FEBRUARY 2008

virus BULLETIN

Fighting malware and spam

CONTENTS

IN THIS ISSUE

ZMIST FOR LINUX?

Peter Ferrie describes Crimea, a *Linux* virus that integrates its code with the host code, making it almost the *Linux* equivalent of *Windows* virus Zmist.
page 4

WINDOWS SERVICE WAR

In the world of Web 2.0, Java, .NET and other hot technologies we are often guilty of forgetting about the core components that make it all possible. Aleksander Czarnowski describes a simple attack scenario based on a high-privilege *Windows* service vulnerability.
page 9

VB100 COMPARATIVE REVIEW: WINDOWS SERVER 2003

John Hawes reports the successes and failures of 27 anti-malware products tested on Windows Server 2003.
page 15

vbSpam supplement

This month: anti-spam news and events, and a group of researchers from the University of Calgary get ahead of the game and explain how they believe prediction spam will work.

ISSN 1749-7027

'Well executed and comprehensive tests will light the way to better products.'

Andreas Marx, AV-Test.org

MALWARE VS. ANTI-MALWARE: (HOW) CAN WE STILL SURVIVE?

The days of the 'hobbyist' virus writer are over. Today's threats are created by a commercial malware industry which has developed quickly and which has access to some billion-dollar resources. The number of MD5-unique malware samples received by *AV-Test.org* increased from about 333,000 in 2005 to 972,000 in 2006, and 5,490,000 in 2007. The AV industry has reacted to the changing situation by issuing more frequent updates to product signatures. Some vendors have switched from weekly updates to daily, or even half-hourly updates.

VTEST, an in-house system we use to measure the response time and proactive detection of 45 AV products, downloaded a total of 111,566 unique AV updates in 2005, compared with 134,484 in 2006 and 148,869 in 2007. These numbers don't sound too extreme when compared with the number of distributed and spreading malware samples. However, the total size of the updates was only 520 GB in 2005, while we had to deal with 1.0 TB in 2006 and 1.6 TB in 2007. The average size of the signature databases has at least doubled and in some cases tripled within the last 18 months. The trend seems to be clear: more updates and more signatures, and with them longer scan times, higher memory consumption, higher false positive rates and the like.

In the past there has often been discussion about the future of signature scanners and speculation as to when they will become obsolete. The AV industry is still alive and quite healthy, however it can only be a matter of time until we need to switch our protection mechanisms to a more effective technology – even if it's not yet clear exactly what form the future products will take.

One possible solution would be a centralized database containing fingerprints of all known good and bad programs, with online checks being performed for all newly received files. However, such a database would need billions of entries in order to keep up with all the programs and patches being released, and some users might have concerns about privacy. Besides this, of course, there is the question as to who should define what is bad and what I can run on a user's PC.

One very promising idea is the behaviour-based technology which is integrated in a good number of security suites already. These offer 'dynamic detection', based on the knowledge of the typical behaviour of 'good' programs and of what combination of actions are likely to be suspicious. In some cases these products present hard to understand or incomplete information to the user, so we need to work on improving these – it is important for the program not to ask the user what to do, but to act automatically, based on all information gathered from the runtime behaviour.

A lot of ideas as to the form future AV products might take have been discussed during the last few months. These include, but are not limited to: buffer overflow protection, URL filtering, web reputation services, browser sandboxing, virtualization, patch management and the like. Let's see what happens and how, alongside the development of new products, the testing of new technologies matures.

Indeed, it is important for testers to understand the importance of their work, as most developers focus on the aspects of a product that are frequently reviewed by testing organizations and which are used to compare and rank products. Developers often only get approval of the required budgets and help from management if they can be shown to help improve the product's performance in tests.

Well executed and comprehensive tests will light the way to better products – it is not only the developers who contribute towards the improvement of products. Thus, it is essential for testers to move on to the next level of product testing, focusing on everything besides the 'traditional' signature detection. If this doesn't happen, an entire industry might run into trouble and with it, billions of users may be misled by inadequate tests.

NEWS

ANTI-MALWARE SCHOOL

Researchers at *F-Secure* have decided to do their bit in helping to educate the next generation of malware analysts. A new course entitled 'Malware Analysis and Antivirus Technologies' starts at Helsinki University of Technology this spring.

The course covers topics including reverse engineering, the use of debuggers, emulators and disassemblers, unpacking and decrypting code and designing an anti-virus engine, as well as more general topics such as mobile malware and malware in the *Windows* environment. Lectures will be given by senior analysts and researchers from *F-Secure*, including Mikko Hyppönen, Mika Ståhlberg and Gergely Erdelyi – all past *VB* conference speakers.

The 40-place course is already fully subscribed, but the organizers say they will consider repeating it if there is sufficient interest.

MORE ROGUE FLASH ADS

Following on from last month's feature on the SWF.AdHijack family (see *VB*, January 2008, p.12), malicious *Flash* ads were found to have made their way into popular travel site Expedia.com and music download site Rhapsody.com. According to *Trend Micro* researchers, the *Expedia* site was infiltrated by a variant of the SWF.AdHijack family – clicking on the ad led to a number of redirections, which eventually resulted in the installation of a piece of rogue anti-spyware detected by *Trend* as TROJ_GIDA.A.

The malicious ad found on the *Rhapsody* site similarly redirected users to a page that attempted to install a bogus program on the user's machine by reporting a (non-existent) system infection, and then urging them to purchase the software needed to 'clean' the infections.

Investigators estimate that the ads were active on the *Rhapsody* site for six days before being removed. According to *Expedia* an 'imposter advertiser' managed to circumvent the company's advertising policy. At the time of writing the company didn't know how long the ad had been active.

ALL IN THE NAME

Last month, Czech firm *Grisoft*, developer of widely used anti-malware product *AVG*, changed its corporate name to *AVG Technologies CZ, s.r.o*. Having operated under the *Grisoft* name for nearly 17 years the company will now go under the same name as its popular product. Similar moves have been made in the past by vendors including *BitDefender* (formerly *SOFTWIN*) and *McAfee* (formerly *Network Associates* – after having originally been *McAfee Associates*).

Prevalence Table – December 2007

Virus	Type	Incidents	Reports
W32/Netsky	Worm	1,511,442	32.90%
W32/Mytob	Worm	1,102,498	24.00%
W32/Lovgate	Worm	384,051	8.36%
W32/Bagle	Worm	338,768	7.37%
W32/Zafi	Worm	288,712	6.28%
W32/Mydoom	Worm	127,331	2.77%
W32/MyWife	Worm	119,616	2.60%
W32/Virut	File	118,541	2.58%
W32/Stration	Worm	89,171	1.94%
W32/Sality	File	75,544	1.64%
W32/Zoek	Worm	41,188	0.90%
W32/Grum	Worm	35,947	0.78%
W32/Autorun	Worm	34,204	0.74%
W32/VB	Worm	32,460	0.71%
W32/Rontokbro	Worm	28,200	0.61%
W32/Bagz	Worm	24,781	0.54%
W32/Fleming	Worm	21,911	0.48%
W32/Klez	Worm	17,659	0.38%
W32/Hakaglan	Worm	15,248	0.33%
W32/Rjump	Worm	12,257	0.27%
W32/Parite	File	11,680	0.25%
W32/Sohanad	Worm	11,182	0.24%
W32/Autoit	Worm	9,909	0.22%
VBS/Areses	Script	9,858	0.21%
W32/Funlove	File	9,805	0.21%
W32/Agent	Worm	9,643	0.21%
VBS/Small	Worm	9,212	0.20%
W32/Jeefo	File	7,698	0.17%
W32/Alman	File	6,304	0.14%
W32/Small	Worm	5,527	0.12%
W32/Bugbear	Worm	4,975	0.11%
W32/Looked	File	4,605	0.10%
Others[1]		74,310	1.62%
Total		4,594,237	100%

[1]The Prevalence Table includes a total of 74,310 reports across 229 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS 1

CRIMEA RIVER

Peter Ferrie
Symantec, USA

In 2001 we received a virus for *Windows* that integrated its code with the host code, making it very hard to find. That virus was Zmist (see *VB*, March 2001, p.6). In 2007, we received a virus that might be considered 'Zmist for *Linux*'. That virus was Crimea.

THE BROTHERS KARAMAZOV

The Crimea virus family contains four variants. The first (version 0.5) was a very early release and gained control via an entry in the .ctors section. This method had been described previously by a virus writer known as izik, and is in some ways the *Linux* equivalent of the *Windows* Thread Local Storage entry point method (see *VB*, June 2002, p.4). The other three Crimea variants (0.23, 0.24, and 0.25.2) are very closely related and are essentially the 'finished product'. These variants will be described in this article.

The 0.23 variant replicates before running the host. This causes a noticeable delay, since the virus runs slowly. In the 0.24 variant, however, the virus starts by running the host code as a separate process, then sets itself to the lowest scheduler priority before replicating. This reduces the CPU usage significantly. However, the change causes another noticeable effect – the child process will not terminate until the parent does, because the child process expects the parent process to be interested in the exit code. This bug was fixed in the 0.25 variant by sending a signal to tell the kernel prior to running the host that the child can terminate on exit.

In all cases, the virus continues by decrypting its data then beginning the search for files to infect.

SEEK AND YE SHALL FIND

The search routine enumerates all entries in the current directory, skipping any that begin with '.'. This allows the virus to skip the '.' and '..' directories, but it also means that it skips any files that begin with '.' (though there are not usually many on a typical system). The virus also skips symbolic links.

For each entry that it considers to be valid, the virus calls chdir(). If the chdir() call succeeds, then the entry must correspond to a directory, and the virus repeats the search in that directory and in any subdirectories that are found. Otherwise, the virus assumes that the entry corresponds to a file. If the file has the executable attribute set, the virus will attempt to infect it.

COARSE FILTERING

The virus applies a number of filters to remove unsuitable files. The conditions of these filters include that the file size is at least 16kb, and not more than 512kb. The file must begin with an ELF header, it must be a shared object for the 32-bit *Intel* 80386 or better CPU, the ELF version must be current, and the OS/ABI version must not be specified. The ninth byte of the padding field must also be zero – a non-zero value is the infection marker for the virus. The final filter checks that each program header describes a valid section.

The virus can only infect position-independent files. The reason for this is that position-dependent files can contain values that are indistinguishable as addresses or constants, since there is no relocation information. This would force the virus to guess – and an incorrect guess would corrupt the host and ruin any chance for the virus to survive. In position-independent files, there is enough context to know what the values represent.

FINE FILTERING

The virus examines the section headers of the files that pass the first level of filtering to look for required items. The virus requires sections with the names '.plt', '.got', '.got.plt', '.rel.plt', '.rel.dyn', '.data', and '.init'. The virus also requires sections of type SHT_DYNSYM and SHT_DYNAMIC, but forgets to check if SHT_DYNAMIC has been found. A missing SHT_DYNAMIC type will cause the virus to crash later and corrupt the host.

Another problem is that the virus uses an AND-mask to check that all items have been found. This is unreliable for certain values of the map address if the section table crosses a page, because the AND will zero out all bits and look as if no pointer was found. However, the file would simply not be infected in that case.

The virus loads all of the data for sections that have file content (that is, ignoring purely virtual sections). For sections that contain executable code but are not the '.plt' section, the virus disassembles the code into a special buffer.

