	82.50%	0	0

Authentium Command Anti-Malware 5.0.8

ItW	100.00%	**Polymorphic**	98.75%
ItW (o/a)	100.00%	**Trojans**	69.35%
Worms & bots	100.00%	**False positives**	0



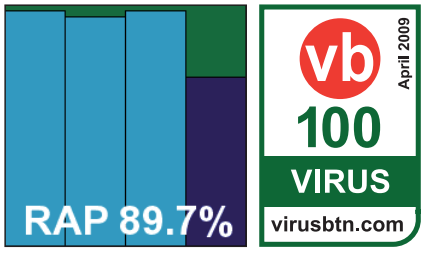
Authentium has been absent from our tests for some time now, and its product returns with a radical new interface designed using the .NET framework. Installation was a straightforward and rapid process, with a custom update system provided for our lab's unusual situation. The interface proved very simple and clearly laid out, with barely any options or configuration to trouble the user – it seemed impossible even to persuade the on-access scanner to check files with non-standard extensions. Reporting also proved rather unmanageable, but results were eventually gathered successfully after a few wrong turns signalled by figures that were way off the expected mark.

When full results were obtained, detection rates still proved rather lower than anticipated in the RAP sets. However, the product fared rather better in the standard sets – including the trojan collection, whose contents are not much older than the samples in the RAP sets and come from much the same sources. Scanning speeds were less than brilliant, but overheads were very reasonable. Nothing was missed in the WildList set, and a single item in the clean set that was alerted on with a vague level of suspicion was adjudged insufficient to prevent *Command* from winning a VB100 award.

AVG 8.0 b 237

ItW	100.00%	**Polymorphic**	99.31%
ItW (o/a)	100.00%	**Trojans**	95.75%
Worms & bots	99.95%	**False positives**	0

AVG's latest iteration includes yet more of the additional functionalities the company seems to be buying in at great speed of late. The design is as professional as ever, with a reasonably fast installation process followed by a 'first run wizard' to set some basic configuration options, followed by a reboot. The interface features an over-abundance of status icons, some of them apparently overlapping or of rather exaggerated significance, but tunnelling down to the advanced options proved no problem and everything we needed was readily to hand.

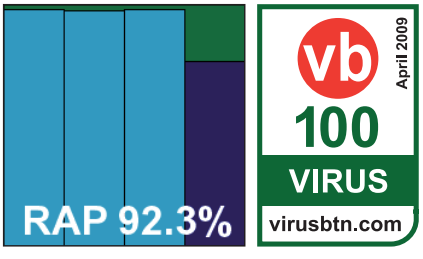


Both scanning speeds and overheads were around the middle of the pack, but detection rates were excellent, missing an overall average of 90% in the RAP sets by just a whisker. The product is another full security suite that provides a range of additional features, including the famous *LinkScanner* as well as the more standard likes of firewall, intrusion prevention, mail and web filters and much else besides, so real-world protection levels are likely to be even higher.

The product encountered no problems in detecting all samples in the WildList set, and generated no false positives in the clean sets, and as a result *AVG* achieves another VB100 award.

Avira AntiVir Professional 8.2.0.612

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	99.08%
Worms & bots	100.00%	**False positives**	0



Avira's product is another which has put in some truly remarkable performances over the last few years, and it continues to excel in a number of independent measures. With the bar for the new RAP tests already set pretty high, we looked forward to another likely candidate to push the bar and set the pace.

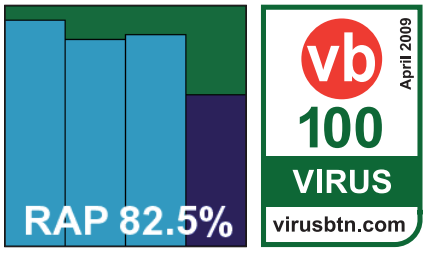
The product has changed little outwardly over the past few years, remaining adorned with friendly faces carrying red umbrellas, and featuring the occasional oddity of layout or syntax but generally proving simply laid out and responsive.

Running through the tests proved a simple process given the ample configuration options and very sensible defaults, and both scanning speeds and on-access overheads were excellent. Detection rates, as hoped, were similarly superlative, with very little missed anywhere. A more than decent score in the RAP 'week +1' set pushed the product's average RAP score to over 90% – the first product to achieve this milestone this month and likely to be one of very few to do so. With nothing to trouble the product in the

clean or WildList sets, a VB100 award is earned along with considerable respect.

BitDefender Total Security 2009 12.0.11.5

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	94.02%
Worms & bots	100.00%	**False positives**	0



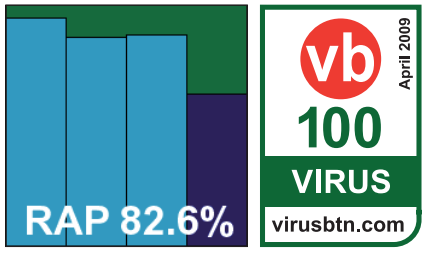
BitDefender returns after a brief absence from VB100 tests, with yet another revamping of the product's interface to reflect some significant changes under the hood. The installation process took a little time, but the new interface looked pretty good, with a nice simple version displaying status information accompanied by an advanced option with more detailed controls.

Scanning speeds were a little below expectation, but on-access overheads were very reasonable, and detection rates decent. Excellent scores were achieved in the standard sets and most of the RAP sets, and only an average-sized decrease in the 'week +1' set brought the product's RAP score down. Yet again, a wide range of additional protection levels are offered by the product, notable amongst which are a vulnerability monitor to check for out-of-date software and the data leak prevention options. The product encountered no problems in the WildList set, and with no problems in the clean sets either a VB100 is well earned.

BullGuard 8.5

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	95.34%
Worms & bots	100.00%	**False positives**	0

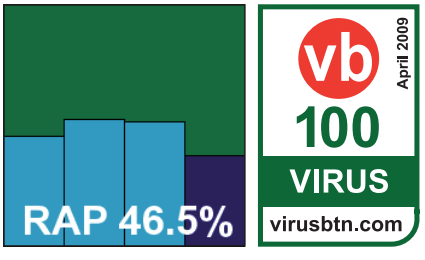
BullGuard's product seems to be making increasing inroads into various markets, thanks not least to free trials coming pre-installed on an impressive range of new hardware. Using the *BitDefender* engine, we expected similar scores and performance. Installation of the product was certainly similarly languorous, and included the rare offer to remove any potentially clashing competitive software. A reboot was required to complete the process. Initially, the product appeared to be misbehaving somewhat, and while a second reboot fixed some on-access issues, the interface frequently proved unresponsive, taking long pauses before responding even under normal activity levels. Logging and selection of post-scan options also proved a little awkward.



Detection rates, however, were excellent – actually showing a fractional improvement on those achieved by the *BitDefender* product, implying that *BullGuard* has either added some extra heuristics of its own or is using slightly stricter settings by default. Once again, the WildList caused the product no problems, and the clean sets likewise, thus securing a VB100 award for *BullGuard*.

CA Anti-Virus 10.0.0.169

ItW	100.00%	**Polymorphic**	93.83%
ItW (o/a)	100.00%	**Trojans**	49.91%
Worms & bots	100.00%	**False positives**	0



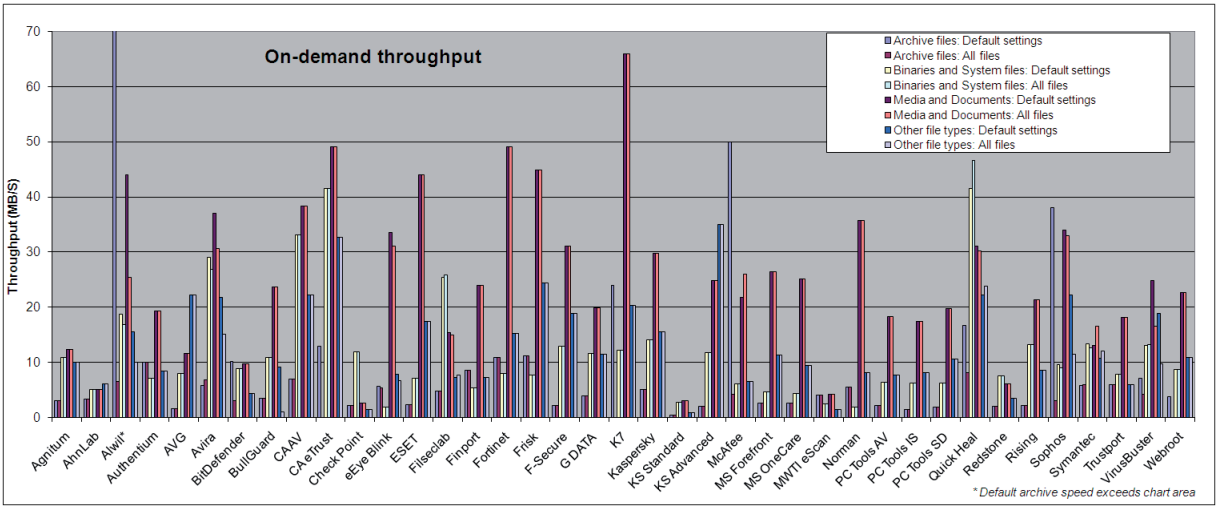
CA's home-user product has proved fairly reliable in recent tests, providing reasonable detection rates coupled with outstanding scanning speeds. Here the product remains little changed, although it surprised us somewhat during installation with an unavoidable attempt to update and with the proposal to install a *Yahoo! Toolbar*. A reboot was required to get things up and running. The interface itself remains clear and simple, with a fairly standard layout making for good usability. As expected, configuration was limited to little more than on or off, but scanning speeds and overheads were every bit as excellent as hoped.

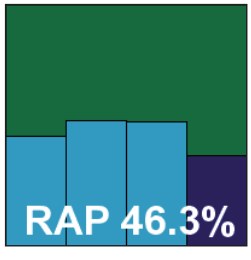
Detection rates lagged a little behind the curve, with stable but disappointing detection rates across the trojans and RAP sets. Elsewhere things were a little better, and with no issues in the WildList or clean sets a VB100 certification is awarded.

CA eTrust Anti-Virus 8.1.637.0

ItW	100.00%	**Polymorphic**	93.83%
ItW (o/a)	100.00%	**Trojans**	49.76%
Worms & bots	100.00%	**False positives**	0

On-access detection	WildList viruses		Worms & bots		Polymorphic viruses		Trojans		Clean sets	
	Missed	%	Missed	%	Missed	%	Missed	%	FP	Susp.
Agnitum Outpost	0	100.00%	2	99.90%	191	88.85%	1967	69.27%	0	0
AhnLab V3	0	100.00%	11	99.47%	24	99.63%	1833	71.36%	0	0
Alwil avast!	0	100.00%	2	99.90%	7	99.40%	173	97.30%	0	0
Authentium Command	0	100.00%	0	100.00%	167	98.75%	1977	69.11%	0	1
AVG	0	100.00%	1	99.95%	22	99.31%	272	95.75%	0	0
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	61	99.05%	0	0
BitDefender	0	100.00%	0	100.00%	0	100.00%	332	94.81%	0	0
BullGuard	0	100.00%	11	99.47%	0	100.00%	299	95.33%	0	0
CA AV	0	100.00%	0	100.00%	860	93.83%	3214	49.79%	0	0
CA eTrust	0	100.00%	0	100.00%	860	93.83%	3216	49.76%	0	0
Check Point Zone Alarm	0	100.00%	0	100.00%	0	100.00%	619	90.33%	0	0
eEye Blink	11	99.55%	0	100.00%	555	79.88%	1311	79.52%	0	0
ESET NOD32	0	100.00%	0	100.00%	0	100.00%	251	96.08%	0	0
Filseclab Twister	53	86.85%	373	81.94%	4131	30.25%	2221	65.30%	8	0
Finport Simple	266	36.72%	756	63.39%	5099	16.47%	4814	24.79%	12	0
Fortinet FortiClient	0	100.00%	0	100.00%	4	99.66%	5984	6.51%	0	0
Frisk F-PROT	0	100.00%	0	100.00%	164	98.90%	1976	69.13%	0	0
F-Secure	0	100.00%	0	100.00%	0	100.00%	589	90.80%	0	0
G DATA	0	100.00%	0	100.00%	0	100.00%	31	99.52%	0	0
K7 Total Security	0	100.00%	4	99.81%	1593	71.33%	595	90.70%	2	0
Kaspersky	0	100.00%	0	100.00%	0	100.00%	628	90.19%	0	0
Kingsoft (Standard)	0	100.00%	17	99.18%	2814	48.30%	5665	11.50%	0	0
Kingsoft (Advanced)	0	100.00%	27	98.69%	2579	52.00%	1740	72.82%	0	0
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	526	91.78%	0	0
Microsoft Forefront	0	100.00%	0	100.00%	575	95.09%	1118	82.53%	0	0
Microsoft OneCare	0	100.00%	0	100.00%	575	95.09%	1124	82.44%	0	0
MWTI eScan	4	99.01%	0	100.00%	0	100.00%	315	95.08%	0	0
Norman Security Suite	10	99.79%	0	100.00%	350	81.68%	1320	79.38%	0	0
PC Tools AV	12	99.75%	13	99.37%	3838	18.55%	5266	17.73%	0	0
PC Tools IS	12	99.75%	11	99.47%	3838	18.55%	5103	20.28%	0	0
PC Tools SD	12	99.75%	11	99.47%	3838	18.55%	5103	20.28%	0	0
Quick Heal	0	100.00%	14	99.32%	201	95.09%	1835	71.33%	0	0
Redstone RedProtect	0	100.00%	0	100.00%	0	100.00%	633	90.11%	0	0
Rising IS	1	99.75%	14	99.32%	1212	66.36%	3006	53.04%	10	0
Sophos Endpoint	0	100.00%	0	100.00%	762	89.25%	1057	83.49%	0	3
Symantec Endpoint	0	100.00%	0	100.00%	5	99.96%	491	92.33%	0	0
Trustport	10	99.79%	0	100.00%	27	98.56%	352	94.50%	0	0
VirusBuster	0	100.00%	3	99.92%	191	88.85%	2012	68.57%	0	0
Webroot	0	100.00%	0	100.00%	775	89.16%	1146	82.10%	0	0



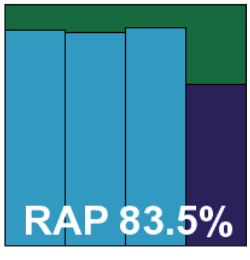


The corporate offering from *CA* has long been something of a bugbear in the VB100, its interface being approached with distaste and dread. The installation process, featuring numerous lengthy EULAs, is as tedious as ever, and the web-style interface (designed for corporate management no doubt) is awkward, fiddly, occasionally opaque, and often extremely slow to respond. Configuration is reasonably ample, although in some cases – such as adjusting archive scanning levels – proves not to react as expected.

Logging is also a little tricky to handle, with the on-screen displays not suited to handling more than a handful of issues at a time, but here experience helps, and our tried and tested techniques to extract data from their obscure format paid off. Once gathered, results showed the expected excellent scanning speeds in both modes. As in the home-user product, detection rates left much to be desired, but the product met all the requirements to achieve VB100 certified status. An award is granted, but a long overdue revamp of the front end remains high on our wish list.

Check Point Zone Alarm Extreme Security 8.0.298.000

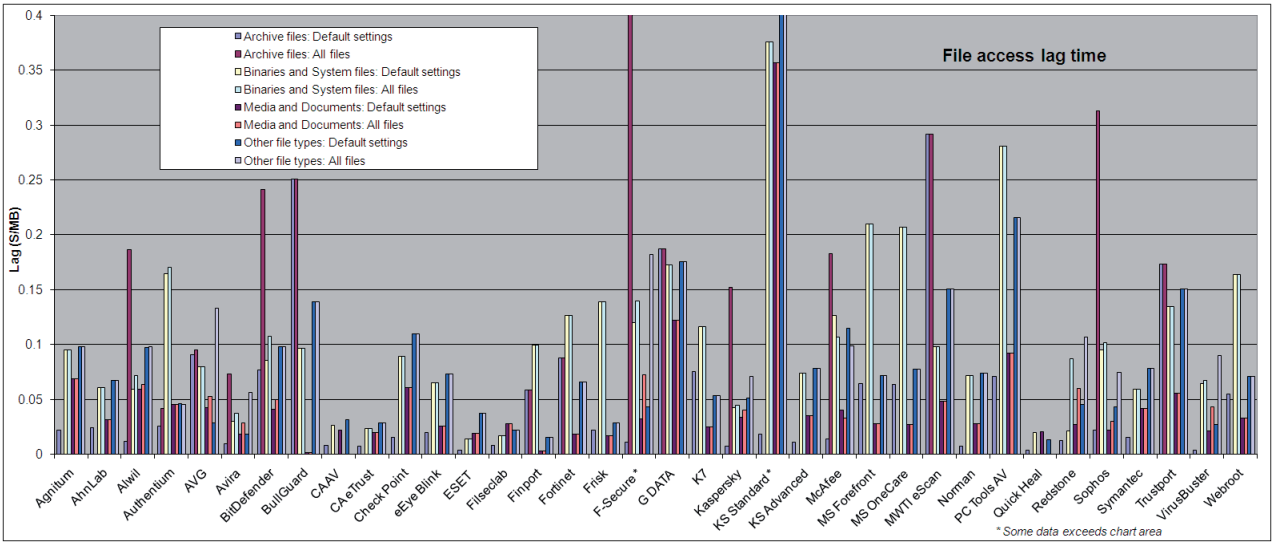
ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	93.06%
Worms & bots	100.00%	**False positives**	0



Zone Alarm has only been entered for VB100 testing once before (see *VB*, April 2008, p.13). The initial installation process presented a few difficulties, with the basic package little more than a downloader for the installer proper. To accommodate the unusual submission style at short notice, the product was installed on a test system on the deadline date and updated online, with a dedicated image taken for later testing. However, it emerged that the 'update' button on the front page of the interface – which responded with a message claiming that the product was up to date – had not, in fact, functioned properly, as actioning a separate update within the anti-malware section of the product produced a much longer process and considerably higher version number. Updates were thus applied manually to one of the numerous folders sprinkled by the product around the system.

Scanning was also a little unconventional, with no clear option for manual scanning in the main interface; on-demand tests were thus performed using a combination of right-click scanning and scheduling. As the 'extreme' of the product title suggests, scanning was pretty thorough, which was reflected in rather slow on-demand scanning speeds, but on-access overheads were not unreasonable and detection rates were for the most part superb, thanks in part to the *Kaspersky* engine included in the product. The 'week +1' results in the RAP test showed a rather steeper downturn than average, from a very high starting point, but the product includes a wide range of extra protection features,

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	995	3.06	995	3.06	241	10.84	241	10.84	171	12.34	171	12.34	98	9.98	98	9.98
AhnLab V3	911	3.34	911	3.34	512	5.10	512	5.10	418	5.05	418	5.05	161	6.08	161	6.08
Alwil avast!	30	101.55	464	6.57	139	18.80	155	16.86	48	43.96	83	25.42	63	15.53	97	10.09
Authentium Command	303	10.05	303	10.05	369	7.08	369	7.08	109	19.36	109	19.36	116	8.43	116	8.43
AVG	1890	1.61	1890	1.61	326	8.02	326	8.02	182	11.59	182	11.59	44	22.23	44	22.23
Avira AntiVir	528	5.77	442	6.89	90	29.03	97	26.94	57	37.02	69	30.58	45	21.74	65	15.05
BitDefender	298	10.22	978	3.12	293	8.92	293	8.92	216	9.77	216	9.77	226	4.33	226	4.33
BullGuard	870	3.50	870	3.50	241	10.84	241	10.84	89	23.71	89	23.71	107	9.14	1007	0.97
CA AV	436	6.99	436	6.99	79	33.08	79	33.08	55	38.36	55	38.36	44	22.23	44	22.23
CA eTrust	235	12.96	NA	NA	63	41.48	63	41.48	43	49.07	43	49.07	30	32.61	30	32.61
Check Point Zone Alarm	1406	2.17	1406	2.17	220	11.88	220	11.88	820	2.57	820	2.57	680	1.44	680	1.44
eEye Blink	544	5.60	564	5.40	1387	1.88	1439	1.82	63	33.49	68	31.03	124	7.89	147	6.66
ESET NOD32	1306	2.33	1306	2.33	367	7.12	367	7.12	48	43.96	48	43.96	56	17.47	56	17.47
Filseclab Twister	633	4.81	643	4.74	103	25.37	101	25.87	137	15.40	141	14.96	134	7.30	127	7.70
Finport Simple	357	8.53	357	8.53	492	5.31	492	5.31	88	23.98	88	23.98	135	7.25	135	7.25
Fortinet FortiClient	278	10.96	278	10.96	325	8.04	325	8.04	43	49.07	43	49.07	64	15.29	64	15.29
Frisk F-PROT	273	11.16	273	11.16	337	7.75	337	7.75	47	44.89	47	44.89	40	24.46	40	24.46
F-Secure	1423	2.14	1423	2.14	202	12.94	202	12.94	68	31.03	68	31.03	52	18.81	52	18.81
G DATA	783	3.89	783	3.89	226	11.56	226	11.56	106	19.91	106	19.91	85	11.51	85	11.51
K7 Total Security	127	23.99	NA	NA	215	12.15	215	12.15	32	65.94	32	65.94	48	20.38	48	20.38
Kaspersky	595	5.12	595	5.12	186	14.05	186	14.05	71	29.72	71	29.72	63	15.53	63	15.53
Kingsoft (Standard)	7788	0.39	7788	0.39	970	2.69	970	2.69	707	2.98	707	2.98	1220	0.80	1220	0.80
Kingsoft (Advanced)	1505	2.02	1505	2.02	223	11.72	223	11.72	85	24.82	85	24.82	28	34.94	28	34.94
McAfee VirusScan	61	49.94	731	4.17	424	6.16	425	6.15	97	21.75	81	26.05	149	6.57	150	6.52
Microsoft Forefront	1153	2.64	1153	2.64	559	4.67	559	4.67	80	26.37	80	26.37	86	11.38	86	11.38
Microsoft OneCare	1146	2.66	1146	2.66	595	4.39	595	4.39	84	25.12	84	25.12	104	9.41	104	9.41
MWTI eScan	749	4.07	749	4.07	1052	2.48	1052	2.48	502	4.20	502	4.20	652	1.50	652	1.50
Norman Security Suite	558	5.46	558	5.46	1428	1.83	1428	1.83	59	35.76	59	35.76	121	8.09	121	8.09
PC Tools AV	1369	2.23	1369	2.23	410	6.37	410	6.37	115	18.35	115	18.35	128	7.64	128	7.64
PC Tools IS	2063	1.48	2063	1.48	423	6.18	423	6.18	121	17.44	121	17.44	120	8.15	120	8.15
PC Tools SD	1672	1.82	1672	1.82	417	6.27	417	6.27	107	19.72	107	19.72	92	10.63	92	10.63
Quick Heal	183	16.65	373	8.17	63	41.48	56	46.66	68	31.03	70	30.14	44	22.23	41	23.86
Redstone RedProtect	1536	1.98	1536	1.98	347	7.53	347	7.53	346	6.10	346	6.10	286	3.42	286	3.42
Rising IS	1410	2.16	1410	2.16	198	13.20	198	13.20	99	21.31	99	21.31	115	8.51	115	8.51
Sophos Endpoint	80	38.08	1010	3.02	274	9.54	291	8.98	62	34.03	64	32.97	44	22.23	85	11.51
Symantec Endpoint	520	5.86	507	6.01	196	13.33	207	12.62	162	13.02	128	16.48	91	10.75	81	12.08
Trustport	512	5.95	512	5.95	332	7.87	332	7.87	116	18.19	116	18.19	165	5.93	165	5.93
VirusBuster	431	7.07	733	4.16	201	13.00	197	13.26	85	24.82	128	16.48	52	18.81	101	9.69
Webroot	801	3.80	NA	NA	302	8.65	302	8.65	93	22.69	93	22.69	90	10.87	90	10.87

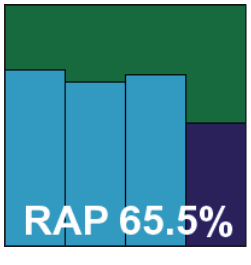


including advanced firewall and intrusion prevention technologies, which should go some way to improving matters in this area.