DISASSEMBLY (HOST)

The virus disassembles the host code instruction by instruction, without regard to the code flow. This is problematic for files that contain embedded data, since the data could appear to be a set of valid instructions, but the interpretation of these could cause the real instructions that follow to be misinterpreted.

While disassembling the code, the virus constructs an ordered list of the instructions. This list will be used later to integrate the virus code with the host code. The virus contains a special check for alignment sequences, since their presence indicates a routine whose alignment must be preserved after any movement.

The disassembly is done by a library called XDE, which was written by the author of Zmist. The XDE library is fairly primitive in a sense – the instruction set that it carries is approximately equal to that of an early *Intel Pentium* CPU. There is no support for *Intel MMX* or *SSE*-style technologies, or even quite common instructions such as CMOV. The XDE library contains a bug in that there is no limit to the length of an instruction. Even though duplicated prefixes will cause a BAD flag to be set, the virus never checks for it. The XDE library also contains another bug, this time in the SIB handling, where certain encodings cause the wrong register to be chosen.

CODE MARKING (HOST)

The virus then parses the host code, beginning with the entry point and following all calls, jumps and branches. Each instruction that is encountered is marked as 'active'. The calls and branches are followed recursively to allow continuation on return from a call, or if a branch is not taken.

The virus keeps up to 15 of the most recent instructions in a special buffer. This buffer is used to deal with jump tables. The problem with jump tables is that they are not single instructions, but collections of them. By keeping the recent instructions in a buffer, when an instruction is seen that corresponds to the last one in a jump table sequence, the buffer can be queried to see if the rest of the code matches the entire jump table sequence. The 0.25 variant improves on the jump table recognition by tracing the register context, since jump tables can have multiple forms.

Two bugs exist in the handling of jump tables in the 0.23 and 0.24 variants. The entries in a jump table are stored in a buffer for later relocation. In the buggy variants the buffer has a fixed size and is shared among all jump tables. One bug is that the entries are added to the buffer without any bounds checking. Thus, if there are more than 1,023 entries, memory corruption will occur. The other bug is an off-by-one calculation which means that the last entry in each jump table is not added to the buffer. Both bugs have been fixed in the 0.25 variant. The 0.25 variant (re)allocates the jump table buffer dynamically and adds all entries correctly.

The marking function looks specially for calls to imported functions, since they cannot be followed to their conclusion. The marking completes when a 'hlt' or 'ret' instruction is seen at the top level. Upon completion, any instruction that has not been marked as active can be discarded.

DISASSEMBLY (VIRUS)

At this point, the virus disassembles itself, if it has not been done already. This disassembly differs from that of the host with respect to the code flow. The virus disassembles itself by following all calls, jumps and branches. The calls and branches are followed recursively.

CODE MARKING (VIRUS)

The virus parses its own code in the same way as for the host, but in addition to marking the instructions as active, the instructions are marked as 'viral'. The reason for this is that after infection the host instructions are discarded from memory, leaving only the virus instructions. This speeds up the infection of other files in the same session, since the disassembly and marking are no longer necessary for the virus code.

THE IMPORT BUSINESS

The virus searches the host import table for all imports that it requires. Any missing import is added to a list, and this leads to a potential bug. The 0.23 and 0.24 variants of the virus use only 24 imports; the 0.25 variant uses 25 imports. Most of these are likely to be imported already by the host. However, any future variants of the virus might make use of more obscure imports that the host will not import. The bug is that the list has a fixed size, and entries are added to the list without any bounds checking. Thus, if there are more than 32 entries, memory corruption will occur.

Once the import processing has been completed, the virus adds the appropriate relocation and stub entries for any newly added symbols and updates the hash tables to allow the symbols to be found. The section sizes are increased as required.

MIX AND MATCH

The virus then reconstructs the code section, alternating a block of host code and a block of virus code. Each of the blocks ends with a 'jmp' or 'ret' instruction. Any routine that was aligned prior to infection will be realigned, if necessary. The instructions that were not marked as active are discarded now. Then, for each block of code in the code section, there is a 1-in-16 chance that the virus will exchange the position of that block with the position of the following block.

INSTRUCTION ISOTOPES

In the 0.24 and 0.25 variants, the virus searches the code section for all two-byte instructions that use MODR/M format in register mode, and replaces some of them randomly with functionally equivalent alternatives.

There is a one-in-four chance of replacing 89/8b (mov reg2, reg1) with push reg1/pop reg2. There is a one-in-five chance of replacing 00-03/08-0b/10-13/18-1b/20-23/28-2b/30-33/38-3b/88-8b with an alternative encoding of the same instruction. There is a one-in-four chance of replacing 28-2b/30-33 (sub/xor) with the opposite instruction when both registers are the same. There is a one-in-five chance of replacing 08-09/84-85 (or/test) with the opposite instruction when both registers are the same. These replacements are identical in nature to those in Zmist. There is also a one-in-four chance of replacing 84-87 with an alternative encoding of the same instruction.

STRETCH GOALS

The virus then extends the data section by the size of its data, plus a random amount of up to 127 bytes. The random amount of extra data is filled with random values. Next, the virus searches within the .init section for the last 'call' instruction, and appends an additional call which points to the virus code. This is how the virus gains control when an infected file is executed.

Now that the infection is complete, the virus builds a new ELF file, placing each of the sections at the appropriate location and aligning them as necessary. All references to individual sections are updated, too. It is here that the SHT_DYNAMIC section is referenced, with the assumption that it is valid. If all goes well, the code section is updated with adjusted label offsets and all branches are fixed and converted into long form (there are no short branches after infection). Then the jump tables and symbol tables are rebuilt, the new entry point value is assigned, and the virus data is encrypted.

Finally, the infection marker is set, and the file is closed. The virus then searches for the next file to infect, and the cycle repeats.

CONCLUSION

The author of Crimea, who calls himself 'herm1t', chose the name 'Lacrimae' (Latin for 'tears') for this virus. The word is used most famously in *The Aeneid*. Aeneas is overcome by the futility of warfare and the waste of human life. If only herm1t would be overcome by the wasting of his own life in this way, we might not have to deal with viruses like this.

VIRUS ANALYSIS 2

HOW TO DISABLE WFP USING PHYSICAL DISK INFORMATION

Ha Young Yang
AhnLab, Korea

Microsoft has released various file system APIs for *Windows* for the purposes of defragmentation and recovery. Unfortunately, it is possible for these APIs to be exploited for malicious purposes. Recently, a threat has appeared which obtains a file's physical disk location information with the aid of these file system APIs. First the malware calculates the physical disk location of its target file and then it modifies the file. The modification of normal system files not only disables *Windows* file protection (WFP), but also causes problems for anti-virus programs in restoring the modified system files.

BASIC INSTINCT

The piece of malware (named Win_Trojan/Rosys.49152 by *AhnLab*) was programmed in assembly language and infects the Userinit.exe file, which specifies which programs *Windows* runs when a user logs on, running logon scripts, re-establishing network connections and starting Explorer.exe.

The malicious program first calculates the physical disk location of the file and then overwrites the code with its own from the beginning to 0x1000. As a result of this overwriting operation, the system file is modified and the *Windows* system is no longer able to protect the file properly. Trojan downloader ability contained in the code written to the file means that, when booted, the system can be connected to specific websites to download malicious files.

Despite being infected, the system can continue to function normally because the modified trojan downloader has the ability to run the Explorer.exe file.

The virus has a driver file, pcihdd.sys, which stores a set of data, encrypted with a four-byte XOR key value, to be used in overwriting. When the virus calls a specific function in the pcihdd.sys file, it starts to decrypt the data and infects Userinit.exe by using an appropriate decoding buffer.

The virus works on NTFS, FAT16 and FAT32 file systems. Although it has a perfect algorithm to calculate the Userinit.exe disk location, it does not work on certain systems due to a bug which leaves the 'Carry' value out at the point of arithmetic calculation. To calculate the physical disk location of the file, the virus makes use of LCN (Logical Cluster Number) information obtained from the RETRIEVAL_POINTERS_BUFFER structure, master boot records (MBR) and boot sector.

MBR, BOOT SECTOR

Three types of information from the MBR are used to calculate the physical disk location: the boot indicator, system ID (volume type) and relative sectors.

Through the boot indicator the virus checks whether the system is Active Partition (0x80) or Logical Partition (0x0). It only infects Active Partition systems. Through the system ID value the virus also checks the type of file system, such as NTFS, FAT16 or FAT32. Relative sectors represent the total offsets from the beginning of a disk to the beginning of a volume or partition as the number of sectors.

Four types of information from the boot sector are used to calculate the physical disk location:

- Reserved sectors
- Number of FATs
- Sectors per FAT
- Sectors per cluster

The number of FATs and sectors per FAT are used only on FAT file systems.

FSCTL_GET_RETRIEVAL_POINTERS

The 'DeviceIoControl' request causes a mapping to be performed between the file address space and the volume address space. A cluster is the smallest allocation unit that the file system will use when allocating physical storage for a file. The VCN (virtual cluster number) provides ordering information about the file from '0' to 'N' expressed in units of cluster, and the LCN (logical cluster number) provides ordering information about the volume from the beginning to the end expressed in units of cluster. Each VCN value is mapped to the corresponding LCN value, which is subsequently translated into the physical byte offset of a volume.

Figure 1 illustrates the three levels of translation, from file byte offsets to virtual cluster block and then to volume logical cluster block.

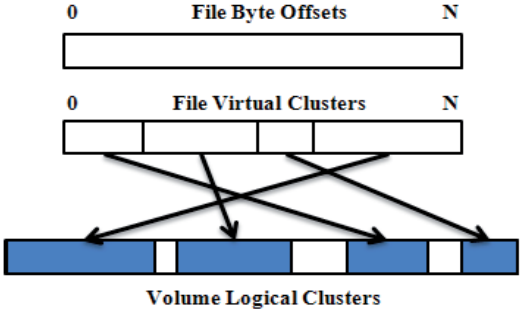


Figure 1: Three levels of translation.

To calculate the disk location the virus performs a 'DeviceIoControl' request, as shown below, and applies 'FSCTL_GET_RETRIEVAL_ POINTERS' to the control codes and the handle of Userinit.exe:

```
BOOL DeviceIoControl(
        (HANDLE)hDevice,
         FSCTL_GET_RETRIEVAL_POINTERS,
        (LPVOID)lpInBuffer,
        (DWORD)nInBufferSize,
        (LPVOID)lpOutBuffer,
        (DWORD)nOutBufferSize,
        (LPDWORD)lpBytesReturned,
        (LPOVERLAPPED)lpOverlapped
);
```

As a result of this request, the virus obtains structure values including LCN and VCN as shown below:

```
typedef struct RETRIEVAL_POINTERS_BUFFER{
        (DWORD)ExtentCount;
        LARGE_INTEGER StartingVcn;
        struct {
                LARGE_INTEGER NextVcn;
                LARGE_INTEGER Lcn;
        } Extents[1];
}RETRIEVAL_POINTERS_BUFFER,
*PRETRIEVAL_POINTER_BUFFER;
```

The following calculation algorithms are applied to obtain the disk location information for each file system. The location information expressed in units of cluster is translated to the byte offset value prior to being used.