The WildList and clean sets presented no difficulties, and *Check Point*'s solid product earns its second VB100 award with its head held high.

eEye Digital Security Blink Professional 4.2.4.2076

ItW	99.55%	**Polymorphic**	84.22%
ItW (o/a)	99.55%	**Trojans**	81.28%
Worms & bots	100.00%	**False positives**	0

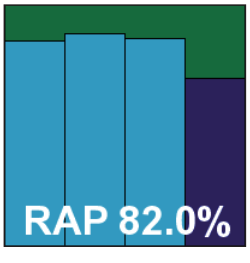


Blink is another semi-regular participant in our comparatives, with a good record in past tests and a reputation in our lab for combining impressive completeness of features with admirable clarity of design and usability. The installation process is lengthy but informative, and no reboot is required to complete, but many of the protection features appear to be disabled by default. This is not the case with the anti-malware portions, fortunately, which have a reasonable level of configuration in an interface which must ration space between numerous modules, notably vulnerability monitoring.

Scanning speeds were pretty good, with equally impressive on-access overheads, although scanning of large numbers of executables on demand did take some time thanks to the use of the *Norman Sandbox* technology. Detection rates were generally reasonable, with performance increasing notably with the age of samples. False positives were absent, but in the WildList set the selection of W32/Fujacks samples were missed, thus denying *eEye* aVB100 award this time.

ESET NOD32 3.0.684.0

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	95.28%
Worms & bots	100.00%	**False positives**	0



ESET's *NOD32* has long been a top performer in the VB100 and still holds the record for the largest number of certifications earned. The product has become considerably more stylish and user-friendly in recent years, but in some measures has lost its long-held lead in terms of both speed and detection rates, with some similarly excellent rivals catching up. The latest version is as slick and attractive as ever, and installation is a pleasant experience despite the occasional unexpected pause. Similar pauses were observed occasionally during scanning, particularly when handling large infected test sets, but such situations are vanishingly rare in the real world.

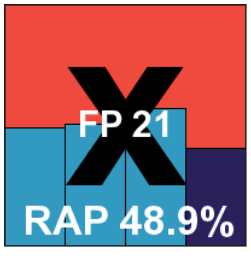
Scanning speeds and overheads over more normal types of data proved as excellent as ever – no longer way ahead of the field perhaps, but certainly among the very best. Detection rates were also excellent – again, not quite at the top of the heap, but putting in a very strong showing, with

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	68	0.02	NA	NA	261	0.10	261	0.10	166	0.07	166	0.07	112	0.10	112	0.10
AhnLab V3	76	0.02	NA	NA	171	0.06	171	0.06	88	0.03	88	0.03	81	0.07	81	0.07
Alwil avast!	38	0.01	570	0.186	166	0.06	200	0.07	145	0.06	154	0.06	111	0.10	112	0.10
Authentium Command	81	0.03	130	0.042	442	0.16	458	0.17	117	0.05	116	0.05	61	0.05	60	0.05
AVG	277	0.09	292	0.095	219	0.08	219	0.08	110	0.04	132	0.05	44	0.03	146	0.13
Avira AntiVir	32	0.01	224	0.073	91	0.03	110	0.04	60	0.02	82	0.03	34	0.02	71	0.06
BitDefender	236	0.08	738	0.24	235	0.09	294	0.11	107	0.04	126	0.05	127	0.11	112	0.10
BullGuard	766	0.25	766	0.251	264	0.10	264	0.10	24	0.00	24	0.00	152	0.14	152	0.14
CA AV	27	0.01	NA	NA	81	0.03	NA	NA	67	0.02	NA	NA	46	0.03	NA	NA
CA eTrust	24	0.01	NA	NA	73	0.02	73	0.02	63	0.02	63	0.02	44	0.03	44	0.03
Check Point Zone Alarm	49	0.02	NA	NA	246	0.09	246	0.09	148	0.06	148	0.06	123	0.11	123	0.11
eEye Blink	62	0.02	NA	NA	183	0.07	183	0.07	74	0.03	74	0.03	87	0.07	87	0.07
ESET NOD32	12	0.00	NA	NA	48	0.01	48	0.01	60	0.02	60	0.02	52	0.04	52	0.04
Filseclab Twister	26	0.01	NA	NA	56	0.02	56	0.02	80	0.03	80	0.03	38	0.02	38	0.02
Finport Simple	181	0.06	181	0.059	271	0.10	271	0.10	28	0.00	28	0.00	31	0.02	31	0.02
Fortinet FortiClient	270	0.09	270	0.088	342	0.13	342	0.13	59	0.02	59	0.02	80	0.07	80	0.07
Frisk F-PROT	70	0.02	NA	NA	374	0.14	374	0.14	56	0.02	56	0.02	43	0.03	43	0.03
F-Secure	36	0.01	1555	0.510	325	0.12	377	0.14	89	0.03	173	0.07	58	0.04	194	0.18
G DATA	573	0.19	573	0.187	463	0.17	463	0.17	279	0.12	279	0.12	187	0.18	187	0.18
K7 Total Security	232	0.08	NA	NA	316	0.12	316	0.12	73	0.02	73	0.02	68	0.05	68	0.05
Kaspersky	25	0.01	466	0.152	122	0.04	129	0.04	91	0.03	106	0.04	66	0.05	85	0.07
Kingsoft (Standard)	58	0.02	NA	NA	995	0.38	995	0.38	773	0.36	773	0.36	1237	1.25	1237	1.25
Kingsoft (Advanced)	36	0.01	NA	NA	205	0.07	205	0.07	95	0.04	95	0.04	92	0.08	92	0.08
McAfee VirusScan	44	0.01	560	0.183	342	0.13	291	0.11	106	0.04	90	0.03	128	0.11	113	0.10
Microsoft Forefront	199	0.06	NA	NA	561	0.21	561	0.21	79	0.03	79	0.03	86	0.07	86	0.07
Microsoft OneCare	197	0.06	NA	NA	553	0.21	553	0.21	77	0.03	77	0.03	92	0.08	92	0.08
MWTI eScan	891	0.29	891	0.292	268	0.10	268	0.10	123	0.05	123	0.05	163	0.15	163	0.15
Norman Security Suite	25	0.01	NA	NA	198	0.07	198	0.07	79	0.03	79	0.03	88	0.07	88	0.07
PC Tools AV	218	0.07	NA	NA	745	0.28	745	0.28	215	0.09	215	0.09	227	0.22	227	0.22
Quick Heal	13	0.00	NA	NA	64	0.02	NA	NA	64	0.02	NA	NA	29	0.01	NA	NA
Redstone RedProtect	40	0.01	NA	NA	68	0.02	239	0.09	77	0.03	147	0.06	60	0.05	120	0.11
Sophos Endpoint	69	0.02	956	0.313	260	0.09	278	0.10	67	0.02	84	0.03	58	0.04	89	0.07
Symantec Endpoint	50	0.02	NA	NA	166	0.06	166	0.06	109	0.04	109	0.04	92	0.08	92	0.08
Trustport	529	0.17	529	0.173	363	0.13	363	0.13	138	0.06	137	0.06	163	0.15	163	0.15
VirusBuster	14	0.00	NA	NA	180	0.06	188	0.07	65	0.02	112	0.04	42	0.03	104	0.09
Webroot	169	0.05	NA	NA	441	0.16	441	0.16	90	0.03	90	0.03	85	0.07	85	0.07

a much lower drop in the 'week +1' RAP set than most. With the product encountering no problems meeting the requirements for VB100 certification, *ESET* adds another award to its sizeable collection.

Filseclab Twister AntiVirus 7.3.2.9971

ItW	86.85%	**Polymorphic**	30.25%
ItW (o/a)	86.85%	**Trojans**	66.77%
Worms & bots	83.44%	**False positives**	21



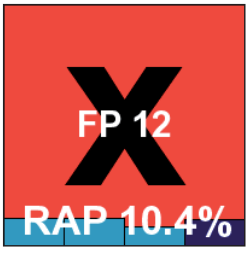
The first of the newcomers in this month's test, *Filseclab*'s *Twister* has picked up a bit of a reputation as a strong up-and-comer on various web forums and discussion boards, and has put in some excellent performances in independent tests run in China. An initial trial version we looked at impressed us with simplicity, stability and better than expected scanning performance, and a later version submitted for the test showed even more promise. With a slick and professional-looking installation process and a clear, attractive and well laid-out interface, the product certainly looks the business and has a very good level of fine-tuning available, as well as a behavioural monitoring system that is given as much importance as the more traditional detection in the layout of the interface.

Running through the tests proved a little less straightforward than hoped thanks to some slightly unusual behaviour: on-access scanning, while triggered on read, seemed not to block access instantly, instead waiting a little before alerting on and taking action against detected items. This meant that our standard opener tool, which logs items it cannot access, recorded having successfully opened everything. Thus, detection data could only be gathered from the product's own logs and the on-access scanning speeds, recorded in the same manner, may not quite reflect the full picture.

Detection rates were not unreasonable, particularly for a product that is entirely new to our testing system and test sets. Fairly good scores were achieved in some of the standard sets, including a surprisingly excellent handling of W32/Virut samples in the polymorphic set, with a little less coverage of older polymorphic items, and a fairly decent showing in the trojan and RAP sets. Several items in the WildList set were not covered, most of which were from the latest batch of additions, and a sprinkling of false alarms were raised in the clean sets (no big surprise on the product's first look at their diverse content), so *Twister* does not qualify for a VB100 award on its first attempt, but it looks like being a strong contender in the very near future.

Finport Simple Anti-virus 4.2.30

ItW	36.72%	**Polymorphic**	16.47%
ItW (o/a)	36.72%	**Trojans**	24.79%
Worms & bots	64.55%	**False positives**	12

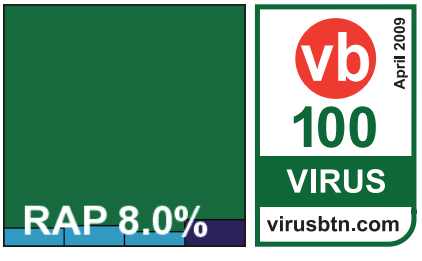


A second new product, this one emerging from the Ukraine and considerably newer on the scene, *Simple* lives up to its name in both its installation process and GUI, which uses the .NET framework and presents all the basic requirements in a very clear, easy-to-use manner. Bright, cheery, uncluttered and easy to navigate, the product stood up very well under the pressure of our tests, which can cause problems for much more seasoned solutions, running solidly and stably throughout.

Scanning speeds were pretty respectable, but detection rates still need a lot of work – which is not surprising for a product so very new to the scene. A smattering of false positives, along with quite a few misses in the WildList, deny *Finport* a VB100 this time, but the company's highly usable product will be very welcome in future tests, and we hope that with some work on detection levels it should soon reach the required standard for VB100 qualification.

Fortinet FortiClient 3.0.614

ItW	100.00%	**Polymorphic**	99.66%
ItW (o/a)	100.00%	**Trojans**	6.44%
Worms & bots	100.00%	**False positives**	0



Fortinet's desktop product has a much longer history in our tests, and has changed little since I first encountered it some years ago. The layout is serious and professional, with a number of additional protection features provided in a clean and uncluttered interface covering the wide range of configuration options required in corporate environments.

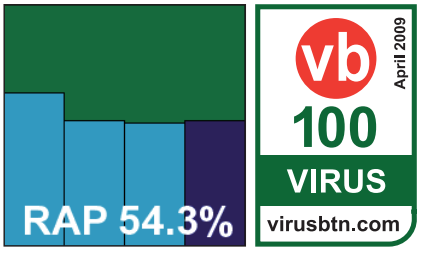
Scanning speeds and overheads were both excellent, and detection rates in our traditional test sets have long proved highly accomplished, but the addition of the new trojan sets in recent tests has highlighted some problems, and the low scores are repeated in the RAP sets here. The addition of optional 'grayware' scanning was tested – the absence of

which has been cited in previous tests as a possible reason for the low scores. The use of this scanning option did result in a small improvement over the rates recorded with the default settings, and enabling the 'heuristic' option (also disabled by default in the submitted product) increased detection rates substantially, to around 70% across the trojan and RAP sets. However, the vast majority of the additional detections were marked only as 'suspicious' – a tag which would not be counted as a full detection if this option were to be tested as part of the default settings.

Thankfully for *Fortinet*, no problems were encountered in the core certification test sets, with the product achieving full detection of samples in the WildList and generating no false positives in the clean sets. A VB100 award is duly granted.

Frisk F-PROT Anti-Virus 6.0.9.1

ItW	100.00%	**Polymorphic**	98.90%
ItW (o/a)	100.00%	**Trojans**	69.27%
Worms & bots	100.00%	**False positives**	0



Frisk's product remains a very simple and straightforward one, with few frills, minimal configuration and no extras beyond the basic requirements of anti-malware scanning and on-access protection.

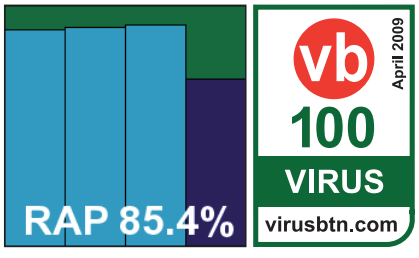
The installation process took a little longer than expected, with a long pause at the 'preparing to install' stage, and on several occasions during testing some stability issues were noted, both in general use of the interface and while running scans. On a few occasions the product generated error messages, but in most cases scanning or protection seemed to continue nevertheless.

Good scanning speeds were noted in the clean test sets, but results in the infected areas were harder to obtain thanks to freezes and other issues. Final figures were obtained after gently coaxing the product through the test sets, with a strong showing in the standard sets but rather lower figures seen in the new RAP sets – something of a disappointment after having achieved a remarkably high score in the first run of the RAP scheme in the recent *Linux* test. As on its previous outing, the product's detection system proved a little controversial, with an extremely finely graded range of detection flags including numerous combinations of vague and unusual terminology to report various levels of heuristic detections. However, even including the full range of 'security risk' and 'possible security risk' alerts – which we would usually adjudge to be only 'suspicious' detections and thus not counted as either detections in the standard sets or false positives in the clean sets – the detection numbers still lagged somewhat behind our high expectations.

Nevertheless, the WildList was covered without problems, and the clean sets likewise handled without issue, and a VB100 certification is awarded.

F-Secure Client Security 8.00 b.232

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	93.20%
Worms & bots	100.00%	**False positives**	0



F-Secure's desktop range continues to expand, but thankfully this busy month saw only the flagship product entered into the test.

The product continues to exert its icy charms with a speedy, informative setup process and an unusual but highly usable interface, which allowed ample configuration and extremely thorough scanning. This resulted in the usual rather slow scanning times, particularly when archive scanning on access was activated against the strong recommendations of the developers – most users would have no requirement for such a level of scanning, but results are recorded here for fairness of comparison against those products which have such scanning enabled by default.

Detection rates were as strong as ever, with some excellent scores in the trojan and RAP sets, again with a fairly clear drop in the 'week +1' set, but the product offers some additional protection features including a cloud-based reputation system, which would doubtless add considerably to its protection capabilities when fully operational. Even without these extras, WildList detection was flawless and no false positives were raised in the clean sets, thus *F-Secure* ably achieves a VB100 award.

G DATA AntiVirus 19.2.0.0

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	99.69%
Worms & bots	100.00%	**False positives**	0

G DATA's multi-engine product, combining the strengths of a pair of high-performing detection engines, is another

Archive scanning		ACE	CAB	JAR	LZH	RAR	TGZ	ZIP	ZIP-SFX	Ren*
Agnitum Outpost	OD	X	√	√	X	√	√	√	√	√
	OA	X	X	X	X	X	X	X	X	√
AhnLab V3	OD	√	√	√	√	√	X	√	√	√
	OA	X	X	X	X	X	X	X	X	√
Alwil avast!	OD	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√	√
	OA	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√	√
Authentium Command	OD	X	5	5	√	5	2	5	5	√
	OA	X	X	X	X	X	X	X	4	X
AVG	OD	X	√	√	√	√	√	√	√	X/√
	OA	X	X	X	X	X	X	X	X	X/√
Avira AntiVir	OD	√	√	√	√	√	√	√	√	√
	OA	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
BitDefender	OD	X/√	X/√	√	X/√	X/√	X/8	1/√	X/8	√
	OA	X/√	X/√	√	X/√	X/√	X/8	1/√	X/8	√
BullGuard	OD	√	√	√	√	√	8	√	8	√
	OA	√	√	√	√	√	8	√	8	√
CA AV	OD	X	√	√	√	√	√	√	√	√
	OA	X	X	1	X	X	X	1	X	√
CA eTrust	OD	X	X	1	X	X	X	1	X	√
	OA	X	X	1	X	X	X	1	X	√
Check Point Zone Alarm	OD	√	√	√	√	√	√	√	√	√
	OA	X	X	X	X	X	X	X	X	√
eEye Blink	OD	X	X	1	1	1	2/8	2	√	√
	OA	X	X	X	X	X	X	X	X	√
ESET NOD32	OD	√	√	√	√	√	√	√	√	√
	OA	X	X	X	X	X	X	X	X	√
Filseclab Twister	OD	5/√	2/√	4/√	1	4/√	X	5/√	2/√	√
	OA	X	X	X	X	1	X	2	X	X
Finport Simple	OD	X	√	√	X	√	X	√	X	√
	OA	X	√	√	X	√	X	√	X	X
Fortinet FortiClient	OD	X	√	√	√	√	√	4	√	√
	OA	X	√	√	√	√	√	4	√	√
Frisk F-PROT	OD	1	√	√	√	√	√	√	√	√
	OA	1	X	2	X	X	X	2	2	√
F-Secure Client Security	OD	X	5	5	5	X	5	5	5	√
	OA	X/10	X/5	X/5	X/5	X/5	X/2	X/5	X/5	X/√
G DATA	OD	√	√	√	√	√	√	√	√	√
	OA	√	√	√	√	√	8	8	4	√
K7 Total Security	OD	X	1	1	1	1	X	1	X	√
	OA	X	X	X	X	X	X	X	X	√
Kaspersky	OD	√	√	√	√	√	√	√	√	√
	OA	X/4	X/4	X/4	X/4	X/5	X/1	X/2	X/1	√
Kingsoft (Standard)	OD	X	X	√	√	√	√	√	X	√
	OA	X	X	X	X	X	X	X	X	√
Kingsoft (Advanced)	OD	X	√	√	√	√	√	√	X	√
	OA	X	X	X	X	X	X	X	X	√
McAfee VirusScan	OD	X/2	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
	OA	X/2	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
Microsoft Forefront	OD	X	X	X	X	X	X	1	1	√
	OA	X	X	X	X	X	X	1	1	√
Microsoft OneCare	OD	X	X	X	X	X	X	1	1	√
	OA	X	X	X	X	X	X	1	1	√
MWTI eScan	OD	√	√	√	√	√	8	√	8	√
	OA	√	√	√	√	√	8	√	8	√
Norman Security Suite	OD	X	X	√	√	√	√	√	√	√
	OA	X	X	X	X	X	X	X	X	√
PC Tools AV	OD	2	√	√	X	√	√	√	√	√
	OA	X	X	X	X	X	X	X	X	√
PC Tools IS	OD	2	√	√	X	√	√	√	√	√
	OA	2	√	√	X	√	5	√	√	√
PC Tools SD	OD	2	√	√	X	√	√	√	√	√
	OA	2	√	√	X	√	5	√	√	√
Quick Heal	OD	X/2	X/5	2/5	X	2/5	X	2/5	X	X/√
	OA	X	X	X	X	X	X	X	X	X
Redstone RedProtect	OD	√	√	√	√	√	√	√	√	√
	OA	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
Rising IS	OD	1	√	√	√	√	√	√	√	√
	OA	1	√	√	√	√	√	√	√	√
Sophos Endpoint	OD	X	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/√
	OA	X	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/√
Symantec Endpoint	OD	X	3/10	3/10	3/10	3/10	1/5	3/10	3/10	√
	OA	X	X	X	X	X	X	X	X	√
Trustport	OD	X	√	√	X	√	√	√	√	√
	OA	X	√	√	X	√	√	√	√	√
VirusBuster	OD	2	√	√	X	√	√	√	√	√
	OA	X	X	X	X	X	X	X	X	X/√
Webroot	OD	X	X	X	X	X	X	X	X	√
	OA	X	X	X	X	X	X	X	X	√

Key:

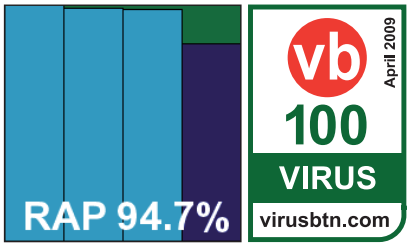
X - Archive not scanned

√ - 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



product which is regularly seen at the top of detection charts in numerous tests, and has an excellent record in our own testing. The latest edition proved quick and simple to install, although it did require a reboot to complete the process, and presented a pleasant and usable interface with a good level of configuration available. Scanning speeds were a little below average, thanks to the multi-engine approach, but the product powered through the infected test sets with no stability problems.

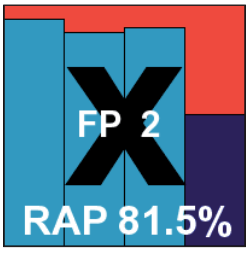
Logging proved a little awkward for our purposes but would probably suit most every-day applications of the product. Detection rates were really quite breathtaking, with over 99% in the trojan set and similarly high scores in most of the RAP sets. Although a slight drop was observed week on week, to a lower level in the 'week +1' RAP set, detection remained highly commendable even here. Attaining a new high in the RAP average scores, and with flawless performance elsewhere, *G DATA* takes maximum honours and an easy VB100 award.