- NTFS

  X = relative sectors + reserved sectors

  Y = LCN * (sectors per cluster)

  Disk offset = (X + Y) * SECTOR_SIZE

- FAT16, FAT32

  X = relative sectors + reserved sectors

  Y = LCN * (sectors per cluster)

  K = (number of FATs) * (sectors per FAT)

  Disk offset = (X + Y + K) * SECTOR_SIZE

Figure 2 shows the structure information on RETRIEVAL_ POINTERS_BUFFER obtained by the 'DeviceIoControl' request. It can be determined that the LCN value of Userinit.exe, applied to calculate the real disk location information, is 0x80B7B.

By applying both the calculation algorithm and the LCN information, the disk location information of the real 'Userinit. exe' is calculated as follows in a FAT32 environment:

X = 0x3F + 0x26 (=0x65)

Y = 0x80B7B * 0x20 (=0x1016F60)

K = 0x02 * 0x270D (=0x4E1A)

Disk offset = 0x101BDDF * 0x200(sector size)

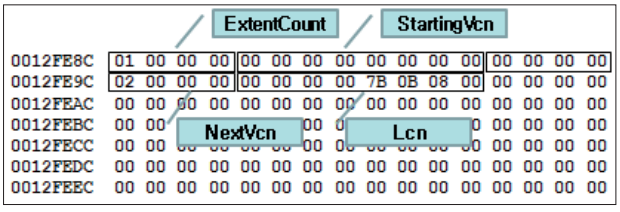


Figure 2: Structure information on 'Userinit.exe' obtained by a 'DeviceIoControl' request.

The disk offset value obtained through the final calculation is 0x2037BBE00. The virus reads a data block of size 0x200 by applying both the disk handle value obtained from '\\.\PhysicalDrive0' and the byte offset value, 0x2037BBE00. It also performs a verification process in which the disk data is compared with the Userinit.exe file data, byte by byte.

INFECTION

When the verification process has been completed successfully, the virus starts to overwrite its code on the corresponding disk location. The data in the pcihdd.sys file, which is encrypted with a four-byte XOR key value (0x3F702D98), is used in this overwriting process. Figures 3 and 4 show the data before and after decryption respectively.

The websites connected to by the action of the overwritten code in Userinit.exe change with every mutation. The following are the sites known to date:

- http://yu.8s7.net/cert.cer
- http://3.joppnqq.com/test.cer
- http://1.jopmm99.com/test.cer

VIRUS BUG

The calculation algorithm applied in the virus is designed to run on FAT16, FAT32 and NTFS environments. But, as mentioned, the virus may not work on certain systems as a result of not considering the 'Carry' value at the point of arithmetic calculation. 'Carry' is generated at the final calculation step of 0x101BDDF * 0x200, when the sector size is multiplied.

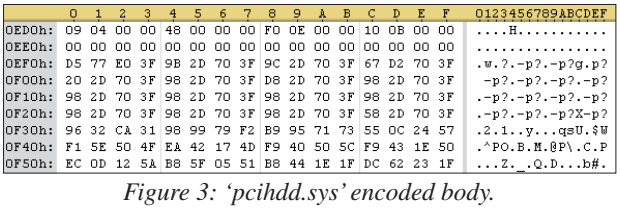


Figure 3: 'pcihdd.sys' encoded body.

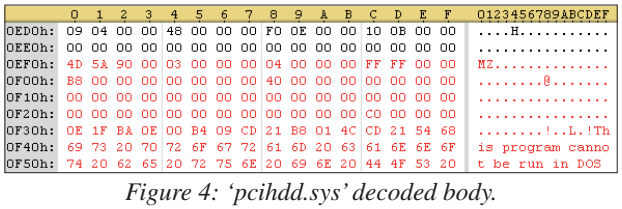


Figure 4: 'pcihdd.sys' decoded body.

The extract shown below shows real codes used in the virus. Bugs are generated at 'IMUL EAX, EAX, 200' codes. To correct the problem, the 'ADC EDX, 0' command should have been added at the end of the arithmetic calculation. In view of the fact that 'Carry' is considered in earlier calculation steps, its absence from the final calculation step could be interpreted as a mistake in programming.

```
// Final arithmetic calculation
IMUL   EAX, EAX, 200

// Move from the beginning of a disk to 0x2037BBE00
MOV    DWORD PTR [EBP-540], EDX
MOV    DWORD PTR [EBP-544], EAX
PUSH   0
LEA    EAX, DWORD PTR [EBP-540]
PUSH   EAX
PUSH   DWORD PTR [EBP-544]
PUSH   DWORD PTR [EBP-530]
CALL   <JMP.&kernel32.SetFilePointer>

// Read at 0x2037BBE00 offset
PUSH   0
LEA    EAX, DWORD PTR [EBP-18]
PUSH   EAX
PUSH   200
LEA    EAX, DWORD PTR [EBP-52C]
PUSH   EAX
PUSH   DWORD PTR [EBP-530]
CALL   <JMP.&kernel32.ReadFile>

// Compare "Userinit.exe" file with a data
LEA    EDI, DWORD PTR [EBP-52C]
LEA    ESI, DWORD PTR [EBP-32C]
MOV    ECX, 200
REPE   CMPS BYTE PTR [EDI], BYTE PTR[ESI]
```

CONCLUSION

There are many forms of this virus infecting *Windows* system files. Recently, a variety of techniques have been introduced to disable WFP (Ex. Patching sfc.dll, sfc_os.dll, Modifying Registry), but the virus described here is the first that we know of that modifies *Windows* system files by calculating the physical disk location of a file. It is quite possible that this will become a prevailing WFP-disabling technique in future.

The virus described here has tried to infect only the Userinit.exe file. It is hard to detect the infection of this file because Userinit.exe is executed upon booting and terminated simultaneously with the end of booting. Appropriate measures should be taken to deal with malicious programs of this type, which are becoming more sophisticated.

FEATURE

ASSESSMENT WAR: WINDOWS SERVICES

Aleksander Czarnowski
AVET, Poland

In the world of Web 2.0, Java, .NET and other hot technologies we tend to forget about the core components that make it all possible. In the case of the *Windows* platform, the base components are the kernel and the *Windows* services. In fact, Service Control Manager (SCM) can be used to load kernel modules and use all ring 0 privileges – not to mention virtualization. Indeed, not much has changed since *Windows NT 4.0*: add RPC and DCOM and we have the foundation of the *Windows* operating system.

TARGETED ATTACKS

In an enterprise environment it is common to find custom-made business applications or plug-ins to well known solutions. This opens an interesting window of opportunity for potential attackers. After years of discussing secure programming, programmers still produce bad (insecure) code – which is later tested and deployed with the highest possible privileges. Because architects have provided programmers with bad architecture, programmers use high-level privileges and testing is based on the same set of access rights as on the developers' machines. In the case of *Windows* services this means running as LocalSystem, even in *XP*, *2003* and *Vista*, which provide two additional built-in accounts for the job: NetworkService and LocalService.

In this article I will describe a simple attack scenario based on high-privilege service vulnerability. It's not a true story, but the experiences and techniques have been gathered and developed over the course of real-life assessments.

THE PLOT

Imagine the following scenario: in our corporate network we have deployed some kind of custom business application. Internally, inter-process communication is provided with the help of *Windows*-based services. Those services have network access and provide some kind of parser to gather data. Also in the environment is an internal attacker – the bad guy. He knows that intrusion prevention systems (IPS) have been deployed in the network, so trying to exploit the good old RPC-DCOM vulnerability or scanning for an 'sa' account with an empty password in *MS SQL Server* will be noticed pretty quickly and probably stopped by the IPS. He needs something 'unusual' to bypass all the protection and yet gain high privileges. The custom business application seems like an ideal potential target. One could ask why he would attack a *Windows* service – looking for SQL injection in an application web front-end would be easier and if done wisely would probably go undetected by the IPS (you should now be thinking of how to deal with SSL/TLS connections on your IPS). Let us assume, however, that our attacker is not only after the data provided by the application, but he also wants to gain high privileges and be able to penetrate the rest of our ActiveDirectory infrastructure. SQL injection might not be the best way in such a case, but it is still worth a try.

To complete the crime scene we also need a service programmer. For the reasons mentioned earlier the programmer decided to run his service with LocalSystem privileges. This has been recorded only in internal documentation, which is not available to the company's customers. Also, source code is not available to any of the company's employees. So our attacker is left with a binary file running with high privileges on *Windows Server* – or is he?

THE RECONNAISSANCE

This is the part of the attack that is usually detected by network IPS systems. However, if done slowly and carefully it could be missed by the IPS or ignored by a security officer. Our attacker needs to learn as much as he can about the server running the targeted service. The simplest method would be to use nmap to detect all of the services:

```
nmap -sS -A server_ip
```

Another great tool for the reconnaissance phase in a *Windows*-based network is *Winfingerprint*. It can detect shares, services etc. as long as the RestrictAnonymous key in the registry is set to 0 or we have sufficiently high privileges within the AD infrastructure. Fortunately, enumerating server resources from an AD user account usually provides good results.

The next step is to learn more about RPC interfaces – rpcdump from *Resource Toolkit* is a great tool for the job:

```
rpcdump.exe /s server_ip /v /i
```

If our attacker were able to log on to the target server he would also be able to gather some more information about the execution environment. The tasklist not only provides a list of all processes running, but can also provide detailed information about services:

```
tasklist.exe /svc
```

The '/SVC' switch shows the list of active services in each process. In the case of *Windows 2000* the attacker would need to use the 'tlist -s' command. It is important to note that some configurations allow remote access to the SCM database which provides similar information over the network.

ANTI-DEBUGGING

In most cases you will not find any anti-debugging techniques in custom services. The probability of dealing with compressed PE files is also low. However, if dynamic analysis goes wrong, it is worth checking if the binary is protected in some way. As noted earlier, the *Windows* service is a typical PE file, so if there is no import table or it contains only a few functions then you know that imports have been protected. This is an important observation as most of the clues we used to look for vulnerabilities were based on import table integrity. Also keep in mind that even if you use function names to identify C/C++ functions only, you may not find any calls. The reason is simple: every compiler uses inline functions, so instead of call instructions you will only find 'unwound' function code. This applies to some string functions, for example.

Returning to anti-debugging, this is a topic that could fill a book (or more), so I'll just describe the most basic technique briefly. Remember that the service programmer probably wasn't getting paid for anti-debugging code, so if you do find any in a custom service then it will probably be based on a simple technique like the IsDebuggerPresent function.

In fact, the IsDebuggerPresent method can be implemented in a number of different ways. The simplest method is based on calling the IsDebuggerPresent function from kernel32.dll. If the function returns 0 then the process is not being debugged. If you peek inside the IsDebuggerPresent function you will find some very simple code:

```
lkd> u kernel32!isdebuggerpresent
kernel32!IsDebuggerPresent:
7c813093 64a118000000  mov     eax,dword ptr
fs:[00000018h]
7c813099 8b4030        mov     eax,dword ptr [eax+30h]
7c81309c 0fb64002      movzx   eax,byte ptr [eax+2]
7c8130a0 c3            ret
```

A quick inspection of the PEB structure tells us that offset 2 is the BeingDebugged field. What is interesting is the fact that you can set this field to 0 after attaching a debugger. This is an even better method than intercepting calls to the IsDebuggerPresent function and always setting the EAX register to 0, because either a direct call to the function or invoking its code directly from the service will always provide the same result.