K7 Total Security 9 Desktop 9.7.0200

ItW	100.00%
ItW (o/a)	100.00%
Worms & bots	99.81%
Polymorphic	74.94%
Trojans	91.28%
False positives	2

K7 has been a sporadic entrant in the VB100 testing, putting in strong performances on the occasions it has taken part, but missing a lot of tests – which puts the company at something of

a disadvantage when it comes to keeping up with additions to our clean test sets.



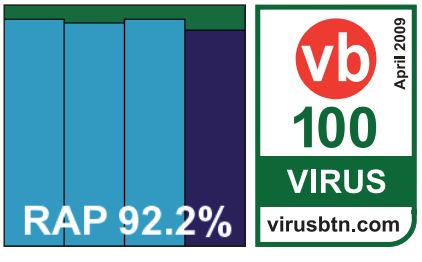
The installation process for the latest product version is fairly smooth, but requires identification details for the user, including email address, as well as a reboot before it can complete – it also offers to remove conflicting third-party software.

The main product interface, once up and running, seemed somewhat cluttered, but offered a good level of configuration and was easy to navigate and use. Detection rates were really quite excellent, with scores above 90% in the key trojan set and in several of the RAP weekly sets (a less spectacular performance in the 'week +1' set brought the overall average down to a still very respectable 81.5%). The product also includes a firewall and privacy guard for added protection.

The WildList was fully covered without issues, but in the clean sets, as feared, a couple of items were flagged as malicious. These were items included on a CD distributed widely in the UK (admittedly somewhat outside of the product's core market regions) by *AOL* in the summer of 2008, and which have been sitting in our clean sets ever since. They were flagged as the Sohanad worm and as an AutoIt trojan, thus spoiling *K7*'s chances of VB100 certification this time despite an otherwise splendid performance.

Kaspersky Anti-Virus 2009 8.0.0.506

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	96.08%
Worms & bots	100.00%	**False positives**	0



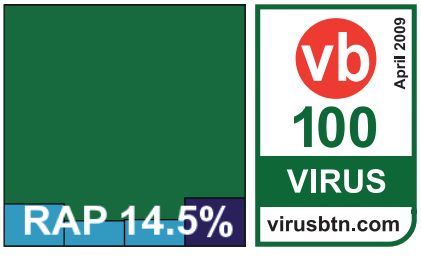
Kaspersky's latest product version is an attractive beast, with a number of added layers of security beyond the standard anti-malware tested here. The installation process includes a data-gathering wizard design to optimize the performance of these various sub-components. This is followed by a reboot to complete the installation.

The new design is very usable as well as visually appealing, and provides plenty of options for fine-tuning the protection levels to suit the individual user. Despite some fairly thorough default settings, scanning speeds were pretty good and on-access overheads fairly negligible. Detection rates, as expected after witnessing the performance of some other products using the same engine, were superb. A particularly strong showing in the 'week +1' RAP set is indicative of some strong heuristics at work in addition to the standard engine that is provided to other products. With an overall RAP average above 90%, *Kaspersky* joins the elite group of top performers, and flawless performances in the WildList and clean sets also earn it VB100 certification once again.

Kingsoft Internet Security 2009 Standard Edition 2008.11.6.63

ItW	100.00%	**Polymorphic**	48.30%
ItW (o/a)	100.00%	**Trojans**	11.97%
Worms & bots	99.27%	**False positives**	0

Kingsoft chose to enter two versions of its product this month, the first of which is a 'budget' edition which lacks some of the more advanced detection features. Although on the surface there are few indications of any difference between the two, some notable variations in performance were observed in several aspects of testing.

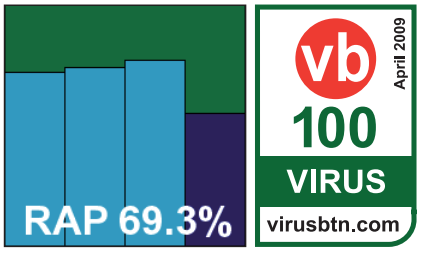


The installation process included a line in the EULA stating that 'basic information about usage' would be collected by the product and passed on to its masters, and also provided a selection box for which the only selection available was 'typical install'. On a few occasions blocks of text seemed to tail off from the installer incomplete, probably due to the integration of translations into the interface.

Scanning speeds were remarkably slow, and overheads similarly intrusive, while detection rates were generally somewhat disappointing, apparently due to a lack of complete functionality in this near-free edition. The WildList was covered without issues however, and there were no false positives in the clean sets, thus earning *Kingsoft* a VB100 award.

Kingsoft Internet Security 2009 Advanced Edition 2008.11.6.63

ItW	100.0%	**Polymorphic**	52.00%
ItW (o/a)	100.00%	**Trojans**	74.05%
Worms & bots	98.84%	**False positives**	0

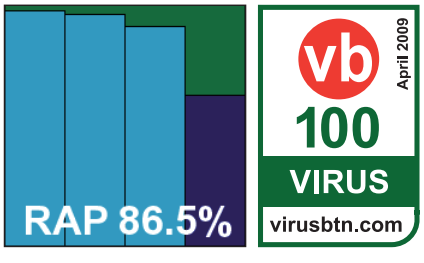


The 'Advanced' or premium version of the *Kingsoft* suite product ran through an identical installation process to that of the basic version, and presented an apparently identical interface. This time, however, scanning speeds were much more impressive. Detection rates also seemed considerably better on first run, causing us to return to the first product for a retry to ensure no logging errors had gone unnoticed – but it appeared that the disparity in detection rates and speeds is entirely due to the additional power of this premium edition.

Again doing well in the core certification requirements, *Kingsoft*'s second product has also done enough to achieve a VB100 award this month.

McAfee VirusScan Enterprise 8.7.0i

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	90.45%
Worms & bots	100.00%	**False positives**	0



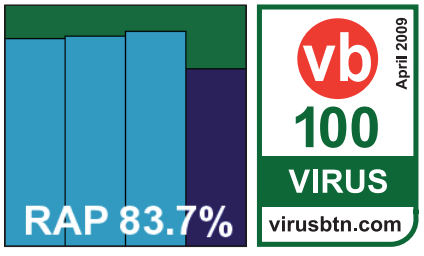
McAfee's corporate product continues to stick to its tried-and-trusted approach, with a very professional and businesslike implementation which won approval from the test team. Setup and configuration for the tests thus proved a joy rather than a chore, and testing chugged through nicely.

Speeds and overheads were both mid-range and fairly unexceptional, but detection rates were excellent in the main, with a notable drop in the 'week +1' RAP set denting the overall RAP average somewhat but still leaving a very respectable 86.5%. Real-world users would have the option of using *McAfee*'s new cloud-based 'Artemis' technology for additional protection from the latest threats, as well as other features including buffer overflow protection.

The sterling work put in across the test sets was carried over to the WildList set and the clean sets, and with nothing to mar an excellent performance VB100 certification is well earned.

Microsoft Forefront Client Security 1.5.1.1955.0

ItW	100.00%	**Polymorphic**	95.09%
ItW (o/a)	100.00%	**Trojans**	84.80%
Worms & bots	100.00%	**False positives**	0



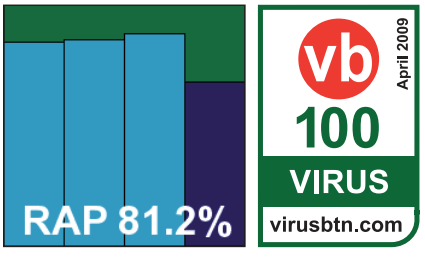
Microsoft's corporate desktop product required a later version of a standard dll before it could install, as our test systems had not been updated since the service pack. This was the only product under test to need such manual adjustments to the environment. With the adjustment made, setup was quite straightforward, and the product proved fairly simple to use, thanks in part to a minimal level of configuration available to the user.

While scanning speeds were reasonable, on-access overheads were fairly high, particularly on executable files, and our test team noticed fairly intrusive slowdowns on the system at several stages during testing.

Detection rates were fairly solid however, and pretty even across the sets, with a much less marked drop in the 'week +1' set than many solutions. With the WildList handled without issues and no false positives, *Forefront* earns itself another VB100 award.

Microsoft Windows Live OneCare 2.5.2900.20

ItW	100.00%	**Polymorphic**	95.09%
ItW (o/a)	100.00%	**Trojans**	83.35%
Worms & bots	100.00%	**False positives**	0



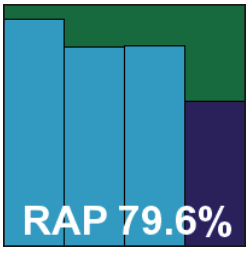
The home-user sibling of *Forefront* proved somewhat simpler to install, with a custom setup process provided to deal with our unconnected environment. The minimal user configuration, absence of progress data and marked system slowdown all made testing rather frustrating. Even worse was the failure of the logging system, which repeatedly refused to generate the 'support log' required to render detection data manageable. On-access scanning of large infected

test sets seemed too much for the product to handle on several occasions, and on a couple of occasions we found the test machine had simply shut down in the middle of a scan (although some suspected hardware issues may have contributed to this issue). On a third attempt at installing and running the product we finally managed to get usable reports, and detection proved much on a par with *Forefront*.

The WildList and clean set provided no unexpected surprises, and *OneCare* thus qualifies for VB100 certification; the team eagerly await its retirement and a more tester-friendly setup in the replacement free version *Morro* due to be made available in the latter half of this year.

MWTI eScan Protection Center 10.0.962.360

ItW	99.01%	**Polymorphic**	100.00%
ItW (o/a)	99.01%	**Trojans**	95.11%
Worms & bots	100.00%	**False positives**	0



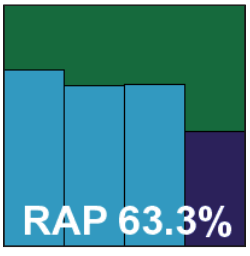
MicroWorld's *eScan* went through a standalone review recently (see *VB*, January 2009, p.16) and was found to be extremely well designed with some excellent additional protection features, the configuration of which is a glowing example of user-friendliness. This latest update was found to be visually appealing by the test team, with a fast installation process that includes a pre-install scan, but which requires a reboot to complete. Default settings are fairly thorough, which is reflected in rather sluggish scanning speeds and fairly hefty on-access overheads.

Previous editions of the product included the *Kaspersky* detection engine alongside various items of in-house technology, but the firm announced a few months ago that its latest range would include entirely in-house engines – a bold move. With the new setup, detection rates were very solid across most of the test sets, with some excellent figures in the trojan and RAP sets, although rates declined somewhat over the very newest items. With all the additional HIPS technology included in the product, the protection provided against threat vectors in the real world would, of course, be increased.

The product encountered no problems in the clean sets, but in the WildList set a couple of the recent additions to the list were missed, showing some minor teething problems for what looks likely to be a strong new detection engine. No VB100 award is forthcoming this month, but *MWTI* looks likely to be back on track very soon.

Norman Security Suite 7.10

ItW	99.79%	**Polymorphic**	83.21%
ItW (o/a)	99.79%	**Trojans**	81.14%
Worms & bots	100.00%	**False positives**	0



Norman's product has undergone a significant facelift of late, but despite a speedy installation process the new look did not go down well with the test team, who found it rather peculiar to look at, very short on options, and difficult to navigate. There appeared to be no option to run on-demand scans from the interface, and the scheduler system seemed not to be working for us, so on-demand tests were run using the right-click scan option.

This produced some fairly slow scan times on demand, thanks to the intensive sandbox technology, but on-access overheads were pretty light. After running some of the detection tests the product ran into some difficulties, in which the right-click option vanished and protection was apparently disabled; even the protection area of the interface appeared to have vanished without trace. Logging of the tests carried out thus far showed results well short of the expected level. With no response from any attempt to revive it, and even a reboot proving inadequate, a fresh install was required to complete the testing.

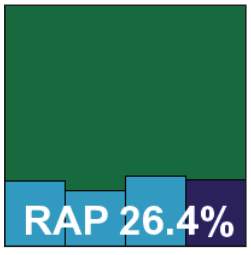
On second attempt things went a little better, with some much more stable behaviour getting us far enough to acquire and process full detection logs. The logs showed detection figures that were pretty much in line with previous performances, before the mysterious shutdown occurred once again. Analysis of the results showed some pretty decent scores in the trojan set, with a gradual decline across the RAP sets to a fairly low level in the 'week +1' set. Elsewhere, the W32/Fujacks samples in the WildList set were missed, and so *Norman* does not make the grade for a VB100 award this month.

PC Tools Anti-Virus 2009 6.0.0.16

ItW	99.75%	**Polymorphic**	18.55%
ItW (o/a)	99.75%	**Trojans**	22.32%
Worms & bots	99.81%	**False positives**	0

The *PC Tools* product lines have caused us some difficulties in the past, as much thanks to their oddities of behaviour and design as to a tendency for more than one version to be submitted. This month, three products were submitted, of which we were told that the simple AV solution was

considered the lowest priority by the vendor, should any have to be excluded from the test due to time constraints.



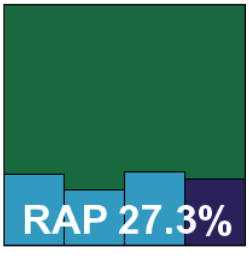
It also proved somewhat simpler to test than the others in the range, with a speedy and simple installation process after which no reboot was required. The interface provides minimal configuration and has a few peculiarities of layout which makes the options that are available less than easy to find. However, it seemed to work reasonably well in the on-access tests over clean and archive sets.

Attempting to run the same test over the infected sets appeared to go smoothly at first, but halfway through protection seemed to shut down and blocking access to infected items ceased; they were no longer logged either. After several attempts at the test, including slowing down the rate of file access, we eventually managed to coax what appeared to be usable results from the product, although the periodic shutdowns continued. On demand tests were less tricky, although the results found in the logs, particularly for the RAP sets, were much lower than expected. In the WildList, the W32/Fujacks set of samples were not detected, with an additional file missed on access only, and as a result *PC Tools* does not earn a VB100 for its AV product this month.

PC Tools Internet Security 6.0.1.440

ItW	99.75%	**Polymorphic**	18.55%
ItW (o/a)	99.75%	**Trojans**	22.79%
Worms & bots	99.85%	**False positives**	0

The second *PC Tools* product, the *Internet Security* suite, combines the anti-malware protection of the company's flagship *Spyware Doctor* product with some additional protection measures, including a firewall.



Installation, which includes the offer of a *Google* toolbar along with the product itself, seemed fairly straightforward until the product was up and running, at which point it was immediately clear that something was not right – all status alert records were marked 'off' or 'checking', and on-access detection was clearly not present. Upon consulting with the developers, we were informed of some recently discovered issues with our rather unusual hardware setup, which should have been resolved by a simple reboot – but this proved ineffective. Eventually, we managed to persuade the product to switch itself on by connecting it to the Internet, with updates disabled; within a few seconds it all came online. This kind of thing is not uncommon these days, but is something of a problem for many users. Although I may be somewhat atypical and overly paranoid, I like to ensure that a new system is fully protected and even up to date before I expose it to the Internet, so always use offline installers and updaters where possible when building a new machine or reimaging from a known safe state – being forced to go online to activate a product is of no interest to me. However, many products seem to want to do such things to prevent piracy or for other reasons best known to them.

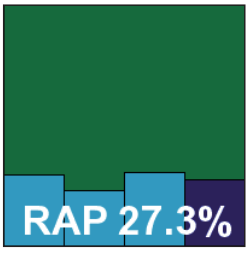
With the product finally activated, we ran through the tests. In this product, on-access scanning is not activated by simple file access, so once again we had to resort to copying test sets across the network and trusting the product's logging to show us if it suffered similar shutdowns to the previous version. Logging, of both on-access and on-demand data, proved less than helpful, regularly imposing apparently random cut-off points, though it was not always clear if this was the protection or the log that had ceased to record new arrivals. Eventually, after much sweating and cursing from the team, we managed to obtain usable data, which fairly closely matched that of the previous product, leading us to believe that both must be representative of the protection offered.

On-demand scanning speeds were rather slow, particularly over the archive set, and while on-access times could not be recorded using our standard methods, it was obvious that the systems were much less responsive, and the product interface itself proved especially slow to respond. Detection results were not great, and the W32/Fujacks samples in the WildList set put paid to *PC Tools*' hopes of certification for this product too.

PC Tools Spyware Doctor with Anti-Virus 6.0.1.440

ItW	99.75%	**Polymorphic**	18.55%
ItW (o/a)	99.75%	**Trojans**	22.79%
Worms & bots	99.85%	**False positives**	0

The third and final *PC Tools* product proved almost identical to the suite product, minus the firewall, and provided the same sort of agonies for the test team, including the need to connect to the web to get it to turn anything on. After repeated attempts and numerous apparent brick walls, some sort of results emerged from the confusion, proving

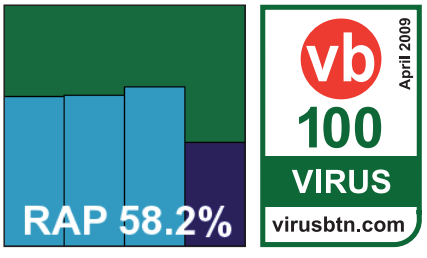


pretty much identical to the suite product right down to the WildList misses and failure to qualify for certification.

Due to the numerous problems with the products, not least the unreliable logging features, it is more than possible that the results recorded here do not show the full detection capabilities of the product range, but they are at least an approximation of the best detection that could be coaxed from the product over several arduous days of repeated tests.

Quick Heal Anti-Virus Lite 2009

ItW	100.00%	**Polymorphic**	95.09%
ItW (o/a)	100.00%	**Trojans**	86.61%
Worms & bots	99.32%	**False positives**	0



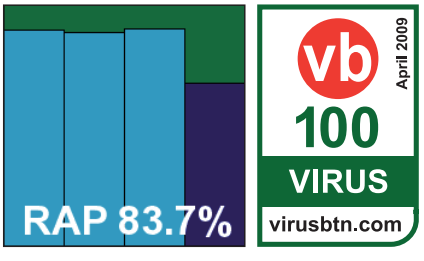
As usual, *Quick Heal*'s product lived up to its name with a very rapid installation process and no reboot necessary. The interface was perhaps a little confusing, with some of the options hidden away in unexpected places, but it generally proved usable and responsive with no stability issues.

Scanning speeds were, as expected, remarkably quick, and on-access overheads extremely light. Detection across the test sets was fairly average, with a pretty marked drop in the 'week +1' RAP set, but the product does include additional features, including some advanced static heuristics based on file locations and names which would not be reflected by our testing methodology.

In the core areas of the WildList and clean sets there were no problems however, and *Quick Heal* duly earns a VB100 award.

Redstone RedProtect 1.7.5

ItW	100.00%	**Polymorphic**	100.00%
ItW (o/a)	100.00%	**Trojans**	93.16%
Worms & bots	100.00%	**False positives**	0



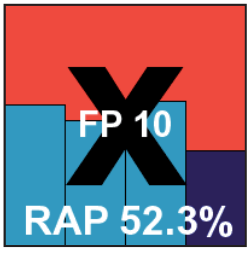
Redstone's product is a rather unusual one, designed to be managed entirely remotely with little user interaction. The installation process, which is dependent on the .NET framework, was thus custom-tweaked for our purposes, and access to configuration was also provided via a custom interface. Both were fast and simple to use, and highly rated by the test team for usability.

The 'default' settings provided for us were thorough, resulting in below-average scanning speeds, but overheads were not too intrusive.

Detection rates from the *Kaspersky* engine were as excellent as we would expect, although a notable drop over that tricky 'week +1' RAP set indicated that some aspects of *Kaspersky*'s detection abilities are not included here. With the WildList set covered flawlessly, and no problems in the clean sets, *Redstone* comfortably earns a VB100 award.

Rising Internet Security 21.27.10

ItW	99.75%	**Polymorphic**	70.02%
ItW (o/a)	99.75%	**Trojans**	56.71%
Worms & bots	99.18%	**False positives**	10



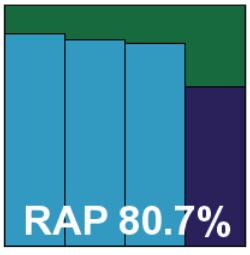
Rising's product is another to have been reviewed in depth recently (see *VB*, March 2009, p.13), and full details of the setup process (rather complex, with a reboot and several post-install wizards) and additional features (which include a dancing lion cartoon and a range of firewall and basic HIPS technologies) are covered in more depth there.

In this case we mostly looked at scanning speeds and detection rates. Despite some very thorough default settings which covered most of our archive sets in full depth, on-demand scanning was fairly rapid, while on-access scanning is only available on write or on execute and thus could not be fitted into our standard overhead measurement.

Detection results were gathered by copying test sets to the system across the network, and proved fairly mediocre across the board. In the clean sets, a smattering of false positives were raised, and in the WildList set a single W32/Autorun variant was not detected, and as a result *Rising* will have to wait a little longer for its next VB100 award.

Sophos Endpoint Security and Control 8.0 (7.64)

ItW	100.00%	**Polymorphic**	89.25%
ItW (o/a)	100.00%	**Trojans**	83.49%
Worms & bots	100.00%	**False positives**	0



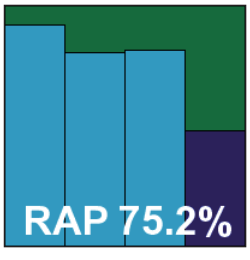
The *Sophos* product proved very smooth and quick to install, and was another one of the select few that offered to remove conflicting third-party software. No reboot was required.

The interface is clear and simple, with a great deal of configuration tucked away under the bonnet, as befits the product's corporate target market. On-demand scanning speeds were pretty decent, and on-access overheads not too intrusive, at least with the sensible default settings. Detection rates were solid and respectable across the sets, with a fairly notable drop in the unknown 'week +1' samples.

The WildList presented no issues, and in the clean sets only a couple of suspicious alerts were raised (on files which turned out to be of rather peculiar makeup). A VB100 award is thus earned by *Sophos*.

Symantec Endpoint Protection 11.0.4010.19

ItW	100.00%	**Polymorphic**	99.96%
ItW (o/a)	100.00%	**Trojans**	91.49%
Worms & bots	100.00%	**False positives**	0



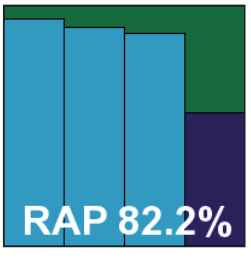
Symantec's corporate desktop product, previously much praised for its plain and businesslike style, has become a lot more glossy and colourful of late, but remains grey and serious in the deeper configuration areas. Installation is pretty simple, and navigation of the interface is reasonably sensible, with the configuration pages, once dug out, providing a fair level of control over the product's behaviour.