The IsDebuggerCode function can be even simpler – you can remove the first line as it gets a self pointer from _NT_TIB (you can look it up using the 'dt' command in *WinDbg*). So the new code may look like this:

```
mov eax, fs:[30h]
mov eax, byte [eax+2]
```

Speaking of PEB, it is worth mentioning that the NtGlobalFlag field at offset 68h is also modified if the process is being debugged. For example, FLG_HEAP_VALIDATE_PARAMETERS will be set. This can also be used for debugger detection. For a good review of different anti-debugging techniques in *Windows* see [1].

AUDITING THE SERVICE BINARY

Auditing *Windows* services is a bit different at first from auditing normal native applications. First of all, services are not run directly but with the help of SCM. Secondly, every service has at least two entry points. Inside the service binary is just a plain PE console application. What makes it different is a call to the StartServiceCtrlDispatcher() function. This function takes only one parameter: lpServiceTable.

lpServiceTable is a pointer to an array of SERVICE_TABLE_ENTRY [2] structures containing one entry for each service that can execute in the calling process. The members of the last entry in the table must have NULL values to designate the end of the table.

SERVICE_TABLE_ENTRY has the following structure:

```
typedef struct _SERVICE_TABLE_ENTRY {
  LPTSTR lpServiceName;
  LPSERVICE_MAIN_FUNCTION lpServiceProc;
} SERVICE_TABLE_ENTRY,
  *LPSERVICE_TABLE_ENTRY;
```

The most important is the lpServiceProc argument which points to the ServiceMain function, which is the real entry point for the particular service. So, to find all entry points in the service we first need to locate SERVICE_TABLE_ENTRY. This is trivial if you use *IDA Pro* – just find all references to StartServiceCtrlDispatcher() and you will have the lpServiceTable pointer. You don't even need to do it manually, as the following *IDC* script will do it for you:

```
auto ea, ref;
ea = LocByName("StartServiceCtrlDispatcher");
if(ea != BADADDR)
{
  if(GetFunctionFlags(ea) != -1)
  {
    Message("\nfound function at %8X:\n", ea);
    for(ref=RfirstB(ea); ref != BADADDR;ref=RnextB(ea,
ref))
    {
        Message(" + called from %s (0x%8X)",
GetFunctionName(ref), ref);
    }
  }
  else
```

```
    Message("No StartServiceCtrlDispatcher function
found in imports.\n");
}
else
  Message("No StartServiceCtrlDispatcher function
found in imports.\n");
```

We also need to take a look at how the service starts. To do this we need to locate the CreateService() function within the audited binary. Here is the function prototype:

```
SC_HANDLE WINAPI CreateService(
  __in        SC_HANDLE hSCManager,
  __in        LPCTSTR lpServiceName,
  __in_opt    LPCTSTR lpDisplayName,
  __in        DWORD dwDesiredAccess,
  __in        DWORD dwServiceType,
  __in        DWORD dwStartType,
  __in        DWORD dwErrorControl,
  __in_opt    LPCTSTR lpBinaryPathName,
  __in_opt    LPCTSTR lpLoadOrderGroup,
  __out_opt   LPDWORD lpdwTagId,
  __in_opt    LPCTSTR lpDependencies,
  __in_opt    LPCTSTR lpServiceStartName,
  __in_opt    LPCTSTR lpPassword
);
```

We are mainly interested in three arguments: dwServiceType, lpServiceStartName and lpPassword. Sometimes – but not very often – you can find a clear text password using the lpPassword pointer. Usually, however, it is an empty string as one of the system accounts is being used. The dwServiceType will tell us if it is the kernel of a user-mode service. In addition, we need to check how the service is being run inside the system – whether as a separate process or not:

- SERVICE_WIN32_OWN_PROCESS will be specified if the service is running within its own process.
- SERVICE_WIN32_SHARE_PROCESS will be specified if the service is sharing a process with other services.

Also, if one of the above options is used we need to check for SERVICE_INTERACTIVE_PROCESS. If it is set then we are dealing with a service that is using the LocalSystem account [3] – a perfect target for exploiting. Also, if lpServiceStartName is NULL or NT AUTHORITY\LocalService, CreateService will use the LocalService account.

Now, once we have identified all entry points and possibly the privileges used by the service we can look further for vulnerabilities.

WHAT TO LOOK FOR

The methods used by our attacker in the reconnaissance phase should be sufficient to identify possible remote attack vectors. However, sometimes the service does not run on the server – instead it is installed on the workstations that use the custom application. In such cases an attacker will have to analyse the service execution environment and enumerate its DACLs. *ProcessExplorer* is the tool for this task – it allows the attacker to check if a low privilege account like 'Everyone' has the relevant permissions to access service objects. This could be another possible local attack vector.

Now, if the service is using objects it is probably also using the SetSecurityDescriptorDacl() function. A quick check of the import table will give us all the information we need (if the binary is not compressed and the import table is not obfuscated). Assuming we have found SetSecurityDescriptorDacl(), let's take a look at the arguments passed to it. If the pDacl argument is NULL then we have probably found an exploitable vulnerability. The same method has been used successfully against *Oracle Database Server 10gR2* for *Windows* [4]. You can also look for other security-related functions that take NULL parameters – every one of them will increase the service attack surface, which is a good thing from the attacker's perspective.

Next it's time for some fuzzing. We should fuzz all interfaces. Before fuzzing it is good practice to attach a debugger to the target if it is possible. In the case of an attack this will not always be possible, however during a legitimate security assessment this should not be a problem. There is one problem, however, in the case of services that start during system boot. Under *Windows 2000* you cannot attach a debugger to a process and detach it later without terminating the target. If the attacker is not able to attach a debugger to the service, how can he find vulnerabilities remotely? There are several possibilities. The most simple and effective is to measure response times – if after a certain request the delay in receiving a reply is longer than usual, this could be something interesting. Sometimes the attacker will be able to crash the service and it will not be restarted automatically.

The process of fuzzing is directly connected with the protocols used by our target. A lot of services use well known protocols like HTTP or RPC for communication, so writing a fuzzer is not a hard task. Some protocols – even internal ones – use some form of authentication. In many cases authentication is based on a static password which is hard-coded somewhere in the service or other parts of the application. If the attacker is lucky the password will be transmitted in clear text over the network. In such a case any sniffer will do the job.

THE DEBUGGERS

In the good old days everyone used the *SoftICE* debugger from *NuMega* (later from *Compuware Corporation*), but some time ago *SoftICE* became defunct and now almost everyone uses *WinDbg* from *Microsoft*. While *WinDbg* is one of very few tools that allows kernel-level debugging and works on x64 systems too, in the case of ring 3 applications there are more options available.

There are at least two user-mode debuggers worth mentioning: *OllyDBG* and *Immunity Debugger*. In fact, the latter is based on *OllyDBG* code. *Immunity Debugger* is interesting because it is one of the very first debuggers to target not bugs but vulnerabilities. Some of its extensions take it one step further: their aim is to speed up exploit development. So if you need to write an exploit for a custom user-mode service, then *Immunity Debugger* is worth checking out. It also supports command line and is integrated with Python so you can use your own or third-party Python modules.

When talking about security one must not forget *IDA Pro* – this great dissembler also offers local and remote debugging. Using *IDA* databases can be convenient if more than one person is working on a project. In reality, though, the whole binary audit is usually performed by just one reverse engineer – it's hard to organize the work within teams because you simply cannot divide tasks per address range within an application. So a simple rule: one binary object, one person, makes a lot of sense here. There is, of course, the *IDA Sync* plug-in that allows the work of multiple analysts to be synchronized, but in real life when you are working on a project it is not that easy. No plug-in will quickly synchronize the knowledge about objects across a team.

We have had a few experiences in which, for various reasons, no third-party product could help us out. The reasons included bugs inside software, the length of time needed to implement extensions, etc. This takes us to debugging frameworks like *PaiMai* [5], but the same problems can apply. So sometimes the best option is write a small debugger yourself. *Windows* has a very nice set of APIs for debugging purposes. Its documentation is far from perfect as it is missing a lot of detail, which means a lot of time must be spent reading header files from SDK and browsing the web. One of the most important things to remember is that the initial breakpoint set by CreateProcess with the DEBUG_* flag enabled is not the first instruction of the application. One of the best strategies is to handle the initial breakpoint event and set up another breakpoint (the most obvious, trivial and simple method is to insert INT 3 opcode at the entry point). When the initial breakpoint is hit, your process sections are already in memory so it is possible to write to and read the code section. Keep in mind that *Windows* enforces memory protection, so before any write operation use VirtualQueryEx and VirtualProtectEx to disable and later re-enable page write protection. The following is an example (in assembly language):

```
invoke VirtualQueryEx, stDE.u.CreateProcessInfo.
hProcess, stDE.u.CreateProcessInfo.lpStartAddress,
addr mbi, SIZEOF MEMORY_BASIC_INFORMATION

invoke ReadProcessMemory, stDE.u.CreateProcessInfo.
hProcess, stDE.u.CreateProcessInfo.lpStartAddress,
addr initalbpbuf, 1, NULL

invoke VirtualProtectEx, stDE.u.CreateProcessInfo.
hProcess, stDE.u.CreateProcessInfo.lpStartAddress,
mbi.RegionSize, PAGE_EXECUTE_READWRITE, addr mbi.
Protect

[...]

invoke VirtualProtectEx, stDE.u.CreateProcessInfo.
hProcess, mbi.BaseAddress, mbi.RegionSize, mbi.Pro-
tect, addr dwOldProtect
```

Another strategy for stopping at the application entry point is to handle the CREATE_PROCESS_DEBUG_EVENT event and set up a breakpoint at this point.

When modifying a code section remember to flush the instruction cache:

```
invoke FlushInstructionCache, stDE.
u.CreateProcessInfo.hProcess, stDE.
u.CreateProcessInfo.lpStartAddress, 1
```

You might be wondering why the above examples are written in assembly language. Actually, if you really need a lightweight tool, assembly is the way to do it. You can have quite a useful debugging tool in less than 10 kilobytes, which is really lightweight and it leaves almost no footprint in the system. One final tip: if you have a lot of time you can write your tools using FASM assembler. FASM is a great tool, but unfortunately it is missing some headers and definitions from *Windows* SDK so you have to write them yourself. While personally I prefer FASM, I must admit that MASM32 is better suited for this task if you need to dive in quickly. MASM32 has all the headers you will need.

TOPSTACK METHOD

Since we are talking about vulnerabilities it is reasonable to take a look at shellcode. Due to the *Windows* architecture, when executing ring 3 shellcode the attacker needs to know the address of at least two functions inside kernel32.dll: LoadLibrary and GetProcAddress. With those two addresses he is able to locate any other function address he needs inside the shellcode. As we are talking about a targeted attack one could argue that, thanks to the 'nmap -A' switch, the attacker will know exactly what system version he is attacking. Thanks to this information he will be able to hard code all the addresses for the functions he needs to call from his shellcode. However, even in the case of targeted attacks, attackers still look for reliability

(reliability is more important in targeted attacks than in the old script-kiddie-style attack when trying to exploit a few thousand hosts). One of the most reliable methods of finding the LoadLibrary/GetProcAddress function addresses is a method called TOPSTACK. This is a relatively new method, so I believe it is worth describing.