Scanning speeds were fairly middling, but on-access overheads were not bad at all, and testing thus progressed fairly rapidly. When scanning the infected sets, the machine shut down unexpectedly during one of the on-access tests, and on another occasion the interface suffered a crash, although protection remained in place.

Detection rates proved pretty decent, although the 'week +1' drop was fairly sharp. With no problems encountered in the WildList test set and no false positives in the clean set, *Symantec* takes another VB100 in its stride.

Trustport Anti-Virus 2009 2.8.0.3012

ItW	99.79%	**Polymorphic**	98.56%
ItW (o/a)	99.79%	**Trojans**	94.50%
Worms & bots	100.00%	**False positives**	0

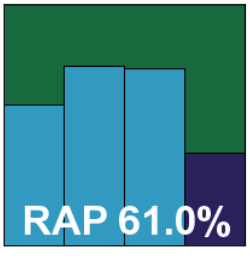


Trustport's multi-engine approach has achieved some superb scores in some recent tests, although frequent changes to the combination of engines included have led to some less distinguished performances too. The latest version offers a fast and simple installation, with some new adornments to what is essentially the same interface, currently using the *Norman* and *AVG* engines under the covers.

With some very thorough defaults on top of the multi-engine design, scanning speeds are understandably rather slow, and on-access overheads also rather heavy, but detection rates were generally pretty good. Scores above 90% were achieved in the trojan set and some of the RAP sets, but the 'week +1' set showed a fairly steep decline. Oddly, a few items including the W32/Fujacks replicants were not detected in the WildList set – suggesting that slightly outdated detection data may have been in use. As a result, *Trustport* is denied a VB100 award this time.

VirusBuster Professional 5.003 b.155

ItW	100.00%	**Polymorphic**	88.85%
ItW (o/a)	100.00%	**Trojans**	69.71%
Worms & bots	99.92%	**False positives**	0



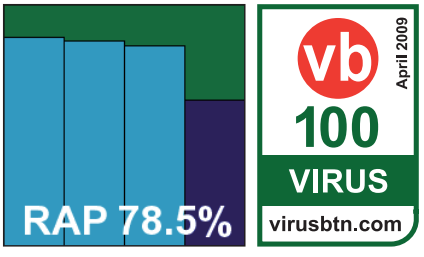
VirusBuster's product is another which has remained little changed over several years of testing, and our test engineer remarked on some awkwardness in the otherwise speedy installation, as well as a rather unintuitive main interface. However, with the help of some experience to navigate its peculiarities, testing proceeded, with some good scanning speeds in both modes helping things along.

Detection rates were somewhat below average, with a particularly sharp drop in the 'week +1' RAP set, but

elsewhere things were a little more respectable, and with no problems in the WildList and no false positives, *VirusBuster* earns another VB100 certification.

Webroot Anti-Virus with Anti-Spyware

ItW	100.00%	**Polymorphic**	89.16%
ItW (o/a)	100.00%	**Trojans**	82.50%
Worms & bots	100.00%	**False positives**	0



Webroot's product has undergone some name changes but seems little changed in layout since the company's first entry. The installation went smoothly thanks to some well-documented additional steps required to fit in with our lab setup, but the interface proved highly unpopular with the lab team, who remarked on its awkward and unintuitive layout, the difficulty of finding the few options available, and also some extremely slow response times to fairly simple button clicks. Other areas where bad behaviour was noted included logging, which was regularly truncated and barely usable in some cases, and on-detection actions, which were often performed despite specific instructions to do nothing.

Eventually, after much hair-tugging, results were obtained, and proved much in line with the *Sophos* engine underlying the product. With no false positives and nothing missed in the WildList set, *Webroot* earns the final VB100 award of this month's test.

CONCLUSIONS

This month's test has presented the usual ups and downs, with some truly excellent products and some real horrors. The first full-scale rollout of our RAP tests has provided some interesting data on the whole, with several products excelling and a few failing to impress. Many products showed a gradual week-on-week decrease in detection rates, which is as predicted and goes some way to validating the test methodology. The severity of the final week drop in detection is perhaps the most telling part of the results, indicating how well heuristic and generic detection is working.

In a couple of cases, where products integrate engines bought in from outside, the results have shown how well some OEMs are adding their own technology to what they have bought in, while in one case the OEM has done less well than the original engine maker in the vital heuristic area.

In the standard areas of the test, a pretty good month was had by most, with a large number of VB100 awards having been handed out. A smattering of false positives ran through a number of products, most of which were caused by the batch of files from a UK *AOL* CD hitting Asian-focused products. As mentioned, we have been trying to work on ways of ensuring our clean sets are kept relevant, and are hoping to introduce some more advanced classification and ranking of clean files at some point. The issue raised here though – that of the locality of clean samples, where samples likely only seen in one specific region have spoiled the chances of products that are focused on an entirely different region – is less simple to solve. Our testing aims to present a global picture, and so our detection standards – both for infected and clean files – must try to reflect the global landscape of malware and software. While we cannot ignore the effects of files from one region on products from another, we can (and do) make efforts to ensure our test sets fairly reflect all regions.

Another major headache this month has been product stability issues, something that has been raised here in several recent tests. In a number of cases it has left our lab techs astounded to see how fragile and unstable some software can be – particularly considering it is supposed to be protecting systems from danger. Some of our advisors have even suggested automatically failing any product which crashes – something we will certainly have to consider when we next update the test procedures.

This month saw a smattering of misses in the WildList, most notably a small number of fairly simple file infectors. We have seen similar incidents before and hope they encourage analysts to ensure that file infectors continue to be handled properly, and not lost in the floods of static samples pouring into labs. The next test (which will take place in May on the *Windows Server 2003* platform) should see some much more tricky polymorphic items making their way onto the WildList, and we look forward to the challenge this will pose for the products on test.

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 XP Professional, Service Pack 3*.

Any developers interested in submitting products for VB's comparative reviews should contact john.hawes@virusbtn.com. The current schedule for the publication of VB comparative reviews can be found at http://www.virusbtn.com/vb100/about/schedule.xml.

END NOTES & NEWS

Black Hat Europe 2009 takes place 14–17 April 2009 in Amsterdam, the Netherlands, with training taking place 14–15 April and the briefings part of the event from 16–17 April. Online registration is now open (onsite registration rates now apply). See http://www.blackhat.com/.

RSA Conference 2009 will take place 20–24 April 2009 in San Francisco, CA, USA. The conference theme is the influence of Edgar Allen Poe, a poet, writer and literary critic who was fascinated by cryptography. For more information including registration rates and packages see http://www.rsaconference.com/2009/US/.

The Computer Forensics Show will be held 27–29 April 2009 in Washington, DC, USA. For more information see http://www.computerforensicshow.com/.

Infosecurity Europe 2009 takes place 28–30 April 2009 in London, UK. For more details see http://www.infosec.co.uk/.

The 3rd International CARO Workshop will take place 4–5 May 2009 in Budapest, Hungary. This year the focus of the workshop will be on the technical aspects and problems caused by exploits and vulnerabilities in the broadest sense. For more details see http://www.caro2009.com/.

The 18th EICAR conference will be held 11–12 May 2009 in Berlin, Germany, with the theme 'Computer virology challenges of the forthcoming years: from AV evaluation to new threat management'. For more information including programme details see http://eicar.org/conference/.

SEaCURE.IT will be held 19–22 May 2009 in Villasimius, Italy. SEaCURE.IT is the first international technical conference to be held in Italy on security-related topics, aimed at bringing together leading experts from all over the world, to create a unique setting for networking and discussion among the speakers and the attendees. For details see http://www.seacure.it/.

NISC 10 will take place 20–22 May 2009 in St Andrews, Scotland. For more details including provisional agenda and online registration see http://www.nisc.org.uk/.

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

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. Online registration is now open and a call for papers has been issued, with a deadline for submissions of 1 May. 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/.

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. 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/.

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

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

SPAM LEVELS RECOVER

Spam volumes have returned to the same levels as those seen prior to the takedown of the *McColo* ISP in November 2008.

Spam levels plummeted after the *McColo* ISP – which hosted botnet control centres that controlled zombies around the world, and which were responsible for more than 75% of the spam sent globally each day – was taken offline by its upstream providers. The ISP was blocked when evidence of suspicious activities on its network was presented to the upstream providers.

Although spam volumes started to rise again just two weeks after the web-hosting firm was taken offline, it has taken four months for them to recover to the same heights. According to *Google*, the seven-day average spam volume observed during the second half of March was the same as that seen prior to the blocking of *McColo*.

According to *Google*'s researchers, data suggests that spammers are adopting new strategies to avoid a similar takedown from occurring in the future – such as building botnets that are more robust but send smaller volumes of messages, or running botnets at less than their full capacity in order to stay under the radar.

EVENTS

The Counter-eCrime Operations Summit will be held 12–14 May 2009 in Barcelona. See http://www.antiphishing.org/.

The 16th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will be held in Amsterdam, The Netherlands, 9–11 June 2009. 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/.

FEATURE

MAIL AUTHENTICATION WITH DOMAIN KEYS IDENTIFIED MAIL – PART ONE

John Levine
Taughannock Networks, USA

Message authentication is a promising technique that can be used to help separate wanted email from unwanted email. Domain Keys Identified Mail (DKIM) is a new authentication technique that seems likely to gain wide acceptance. In this article we start by looking at what message authentication is (and isn't), then at how DKIM works, and finally at how DKIM fits into an overall mail-handling strategy.

WHAT IS MESSAGE AUTHENTICATION?

Internet email dates from an era when everyone on the Internet behaved themselves. (If they didn't, they lost their net access – a penalty too awful to contemplate.) As a result, the design of the message formats and SMTP delivery protocol didn't concern itself with security, meaning that anyone could (and can) send mail that purports to be from anyone else. At the time this was an entirely reasonable design. After all, there is no difference between this and paper mail, where anyone can scribble anyone else's return address on an envelope and drop it in a mailbox.

These days, the security weaknesses of Internet mail are painfully apparent. The ability to lie about the origin of mail makes phishing (the practice of sending fraudulent mail that attempts to trick users into revealing their banking credentials or similar) far easier. It also makes spam filtering a lot harder, since a spammer can make spam that really comes from a single source appear to come from thousands of different people.

Message authentication addresses this problem by associating a hard-to-forge identity with every legitimate message. Once you have a reliable identity associated with a message, you can make mail-handling decisions based on that identity, as well as on other characteristics of the message.

What authentication is and isn't

Although message authentication is an important tool for mail management, it is not the silver bullet that some people have taken it to be. In particular, knowing that the identity of a message is authenticated is not useful unless you know something about the identity. It is easy to assume that an authenticated message is better than an unauthenticated message, but bad guys can (and do) authenticate their mail just as much as good guys.

In the follow-up part of this article next month we will discuss some of the ways in which an authenticated identity can be used in mail management.

WHERE DID DKIM COME FROM?

People noticed Internet mail's lack of authentication a long time ago. Phil Zimmerman's Pretty Good Privacy (PGP) was used to sign mail messages as long ago as 1991, and by 1998 the Internet Engineering Task Force (IETF) had defined the S/MIME standard for signed messages. Each allows every individual email address to have its own signing key. Even though S/MIME is now built into every popular user mail program, neither it nor PGP has gained more than niche acceptance. Both require each individual user to install signing keys into his or her own mail program, and this key distribution has proved to be a major barrier to acceptance.

In 2003, a number of different domain path authentication schemes were proposed, the most successful of which were Meng Wong's SPF and *Microsoft*'s Sender-ID. Unlike PGP and S/MIME, their granularity is the domain, the part of an email address after the '@' sign. They attempt to authenticate a domain in the message (the envelope sender domain for SPF, and the From: or Sender: domain in Sender-ID) against a list of IP addresses of servers that are allowed to send messages from the domain in question.

While path authentication can work reasonably well for some kinds of mail, such as commercial mail sent in bulk from a fixed source, it is a less-than-adequate authentication technology. For example, many professional societies offer permanent email addresses to their members, who can arrange for mail sent to the society address to be forwarded to whatever ISP or work address they are currently using. This means that the member's ISP sees the incoming mail sent from the society's mail-forwarding server, not the system that originally sent it – which makes path authentication that depends on matching the original sending system fail. Even worse, the members send mail with their society address from their own ISPs, not through the society's mail server, which means that for path authentication to work, the paths for the society's domain would need to include every ISP and other server that any of the members use. There are some proposed workarounds to the forwarding problem, but they are worse than the disease they attempt to cure.

Signing systems like DKIM don't care what path the message has taken, since authentication is based on the signature which is part of the message itself, rather than its path.

Yahoo's Mark Delany developed the DomainKeys (DK) message-signing system in 2003. Experiments with it were sufficiently promising that *Yahoo* offered it to the IETF as a candidate for standardization. In 2004, Jim Fenton at *Cisco* developed a similar system called IIM, Identified Internet Mail. The IETF DKIM working group started with DK, added some bits of IIM, and made a variety of other changes to develop DKIM, which was published as RFC 4871 in 2007 (http://www.ietf.org/rfc/rfc4871.txt).

HOW DKIM WORKS

DKIM is a domain-level message authentication system. Unlike PGP and S/MIME, but like SPF and Sender-ID, DKIM is intended to provide authentication of mail in transit from one mail system to another, not long-term end-to-end security. A message can have a signature added as it is sent or at any other stage as it is relayed through the mail system, and that signature can be verified at any stage until the message is displayed to its recipient(s). However, the signature is most often added by the sender's outgoing mail server and checked at the recipient's incoming mail server (Figure 1).

A DKIM signature is a message header added to a mail message, usually at the beginning, like this:

```
DKIM-Signature: v=1; a=rsa-sha256; c=simple;
d=taugh.com; h=date:message-id:from:subject:to:
mime-version:content-type:content-transfer-encoding;
s=k0903; bh=5o0hMsSoDxzLnalxFjRtVg5UjkyYctOb5I8vMpc6h
60=; b=53KLFMz5RX06C/nX3uTiaR5dWuYw083+jBkb1jOKsejSB
Tw7CWrZdFV1unbb6pGbIELAaWywCVQxB+DDhkXpDGXaa7oedMJud/
xwmOdqCZAsFBlTOh+0DpF1B81LjfPClsgNoNpKIh2HuzxX0TwJr3g
Ick6cYS4EpwdIrARA=
Date: 19 Mar 2009 22:10:36 -0000
Message-ID:
<20090319221036.7794.qmail@simone.iecc.com>
From: John Levine <johnl@taugh.com>
Subject: DKIM article
To: Helen Martin <helen@virusbtn.com>
Mime-Version: 1.0
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: 7bit

I agree, it's one of the finest works ever written in
the English language.
```

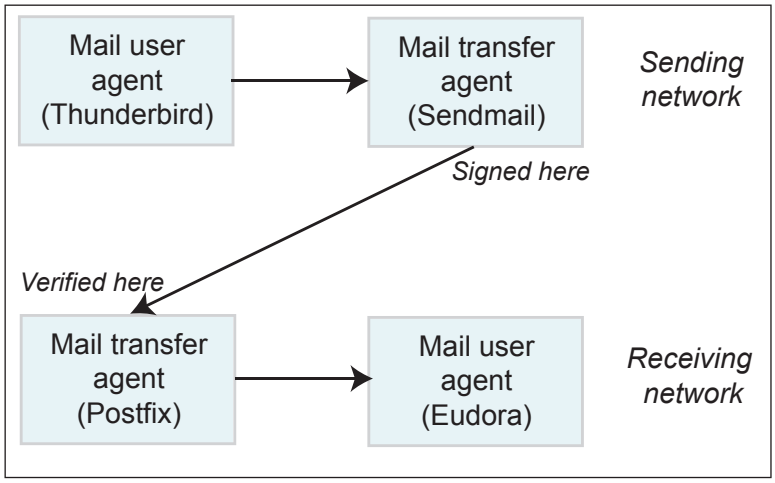


Figure 1: Typical message flow.

The verifier can first check whether the message has been modified since it was signed, and if it hasn't been modified, check if the signature validates using a verification key found in the DNS.

The signature is created in such a way it can be verified even if the message suffers minor changes (caused by mail relay software) between the time it is signed and the time the signature is verified.

Domains and selectors

The choice of identity is a major way in which DKIM differs from its predecessors. Sender-ID and DK both used the address in the From: or Sender: header to get the domain for authentication. A DKIM signer, on the other hand, can sign with any domain for which it has a signing key. This matches the structure of Internet mail much better than tying the identity to a message header. The mail server that applies the signature may belong to a mail provider that handles thousands of customers with their own domains. Even free mail providers such as *Google* and *Yahoo* (both early adopters of DKIM) allow their users to use any return address they want, subject only to a simple one-time verification (they are required to click on a URL sent to the address in question).

DKIM does not inherently assert that anything in the message is 'real', or that the From: address belongs to anyone in particular, but rather than the signing domain is taking responsibility for the message as it was at the time the signature was added.

DKIM signatures include selectors – arbitrary names used for key management. Each DKIM signature includes both a domain name and a selector, and the key is specific to the domain+selector pair. Typical uses of selectors are for key rotation, periodic switches from an old selector and key to a new one, or to allow organizations that have many physical locations to use different keys at different locations.

Since the signature identity is not tied to addresses in the message, it is possible, and often useful, to put several signatures on the same message. For example, a mail service bureau sending mail on behalf of a client might add both its own signature and that of its client when it sends the mail.

The mechanics of message signing

Creating a DKIM signature is a multi-stage process. The signer conceptually divides the message into two parts, the header and the body. First, it computes a hash value of the body. Then it selects some of the message headers, creates a second hash of the headers, signs that, and puts the signed value into a DKIM-Signature: header. The header is a sequence of field names and value pairs, in which the body hash and signed header hash are two of the fields.

Before the signer creates the body hash, it *canonicalizes* the message body, putting it into a standard form intended to deal with possible modifications in transit.

Two different algorithms can be used to perform body canonicalization, simple and relaxed. The simple one discards any blank lines at the end of the message and otherwise leaves the body as it is, while the relaxed one also removes white space at the ends of lines and squashes each sequence of white space to a single space. Either way, the signer computes a hash of the canonicalized body. The current version of DKIM uses the standard SHA-256 hash, although the spec allows for new hashes to be added in case SHA-256 turns out to have security weaknesses (as its predecessor SHA-1 did). The body hash is encoded using MIME-style base64 to become the value of the bh= field in the DKIM signature.

Next, the signer creates the header hash. Since it is quite common for headers to be added, changed and deleted in transit, the signer picks a subset of headers, leaving out the ones that are either likely to change or are not very important. The list of headers included in the signature make up the h= value in the DKIM signature.

Although not listed in the h= value, the DKIM-Signature header itself is always the last header in the list to be signed. Again, there is a canonicalization step with two options, simple and relaxed. The simple header canonicalization algorithm takes the headers exactly as they are, while the

relaxed one turns all of the header names into lower case, makes each header a single line by removing the CR/LF between continuation lines, and squashes white space into a single space. It computes a SHA-256 hash of the canonicalized headers, which includes the body hash as part of the DKIM-Signature header. It then signs the hash using its private signing key. DKIM currently uses the RSA signature algorithm but allows for new algorithms to be added in the future. It then inserts the signed hash into the DKIM-Signature header and adds it to the beginning of the message.

Having been designed by a committee, DKIM signatures have a large number of optional fields, many of which are of debatable utility at best, so I won't try to cover them all. In this example the signature includes v=1 for DKIM version 1 (in case there are future versions), a=rsa-sha256 to identify the RSA signature and SHA-256 hash, c=simple to indicate simple canonicalization for the header and body, d=taugh.com to identify the signing domain, h= the list of signed headers, s=k0903 for the key selector, bh= the body hash, and b= the signed message hash.

A controversial feature of DKIM is i=, the 'identity of the user or agent on behalf of which this message is signed'. The i= value has the syntax of an email address, and must be in the same domain as or a subdomain of the d= signing domain, but it doesn't actually have to be an email address, since there are plenty of computer systems where addresses and identities don't directly map onto each other. At the time the DKIM standard was drafted, the committee wasn't really clear whether the i= was supposed to be an email address, an address-like thing that should make sense to recipients, or an opaque token – basically a private note from the signer to itself to help track internal mail sources. (I am on the DKIM committee so this lack of clarity was partly my fault.)

An errata document likely to be published by the DKIM working group clarifies that i= is an opaque token for the signer, and verifiers should use the d= domain as the responsible identifier. Even without depending on i=, signers can still use a variety of identifiers to sign their mail if they want, since subdomains are cheap. For example, my main domain is iecc.com so I put d=iecc.com on all my outgoing mail, but my mailing lists are in lists.iecc.com, so I also put a d=lists.iecc.com signature on mail from the list manager.

Mechanics of signature verification

Verifying a signature involves first checking that the signature matches the message, then that it matches the verification key. The verifier computes the body hash in the same way as the signer. If it doesn't match, it stops, since the message it is attempting to verify isn't the one that was signed.

Then it computes the header hash in the same way as the signer, and checks that the decrypted version of the hash in the DKIM-signature matches. It looks up the decryption key (also called the verification key) in the DNS. Each key record is identified by the combination of selector and domain, named <selector>._domainkey.<domain>. (The _domainkey token ensures that the name won't conflict with names used for other purposes. Names of hosts and mail servers can't contain underscores, only names used for other purposes.)

The key record used to verify the signature shown earlier is:

```
k0903._domainkey.taugh.com. IN TXT "v=DKIM1;
h=sha256; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBg
QDoLLTbRvOcbGSFujXff4R08XXMxE5kJhFpIxBd/n/O7+YOTf
g7lUWO8D14J6bXfOC0Bm93WHj1Dj3yXfJ/QTO5TjcmsjBNwW/
XItJ4dFnEHWUg6Ta8g7intJMtdVvMjW86/LpmFy/
x3wxtHrbzifbjh0hxi54pAsCeIRuhfWyeKQIDAQAB;"
```

The key record is formatted similarly to the signature: a series of key=value pairs separated by semicolons. Once again, there are a lot of options of limited usefulness, but this key record is typical with a v=DKIM1 to indicate DKIM version 1, h=sha256 to say that this key is only to be used with SHA-256 hashes, and p= the verification key. The key type defaults to RSA, but an optional k= field will allow new keying schemes.

If the DNS key lookup succeeds, the verifier performs an RSA decryption of the b= signature using the public key from the DNS, and checks that it gets the proper header hash. If it does, the signature verification has succeeded. A message may have multiple signatures, all of which are checked in the same way. All the signatures that use the same canonicalization algorithm should have the same body hash, so the checker needs to compute the body hash at most once for each algorithm.

The result of each verification is a single bit – either it succeeds or it fails. In particular, the presence of a signature that doesn't verify doesn't imply that the message is forged or anything else bad about the message, since there are plenty of innocent reasons why a signature could break.

MAIL-HANDLING AND RELATED TECHNOLOGIES

In next month's instalment we will look at the ways in which a DKIM-authenticated domain fits into a mail-handling system, and at some related technologies that build on DKIM to help recognize good mail senders and deter phishing.