TOPSTACK is a method of finding kernel32.dll in memory. We need it to get the addresses of LoadLibrary and GetProcAddress so that we can use those functions later to get the addresses of other *Windows* API functions required by our shellcode:

```
   xor eax, eax
   mov eax, fs:[eax + 18h]  ;get TEB address
   mov esi, eax             ;store it at ESI register
   lodsd                    ;add 4 to ESI
   lodsd                    ;grab the top of stack
   mov eax, [eax - 1Ch]     ;this pointer is address
                            inside kernel32.dll
loop:
   dec eax                  ;scan memory at 64kb
                            boundary
   xor ax, ax
   cmp word ptr [eax], 5A4Dh  ;check for MZ signature
                              (start of PE file)
   jnz loop                 ;nope - search further
```

The TOPSTACK method has several advantages:

- It can occupy around 25 bytes of memory.
- It works on *NT*, *2000*, *XP* and *2003*.
- It works reliably, thanks to its simplicity.
- The example shown above is free from bad bytes, so it can be used right away.

Actually, the example above can be optimized further – but I will leave that as an exercise for the reader (as a tip, take a look at how the ESI register is being used). To understand fully how it works we need to take a look at two *Windows* structures: TEB (Thread Environment Block) and TIB (nt!_NT_TIB for those using *WinDbg*). TEB is always located at address fs:0 and its layout is as follows:

```
lkd> dt nt!_TEB
   +0x000 NtTib                      : _NT_TIB
   +0x01c EnvironmentPointer         : Ptr32 Void
   +0x020 ClientId                   : _CLIENT_ID
   +0x028 ActiveRpcHandle            : Ptr32 Void
   +0x02c ThreadLocalStoragePointer  : Ptr32 Void
   +0x030 ProcessEnvironmentBlock    : Ptr32 _PEB
```

Please note that we are talking about 32-bit systems – on x64 the _NT_TIB structure ends at address 38h and PEB is located at 60h. Now let's take a look at _NT_TIB:

```
lkd> dt nt!_NT_TIB
   +0x000 ExceptionList     : Ptr32 _EXCEPTION_
                             REGISTRATION_RECORD
   +0x004 StackBase         : Ptr32 Void
   +0x008 StackLimit        : Ptr32 Void
   +0x00c SubSystemTib      : Ptr32 Void
   +0x010 FiberData         : Ptr32 Void
   +0x010 Version           : Uint4B
   +0x014 ArbitraryUserPointer   : Ptr32 Void
   +0x018 Self              : Ptr32 _NT_TIB
```

As you can see it starts with an exception record – this is why the SEH handler is installed using the mov fs:[0] instruction. At offset +4 we have a pointer to the stack base which we will use in our method. Using the top of the stack and going down 1Ch bytes we find an address that lies somewhere inside kernel32.dll.

After finding the start of kernel32.dll we just need to extract data from the export table, and voilà! We can start calling all *Windows* API functions.

THE FINAL SCENE

With all the tools and methods presented here, an attacker would be able to perform a successful targeted attack against most custom business applications. Of course, the aim of this article was not to educate the attacker but to provide readers with tools for auditing closed-source *Windows* services. We cannot afford to forget about the building blocks of our infrastructure because it leads to exploitable vulnerabilities. It also leads to a loss of compliance and in the world of Sarbanes-Oxley, PCI and BASEL II this could mean financial losses that are more significant than the consequences of an attack itself.

To be prepared for an attack you need to think like the attacker. Penetration testing strengthened by an application audit is a wise investment.

BIBLIOGRAPHY

[1] Falliere, N. Windows Anti-Debug Reference, SecurityFocus. http://www.securityfocus.com/infocus/1893.

[2] http://msdn2.microsoft.com/en-us/library/ms686001(VS.85).aspx.

[3] http://msdn2.microsoft.com/en-us/library/ms682450(VS.85).aspx.

[4] Cerrudo, C. Practical 10 Minutes Security Audit Oracle Case. http://www.blackhat.com/presentations/bh-dc-07/Cerrudo/Presentation/bh-dc-07-Cerrudo-ppt.pdf.

[5] http://paimei.googlecode.com/.

CALL FOR PAPERS

VB2008 OTTAWA

Virus Bulletin is seeking submissions from those wishing to present papers at VB2008, which will take place 1–3 October 2008 at the Westin Ottawa, Canada.

The conference will include a programme of 40-minute presentations running in two concurrent streams: Technical and Corporate. Submissions are invited on all subjects relevant to anti-malware and anti-spam.

In particular, *VB* welcomes the submission of papers that will provide delegates with ideas, advice and/or practical techniques, and encourages presentations that include practical demonstrations of techniques or new technologies.

SUGGESTED TOPICS

The following is a list of topics suggested by the attendees of VB2007. Please note that this list is not exhaustive – the selection committee will consider papers on any subjects relevant to the anti-malware community.

- Forensics
- Non-*Windows* malware
- Demonstrations of malware in action
- Mobile threats
- Analysis tools
- Botnets
- Fast-flux network threats
- Banking trojans
- Rootkits
- Behavioural detection & behaviour blocking
- Virtualization
- Network-based malware control (IDS/IPS)
- Search engines in research/vulnerability assessment
- Targeted attacks
- Data mining and analysis
- Spyware
- Pattern matching
- Formal mathematical approaches
- Zombie networks
- Obfuscation methods
- Reverse engineering
- Automation in sample gathering, processing and analysis
- Wireless security
- Unpackers/emulators
- Server-side polymorphism
- Anti-malware testing
- Whitelisting/application control
- Infection case studies (corporate and technical)
- Maintaining layered defence in the enterprise
- Attack scenarios – how to handle them
- End-user impact and statistics
- Social engineering
- Law enforcement and legal aspects of spam & malware
- Phishing & anti-phishing techniques
- Anti-spam performance testing
- Managing spam in a corporate environment
- Latest anti-spam techniques

HOW TO SUBMIT A PROPOSAL

Abstracts of approximately 200 words must be sent as plain text files to editor@virusbtn.com no later than **Friday 7 March 2008**. Please include full contact details with each submission and indicate whether the paper is intended for the technical or the corporate stream.

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

Authors are advised that, should their paper be selected for the conference programme, the deadline for submission of the completed papers will be Monday 9 June 2008. Full details of the paper submission process are available at http://www.virusbtn.com/conference/vb2008/call/.

LAST-MINUTE PRESENTATIONS

In addition to the 40-minute presentations, a portion of the technical stream will be set aside for 20-minute, 'last-minute' technical presentations, proposals for which need not be submitted until three weeks before the start of the conference. Presenting a full paper will not preclude an individual from being selected to present a last-minute presentation. Further details will be released in due course.

COMPARATIVE REVIEW

WINDOWS SERVER 2003

John Hawes

While *VB*'s past comparative reviews on server platforms have generally been less heavily subscribed than desktop tests, this month sees the continuation of the recent upward trend in the number of products taking part, with a total of 27 products on the test bench. While some vendors submitted dedicated server, or at least business-oriented versions of their products, several entries comprised much the same products as appear in desktop platform tests, which should be assumed to work perfectly well in the *Server 2003* environment.

With time pressing (a post holiday season illness meaning things got under way even later than originally planned), I could only hope for simple installation procedures, easily navigated configuration systems and solid, stable operation. Past experience has, of course, taught me that this was a little too much to hope for, but I went into the lab with my fingers crossed.

PLATFORM AND TEST SETS

Windows Server 2003 bears great similarity to *XP* (on which it is based) – with a number of adjustments to the default settings providing a little extra security – and the process of setting up the test systems presented few difficulties.

The deadline for product submission was the first Monday of the year, 7 January, with the content of the test sets frozen the preceding Friday. Rather hasty pre-Christmas preparations for the review meant that my usual check of significant calendar events was omitted, and the product submission deadline coincided unwittingly with Russian Orthodox Christmas celebrations and Christmas in some other areas, but vendors based in these territories still managed to get their products in without too much grumbling.

The test sets were based on the November issue of the WildList (released in mid-December), which included a fairly standard number of additions heavily dominated by worms with familiar names, or at least behaviours. There were once again a handful of polymorphic file-infectors, including several of the W32/Virut variants which caused such mayhem in the last test. A fairly large number of items also fell from the list and were thus relegated to other test sets.

These other sets were subject to minimal updating, due to the shortage of time for preparations, and the clean set was also expanded in only a minor way, with a few dozen packages and their contents added. With limited changes from the test sets used in the last round of testing, I hoped for considerably better performance from the products this time around.

In addition to testing basic detection performance, we have once again included tests of the products' archive scanning depth, both in default settings and with more complex scanning options enabled. Only products which could be cajoled into detecting the EICAR test sample hidden three levels deep in archives are included in the tables for these sets, and only those spotting the test string in a file with a randomly selected extension appear on the 'all files' tables (although in some cases this only indicates that products are getting file type information from within files rather than simply from the extension, and full scanning may not always be occurring). We hope that the data provides some insight into the efficiency of the products under test.

AEC Trustport Antivirus 2.8.0.1628

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	3

AEC's *Trustport* suite contains a number of items beyond the anti-virus component, but as usual only this module was submitted for testing. This made installation a pretty straightforward process, and left me with no main interface from which to operate – configuration and functions are instead accessed from a system tray menu. The default settings are pretty thorough, detecting everything in our archive and file-extension scanning test set, and combined with the multi-engine layout this led to some rather languid scanning times.

AEC's entry in the last comparative review (see *VB*, December 2007, p.16), its first since the *BitDefender* engine was dropped from the product in favour of those of *Dr.Web* and *VirusBlokAda*, suffered from some false positive issues as well as several WildList misses. Detection was greatly improved this time, with nothing at all missed on demand, and only a few older items missed on access (where not all the available engines are used by default). However, despite one of the engines apparently being disabled entirely, and greyed out in configuration dialogs, several false positives were recorded, which once again deny *AEC* a VB100 award.

Agnitum Outpost Security Suite Pro 6.0.2227.232.0465

ItW	99.80%	**Worms & bots**	99.91%
ItW (o/a)	99.80%	**DOS**	99.77%
File infector	99.21%	**Macro**	100.00%
Polymorphic	85.91%	**False positives**	0

Agnitum's *Outpost* was subject to an in-depth review last month (see *VB*, January 2008, p.17), after achieving its first VB100 certification in the previous comparative. With the review still fresh in my mind, the installation process and configuration were fairly straightforward, although the product is sufficiently well designed to present few difficulties for those without any prior knowledge.

The available configuration is somewhat limited, with no option to scan archives in on-access mode, but other files did seem to be inspected regardless of their extension, and speeds were fairly reasonable considering. False positives were entirely absent, and detection in most of the test sets at the pretty high level expected from the *VirusBuster* engine in use. In the WildList set, however, a single instance of a W32/VB worm was missed, as well as two samples of one of the new W32/Virut variants. This presaged problems for some of the products further down the list using the same technology, and meant *Agnitum* didn't quite manage to add to its VB100 tally.

AhnLab V3Net for Windows Server 6.1.21.711

ItW	100.00%	**Worms & bots**	99.70%
ItW (o/a)	100.00%	**DOS**	97.18%
File infector	98.95%	**Macro**	98.99%
Polymorphic	92.88%	**False positives**	0

AhnLab has not been a regular participant in VB100 tests recently, but the AVAR conference the company hosted in Seoul a few weeks before the test deadline provided an opportunity to pester the developers into joining in again – an effort which paid off with this entry.

The *V3Net* product is quick and easy to install and set up, with a clear and pleasant interface adorned with a touch of cartoonishness without seeming too silly. The configuration is a little lacking on access, with no option to delve inside archives in this mode – something which seemed a little odd in a dedicated server product, as one might expect experienced admins to be interested in having a fuller range of options available. Even in on-demand mode, where most archive types were examined quite deeply, self-extracting executables and installer files were omitted. Another oddity which may cause admins frustration is the format of logs, which record only filenames with no information as to where the files in question may be found – this made for considerably more work in processing the test results.

Detection itself was less of an issue. No false positives were recorded and, despite a handful of misses in some of the older test sets, nothing significant was missed in the WildList set, thus *AhnLab* earns a VB100 award on its return to the test bench.

Alwil avast! Server Edition 4.7.865

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	98.67%
File infector	100.00%	**Macro**	99.98%
Polymorphic	86.99%	**False positives**	0

The server version of *avast!* seems little different from the standard version, or at least from the 'enhanced' interface usually necessary for the VB100 test. All the required configuration was readily available, with the defaults set not to scan archives internally but options available to scan the full range of archive types included in our test sets. Oddly, the renamed version of the EICAR test file was spotted on access but not on demand, implying that the on-access scanner is set up a little more thoroughly than the normally more in-depth manual scans.

Speeds were impressive, and still fairly decent with the more complete scanning options enabled. Detection levels were reasonable across the sets, with nothing at all missed in the WildList set. With no false positives either, *Alwil* wins another VB100 award.

Avira AntiVir Server 8.00.00.1547

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	99.78%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.87%	**False positives**	0

Despite an installation process which seemed very familiar, after the required reboot *AntiVir*'s *Server* edition displayed significant differences from the desktop variant, with an MMC-based console provided for most of the required configuration options. The interface was not as simple to navigate and use as *Avira*'s desktop range, but seems to provide a pretty thorough range of controls for the administrator. On-access scanning was fairly straightforward, and thorough once fuller scanning was enabled, although a few files compressed with the ACE algorithm were missed despite more deeply nested samples of the same format being detected.

Some very good speeds were recorded in both modes, although the actual setup and running of on-demand scans

On-demand tests	WildList		Worms and bots		DOS		File infectors		Macro		Polymorphic		Clean sets	
	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	FP	Susp.
AEC Trustport	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	3	
Agnitum Outpost	3	99.80%	2	99.91%	20	99.77%	8	99.21%	0	100.00%	220	85.91%		
AhnLab V3Net	0	100.00%	5	99.70%	656	97.18%	2	98.95%	46	98.99%	544	92.88%		
Alwil avast!	0	100.00%	0	100.00%	1022	98.67%	0	100.00%	1	99.98%	664	86.99%		
Avira AntiVir	0	100.00%	0	100.00%	32	99.78%	0	100.00%	0	100.00%	3	99.87%		
BitDefender Security	0	100.00%	0	100.00%	8	99.78%	2	98.95%	3	99.93%	0	100.00%		2
CA eTrust	0	100.00%	0	100.00%	235	99.67%	1	99.74%	12	99.82%	9	99.64%		
Doctor Web Dr.Web	4	99.28%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		9
ESET NOD32	0	100.00%	0	100.00%	500	99.78%	0	100.00%	0	100.00%	0	100.00%		
Fortinet Forticlient	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Frisk F-PROT	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.95%		
F-Secure Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.97%		3
Grisoft AVG	0	100.00%	2	99.91%	197	99.10%	7	98.43%	0	100.00%	695	78.55%		
Ikarus Virus Utilities	37	99.55%	4	99.60%	2460	91.37%	19	96.28%	151	96.45%	365	82.05%	8	
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.97%		
Kingsoft Anti-virus	19	99.26%	639	16.85%	14050	12.26%	114	71.83%	355	91.56%	2020	38.49%		
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Microsoft Forefront	0	100.00%	0	100.00%	0	100.00%	1	99.90%	0	100.00%	80	96.46%		
MWTI eScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.97%		2
Norman Virus Control	4	99.95%	0	100.00%	269	99.12%	7	99.15%	0	100.00%	706	84.20%	1	
PCTools AntiVirus	3	99.80%	2	99.91%	20	99.77%	8	99.21%	0	100.00%	220	85.91%		
Quick Heal AntiVirus Lite	0	100.00%	0	100.00%	1149	95.23%	17	97.64%	73	98.23%	1081	81.86%	5	
Redstone Redprotect	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.97%		2
Sophos Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	8	99.80%	0	100.00%		22
Symantec Endpoint Protection	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
VirusBuster for Windows Servers	1	99.82%	2	99.91%	20	99.77%	8	99.21%	0	100.00%	220	85.91%		
Webroot SpySweeper with AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	6	99.93%	0	100.00%		3

took much more time, with a rather awkward and fiddly setup process, and no indication of scanning progress at all. Once the complexities of the design were cracked, scan results showed the product's usual excellent detection rates and no false positives, giving *Avira* another VB100 award.

BitDefender Security for Windows Server 2.4.227

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	99.78%
File infector	98.95%	**Macro**	99.93%
Polymorphic	100.00%	**False positives**	0

BitDefender also provided a special server version for this test, again incorporating a console interface using the MMC framework. This seemed rather more logically laid out and took less effort to decipher, but also seemed to be missing some useful options. The on-access scanner, for example, seemed to offer no option to block access only, making this action available only after attempts at other 'cleaning' methods had failed. This resulted in my test collection being trashed and requiring restoration between tests. Another apparent failing was an issue with setting up on-demand scans. Assuming at first that these could again only be run from the scheduler, I set up a scan using the default time offered, which was in fact the current time – ideal for my needs. However, by the time the setup process had finished, the moment had passed and the scan thus failed to initiate, waiting instead for the same time to roll around the following day. My frustration was quickly sidestepped when I found the proper place to run manual scans, with a 'scan now' option available.

Having deciphered the interface, testing continued without further stumbles, with fairly good speeds and the default settings covering most file types in depth. Detection was pretty close to flawless across the test sets including the WildList, and in the clean sets a few items were flagged as adware but no false positives were recorded, granting *BitDefender* a VB100 award.

CA eTrust Antivirus 8.1.6370

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	99.67%
File infector	99.74%	**Macro**	99.82%
Polymorphic	99.64%	**False positives**	0

CA's *eTrust* is a corporate-focused product, and has been submitted in much the same form for just about all VB100 tests I have run. This month was no different, and the familiar interface, its frustrations of slow connection times slightly less intrusive than usual, powered through the tests in splendid time. On-access archive scanning appeared to be absent, despite a number of options relating to such scanning being activated – single-level zip and jar archives were penetrated in this mode, but no other types or greater depths. On-demand scanning proved more thorough, although ACE and self-extracting EXEs were only probed one level deep.

Detection levels were very high, with almost complete coverage across the test sets and the WildList covered without difficulty. Without false positives *CA* easily makes the grade required for a VB100 award.

Doctor Web Dr.Web Antivirus for Windows Server 4.44.1.01090

ItW	99.28%	**Worms & bots**	100.00%
ItW (o/a)	99.28%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	0

Doctor Web's product presented the same slick and solid design which impressed me in the last test, although the rather basic font used in the installer looks slightly out of place in its glossy surroundings. The clear layout of the interface made testing smooth and problem-free, with sensible defaults and deep configuration available. A few times on shutting down the on-access scanner there were error messages that claimed there were issues with disabling the protection, but it certainly seemed to have closed properly and restarted without further problems.

Scanning speeds were excellent, particularly in the default mode, which uses a 'smart' setting to determine which files are worth scanning. With thorough scanning of all files enabled things slowed down somewhat, but detection was pretty good across the board, with no more than a few files missed in each set, most of them down to file types not scanned by default. No false positives were in evidence, but unfortunately for *Doctor Web* a few items added to the latest WildList were not covered, and the VB100 award remains just out of reach.

ESET NOD32 Antivirus 3.0.621.0

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	99.78%
File infector	100.00%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	0

On-access tests	WildList		Worms and bots		DOS		File infectors		Macro		Polymorphic		Clean sets	
	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	FP	Susp.
AEC Trustport	0	100.00%	0	100.00%	90	99.78%	0	100.00%	0	100.00%	553	90.61%	2	
Agnitum Outpost	3	99.80%	2	99.91%	20	99.77%	10	98.69%	0	100.00%	220	85.91%		
AhnLab V3Net	0	100.00%	5	99.70%	656	97.18%	4	98.95%	46	98.99%	544	92.88%		
Alwil avast!	0	100.00%	0	100.00%	1022	98.67%	0	100.00%	4	99.93%	664	86.99%		
Avira AntiVir	0	100.00%	0	100.00%	32	99.78%	0	100.00%	0	100.00%	3	99.87%		
BitDefender Security	0	100.00%	2	99.96%	8	99.78%	4	98.43%	1	99.98%	0	100.00%		2
CA eTrust	0	100.00%	0	100.00%	235	99.67%	3	99.21%	12	99.82%	9	99.64%		
Doctor Web Dr.Web	4	99.28%	2	99.72%	0	100.00%	2	99.48%	0	100.00%	0	100.00%		6
ESET NOD32	0	100.00%	0	100.00%	500	99.78%	0	100.00%	0	100.00%	0	100.00%		
Fortinet Forticlient	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Frisk F-PROT	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.95%		
F-Secure Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.97%		1
Grisoft AVG	0	100.00%	2	99.91%	197	99.10%	9	97.90%	3	99.93%	695	78.55%		
Ikarus Virus Utilities	37	99.55%	4	99.60%	2460	91.37%	19	96.28%	159	96.26%	365	82.05%	8	
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	2	99.48%	0	100.00%	1	99.97%		
Kingsoft Anti-virus	19	99.26%	639	16.85%	14050	12.26%	114	71.83%	355	91.56%	2020	38.49%		
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Microsoft Forefront	0	100.00%	0	100.00%	0	100.00%	3	99.38%	0	100.00%	80	96.46%		
MWTI eScan	0	100.00%	0	100.00%	2	100.00%	0	100.00%	0	100.00%	0	100.00%		2
Norman Virus Control	8	99.90%	0	100.00%	269	99.12%	9	98.62%	8	99.80%	865	79.21%	1	
PCTools AntiVirus	3	99.80%	2	99.91%	22	99.55%	10	98.69%	0	100.00%	220	85.91%		
Quick Heal AntiVirus Lite	0	100.00%	0	100.00%	1197	95.12%	20	96.72%	82	98.04%	1081	81.86%	5	
Redstone Redprotect	0	100.00%	0	100.00%	0	100.00%	2	99.48%	8	99.80%	0	100.00%		2
Sophos Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	8	99.80%	0	100.00%		22
Symantec Endpoint Protection	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
VirusBuster for Windows Servers	1	99.82%	2	99.91%	20	99.77%	10	98.69%	0	100.00%	220	85.91%		
Webroot SpySweeper with AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	6	99.93%	0	100.00%		3

The latest incarnation of *ESET*'s product was reviewed on its release a few months ago (see *VB*, November 2007, p.19), and received some rather effusive praise for its stylish looks and smart design. As *NOD32* version 3 appeared on the VB100 test bench for the first time, the stylishness and clever layout continued to impress, allowing the tests to be run through with great simplicity and making the testing experience a joy.

Speeds were as excellent as ever, although probing into archives slowed things down somewhat, and this depth of scanning was not available on access – one of the only options notably absent. Detection could not be faulted in most sets, although a set of samples of an aged DOS polymorphic virus which caused no problems in previous tests were not detected with this version, returning an 'internal error' message in logs. This does not affect *NOD32*'s qualification for the VB100 award, which was achieved easily with full detection of the WildList set and no false positives.

Fortinet Forticlient 3.0.470

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	0

Fortinet's product provided a similarly problem-free run through the tests. The installation, updating and configuration processes are familiar, the core interface having changed little for some time. The product is clearly laid out with all the required elements readily to hand, despite a wide range of other functionality (beside the anti-malware protection) being controlled from the same interface.

Little configuration was required, with the default settings including most file types. Somewhat oddly, ZIP files – perhaps the most common archive format – were scanned less deeply than others. This could be a resource-saving measure introduced due to the very popularity of the format. Despite the thoroughness speeds were quite impressive, and coverage of the sets excellent, with no misses and no false positives earning *Fortinet* a VB100 award.

Frisk F-PROT Antivirus for Windows 6.0.8.1

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.95%	**False positives**	0

F-Prot is a far simpler product than many, with a pared-down interface offering basic control of anti-malware protection and scanning, and little else. With minimal configuration available, and functionality such as logging generally excellently implemented, testing zipped through. Minimal configuration options cut the speed test requirements down, with only the product's seemingly unstoppable urge to remove infected files drawing out the process (an initial run was stopped and replaced with one in which detections were logged only after the first attempt proved to be spending considerable time disinfecting and quarantining).

Default archive settings were among the most sensible so far, with most archive types covered in depth on demand and the basics, self-extractors, ZIPs and the almost identical JAR files delved into a couple of levels deep on access. Speed times were splendid, and detection almost impeccable, earning *Frisk* a VB100 award too.

F-Secure Anti-Virus 7 for Windows Servers 7.00.213

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.97%	**False positives**	0

F-Secure's product is a little more complex and in-depth, though the server version tested here seems little different from the desktop editions seen in previous comparatives. The installation is slick and smooth, lending a solid and trustworthy feel to all components. This weightiness is not too evident in the scanning times, which were surprisingly good over most of the sets although, with the default setting to scan most archive types to a depth of five levels, this set took rather longer. Somewhat oddly next to this thorough setting, file types are identified only by extension, but scanning with 'all files' enabled did not take too much longer to complete, although an occasional moment of sluggishness was observed during operation of the machine thereafter.

F-Secure has presented me with considerable difficulty recently thanks to its rather flaky logging behaviour, which was in evidence once again here, with the 'display log' button bringing up an attractively formatted HTML log in a browser window. As in previous tests, the contents of this log varied wildly, apparently containing a random sampling of items discovered during a scan. Attempting to access the

On-demand throughput	Archive files - default		Archive files - all files		Binaries and system files - default		Binaries and system files - all files		Media & documents - default		Media & documents - all files		Other file types - default		Other file types - default	
	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Through-put (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)
AEC Trustport	4965	0.8	4965	0.8	1677	1.6	1677	1.6	282	5.5	282	5.5	287	2.4	287	2.4
Agnitum Outpost	1185	3.3	1185	3.3	272	10.0	272	10.0	92	16.7	92	16.7	61	11.2	61	11.2
AhnLab V3Net	788	4.9	788	4.9	354	7.7	354	7.7	31	49.7	31	49.7	35	19.6	35	19.6
Alwil avast!	19	204.2	988	3.9	168	16.2	201	13.6	26	59.2	61	25.2	17	40.3	42	16.3
Avira AntiVir	738	5.3	738	5.3	114	23.9	114	23.9	32	48.1	32	48.1	25	27.4	25	27.4
BitDefender Security	1242	3.1	1242	3.1	322	8.5	322	8.5	71	21.7	71	21.7	73	9.4	73	9.4
CA eTrust	520	7.5	520	7.5	71	38.4	71	38.4	29	53.1	29	53.1	22	31.2	22	31.2
Doctor Web Dr.Web	4455	0.9	4455	0.9	756	3.6	756	3.6	102	15.1	102	15.1	108	6.3	108	6.3
ESET NOD32	1093	3.5	1093	3.5	432	6.3	432	6.3	33	46.7	33	46.7	27	25.4	27	25.4
Fortinet Forticlient	506	7.7	506	7.7	483	5.6	483	5.6	44	35.0	44	35.0	35	19.6	35	19.6
Frisk F-PROT	254	15.3	254	15.3	259	10.5	259	10.5	32	48.1	32	48.1	22	31.2	22	31.2
F-Secure Anti-Virus	3144	1.2	3375	1.1	254	10.7	254	10.7	39	39.5	89	17.3	25	27.4	93	7.4
Grisoft AVG	1379.2	2.8	1379.2	2.8	578.1	4.7	578.1	4.7	78.1	19.7	78.1	19.7	91.8	7.5	91.8	7.5
Ikarus Virus Utilities	211	18.4	211	18.4	206	13.2	206	13.2	40	38.5	50	30.8	59	11.6	61	11.2
Kaspersky Anti-Virus	287	13.5	287	13.5	149	18.3	149	18.3	94	16.4	94	16.4	87	7.9	87	7.9
Kingsoft Anti-virus	292	13.3	292	13.3	385	7.1	385	7.1	25	61.6	25	61.6	29	23.6	29	23.6
McAfee VirusScan	58	66.9	843	4.6	328	8.3	339	8.0	60	25.7	57	27.0	63	10.9	60	11.4
Microsoft Forefront	1190	3.3	1190	3.3	306	8.9	306	8.9	68	22.6	68	22.6	45	15.2	45	15.2
MWTI eScan	2314	1.7	2314	1.7	468	5.8	468	5.8	300	5.1	300	5.1	298	2.3	298	2.3
Norman Virus Control	807	4.8	807	4.8	1429	1.9	1429	1.9	53	29.0	53	29.0	153	4.5	153	4.5
PCTools AntiVirus	487	8.0	713	5.4	1115	2.4	1141	2.4	864	1.8	880	1.7	922	0.7	925	0.7
Quick Heal AntiVirus Lite	572	6.8	594	6.5	60	45.5	60	45.5	42	36.7	46	33.5	24	28.6	30	22.9
Redstone Redprotect	1863	2.1	1863	2.1	348	7.8	348	7.8	178	8.6	178	8.6	166	4.1	166	4.1
Sophos Anti-Virus	38	102.1	1446	2.7	210	13.0	260	10.5	29	53.1	48	32.1	15	45.7	58	11.8
Symantec Endpoint Protection	810	4.8	850	4.6	281	9.7	294	9.3	75	20.5	75	20.5	71	9.7	73	9.4
VirusBuster for Windows Servers	510	7.6	999	3.9	223	12.2	234	11.7	26	59.2	49	31.4	16	42.9	44	15.6
Webroot SpySweeper with AntiVirus	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

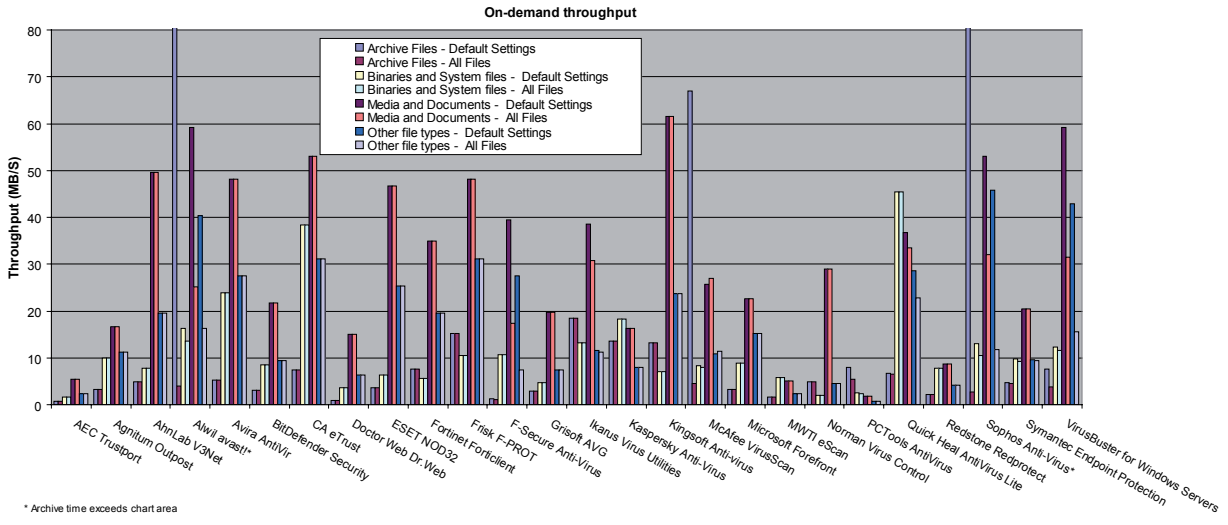
results of scanning the full test collection produced logs varying in size from 50 to 1500 KB. After much frustration trying to achieve the best results with this method, a series of smaller scans set to delete files proved the simplest way of judging the product's effectiveness.

This effectiveness was considerable, with splendid detection rates and no false positives, just a few alerts on suspect tools with potentially unwanted uses. With no problems at all in the WildList *F-Secure* also qualifies for a VB100 award.

Grisoft AVG 7.5.516

ItW	100.00%	**Worms & bots**	99.91%
ItW (o/a)	100.00%	**DOS**	99.10%
File infector	98.43%	**Macro**	100.00%
Polymorphic	78.55%	**False positives**	0

After an initial problem with an activation code inappropriate for use on a server, *AVG* proved somewhat



* Archive time exceeds chart area

simpler to handle, slipping slickly through its install and skipping lightly over the test sets. Although the multiple-window configuration system remains somewhat baffling, the limited configuration options were eventually tracked down and testing produced no major frustrations.

Scanning times were fairly decent, although again by default files with altered extensions are ignored. Detection rates were similarly solid rather than excellent, but the WildList was covered without difficulty, and with no false positives recorded *Grisoft* also makes the VB100 grade.

Ikarus Virus Utilities 1.0.61

ItW	99.55%	**Worms & bots**	99.60%
ItW (o/a)	99.55%	**DOS**	91.37%
File infector	96.28%	**Macro**	96.45%
Polymorphic	82.05%	**False positives**	8

Ikarus has bravely battered at the VB100 door for some time now, and has gradually moved closer to the required standard for qualification, with high levels of false positives having been the major stumbling block in recent tests.

The product's interface uses the .NET framework, and has suffered some flakiness in the past, which this month was considerably lessened. However, on a few occasions the GUI seemed to fail to open, and during the scanning of large infected sets the whole thing seems to flicker and spasm rather worryingly.

An initial run over the clean test set produced some remarkable speed times and an even more eyebrow-raising absence of false alarms. Some quick investigation quickly showed that I had omitted to apply the update, and that in its bare state the product has hardly any detection capabilities at all. Re-running the tests showed that a small number of clean files has been mislabelled, and a handful of WildList items missed, a few odd samples of several of the latest polymorphic additions. Although speed times were impressive and detection in the other sets fairly reasonable, *Ikarus* still has a few more issues to resolve before attaining a VB100 award.

Kaspersky Anti-Virus 6.0 for Windows Servers 6.0.3.837

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.97%	**False positives**	0

Kaspersky, meanwhile, is a seasoned competitor with a long history of excellent performance, a few minor technical issues in recent tests notwithstanding. The product, not quite as glossy and glitzy as the home-user offering provided lately, is no less solid or reliable for it, and offers a well-designed, intuitive interface with an excellent level of configuration, although scanning of archives on access seemed to produce a fairly erratic selection of depths for different formats.

After a few brief and easy tweaks the product stomped through the tests, speeds reflecting a more thorough attitude to scanning than many, but results showing splendid coverage and no false positives, thus earning *Kaspersky* yet another VB100 award.

File access lag time	Archive files - default		Archive files - all files		Binaries and system files - default		Binaries and system files - all files		Media & documents - default		Media & documents - all files		Other file types - default		Other file types - default	
	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)
AEC Trustport	1103	0.3	1103	0.3	476	0.2	476	0.2	117	0.1	117	0.1	145	0.2	145	0.2
Agnitum Outpost	64	0.0	N/A	N/A	293	0.1	293	0.1	95	0.1	95	0.1	75	0.1	75	0.1
AhnLab V3Net	77	0.0	N/A	N/A	355	0.1	355	0.1	37	0.0	37	0.0	40	0.0	40	0.0
Alwil avast!	96	0.0	1045	0.3	258	0.1	261	0.1	144	0.1	149	0.1	69	0.1	51	0.1
Avira AntiVir	35	0.0	284	0.1	118	0.0	141	0.0	31	0.0	42	0.0	19	0.0	47	0.1
BitDefender Security	927	0.2	927	0.2	364	0.1	364	0.1	150	0.1	150	0.1	147	0.2	147	0.2
CA eTrust	22	0.0	N/A	N/A	76	0.0	N/A	N/A	36	0.0	N/A	N/A	29	0.0	N/A	N/A
Doctor Web Dr.Web	7	0.0	3499	0.9	40	0.0	834	0.3	32	0.0	101	0.1	34	0.0	107	0.1
ESET NOD32	11	0.0	N/A	N/A	61	0.0	61	0.0	42	0.0	42	0.0	32	0.0	32	0.0
Fortinet Forticlient	385	0.1	385	0.1	478	0.2	478	0.2	38	0.0	38	0.0	47	0.1	47	0.1
Frisk F-PROT	70	0.0	N/A	N/A	251	0.1	251	0.1	40	0.0	40	0.0	25	0.0	25	0.0
F-Secure Anti-Virus	37	0.0	1819	0.5	222	0.1	370	0.1	44	0.0	249	0.2	28	0.0	114	0.2
Grisoft AVG	53	0.0	N/A	N/A	345	0.1	345	0.1	32	0.0	39	0.0	13	0.0	46	0.1
Ikarus Virus Utilities	214	0.1	214	0.1	218	0.1	218	0.1	52	0.0	52	0.0	71	0.1	71	0.1
Kaspersky Anti-Virus	34	0.0	232	0.1	203	0.1	289	0.1	76	0.0	133	0.1	49	0.1	119	0.2
Kingsoft Anti-virus	27	0.0	N/A	N/A	359	0.1	359	0.1	25	0.0	25	0.0	31	0.0	31	0.0
McAfee VirusScan	52	0.0	503	0.1	325	0.1	338	0.1	52	0.0	54	0.0	62	0.1	65	0.1
Microsoft Forefront	109	0.0	N/A	N/A	299	0.1	299	0.1	68	0.0	68	0.0	49	0.1	49	0.1
MWTI eScan	1162	0.3	1162	0.3	251	0.1	251	0.1	102	0.1	102	0.1	96	0.1	96	0.1
Norman Virus Control	22	0.0	N/A	N/A	226	0.1	226	0.1	58	0.0	58	0.0	77	0.1	77	0.1
PCTools AntiVirus	368	0.1	N/A	N/A	1036	0.4	N/A	N/A	144	0.1	N/A	N/A	110	0.1	N/A	N/A
Quick Heal AntiVirus Lite	12	0.0	N/A	N/A	58	0.0	N/A	N/A	36	0.0	N/A	N/A	15	0.0	N/A	N/A
Redstone Redprotect	3208	0.8	3208	0.8	299	0.1	299	0.1	127	0.1	127	0.1	121	0.2	121	0.2
Sophos Anti-Virus	40	0.0	1405	0.4	243	0.1	271	0.1	39	0.0	55	0.0	29	0.0	66	0.1
Symantec Endpoint Protection	29	0.0	N/A	N/A	200	0.1	200	0.1	41	0.0	41	0.0	38	0.0	38	0.0
VirusBuster for Windows Servers	34	0.0	N/A	N/A	225	0.1	226	0.1	31	0.0	50	0.0	18	0.0	43	0.0
Webroot SpySweeper with AntiVirus	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Kingsoft Anti-virus 2008.1.7.10

ItW	99.26%	**Worms & bots**	16.85%
ItW (o/a)	99.26%	**DOS**	12.26%
File infector	71.83%	**Macro**	91.56%
Polymorphic	38.49%	**False positives**	0

Kingsoft is another firm which has had some trouble in recent comparative reviews but has nevertheless continued to strive for the excellence required for a VB100 award. The company's product has grown in stability and responsiveness in the year or so since it first visited the *VB* test bench, and seems very pleasant to look at and rational to use.

Available configuration is less than complete but adequate for my needs, and testing trotted nicely along with impressive scanning times. False positives were pleasingly absent and detection rates showed further improvement,

but alongside a fair number of recent items in the older sets (including some quite significant W32/Sdbot and W32/Mytob variants), several worms in the WildList set were missed, as well as a few samples infected with W32/Virut and W32/Bacalid. As a result, a VB100 award still proves to be a little way out of reach for *Kingsoft* this month.

McAfee VirusScan Enterprise 8.5.0i 5200.2160

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	0

McAfee's enterprise product is a regular on the *VB* test bench and it took me little time to find my way around it. The layout is somewhat individual, but simple to operate and provides the full range of settings and controls expected in a complex corporate environment.

Adjusting the defaults to cover a wider range of file formats did not add too significantly to the pretty fair scanning times, although of course delving deeply into a broad range of archives was a little slower than leaving them unchecked.

The solidity of design and implementation was reflected in some effortlessly impressive detection rates, with nothing missed or mislabelled anywhere, and *McAfee* thus wins a VB100 award.

Microsoft Forefront Client Security 1.5.1941.0

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	99.90%	**Macro**	100.00%
Polymorphic	96.46%	**False positives**	0

Microsoft's product seems to take quite the reverse approach, assuming a mother-knows-best attitude and offering almost nothing by way of configuration options.

Rather amusingly, the installation process required an update to the Windows Update Agent before it could complete, and once installed the simple interface offered some basic information and a page of rather random controls.

The client in use here is part of a more complex suite of products, so it is possible that much of the configuration can be controlled from above. Nevertheless, it would seem appropriate to provide the user with a little more information on how their system is being monitored.

After running some scans and on-access tests a small amount of information emerged about how the product was operating, though little of this came from the product itself. After scanning several thousand infected files the GUI displayed the message 'Items Detected – Severe/High Alert level: 24', while all detections were logged only to the system event log once the on-screen display was closed. A 'History' button reopened the display from each scan, but regularly froze while trying to access the results of large scans and on occasion caused the whole interface to disappear from view.

Despite these annoyances, results were eventually dragged together and showed fairly good speeds. A sensible default selection of files handled all the archive sets without problem on demand and looked briefly into the most common types on access. Detection rates were very good indeed, and without any false positives *Forefront* is awarded a VB100.

MWTI eScan Corporate for Windows 9.0.764.1

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.97%	**False positives**	0

The corporate edition of *eScan* is a little more sober than the normal home desktop version, although its installation process with automatic scanning of important system areas remains much the same.

Configuration is provided via a console resembling the MMC, but dubbed 'EMC', and seems fairly comprehensive. However, little adjustment was needed as the default settings scanned pretty much everything thrown at it.

This resulted in some rather slow scanning speeds but of course excellent detection rates. A couple of items spotted as suspected malware by the *Kaspersky* engine in its other guises were missed here on access, and a few others that were not identified elsewhere were flagged here as potentially risky. However, with no samples missed in the WildList test set, and no false positives, *eScan* also qualifies for a VB100 award.

Norman Virus Control v.5.90.10

ItW	99.95%	**Worms & bots**	100.00%
ItW (o/a)	99.90%	**DOS**	99.12%
File infector	99.15%	**Macro**	100.00%
Polymorphic	84.20%	**False positives**	1

Norman's product is another which makes use of a variety of windows for various facets of its control and operation, and as usual this led to a certain amount of confusion and frustration. However, once their interoperation had been mastered things proceeded reasonably well, with the only issue found in the actual running of the product being a problem with the redirection of logs. An option to change the folder in which logs are saved seemed ideal for my use, but on checking the selected location at the end of the test it was found to be entirely log-free. Fortunately all the required data was stored within *Norman*'s own logging folder and results were thus gathered after only a brief moment of worry.

There was not a great deal of flexibility in the types of files scanned, with a handful of the more common archives investigated on demand but none on access. All file extensions were analysed for malicious content by default however, and this resulted in some rather below average speed times, as well as a single file in one of the clean sets being labelled as malware.

Detection rates were also less than perfect, with a handful of polymorphic variants in the WildList set not fully covered, the on-demand scanner faring slightly better than the on-access. *Norman* thus misses out on a VB100 award this month.

PCTools AntiVirus 3.6 for Windows 3.6.1.8

ItW	99.80%	**Worms & bots**	99.91%
ItW (o/a)	99.80%	**DOS**	99.77%
File infector	99.21%	**Macro**	100.00%
Polymorphic	85.91%	**False positives**	0

PCTools products have been a little awkward in the past, with an inflexibility of configuration providing some frustration. This time, however, everything I needed seemed to be both available and easily accessible. The installation offers an accompanying install of the *Google* toolbar, which I turned down for my tests, but few other difficult decisions were required.

Despite the default settings covering no archive types or renamed files on access, scanning speeds were on the slow side, and the system seemed less than usually responsive.

On-demand scans had slightly more thorough settings, with most archives probed to a single level, and the resulting speeds were even less impressive.

Scanning infected sets brought up a beautiful cascade of alert popups, scrolling and interweaving with each other down one side of the screen. Detection rates closely mirrored those of *Agnitum*, as both products use the *VirusBuster* engine, and thus it was hardly a surprise to see the same handful of misses in the WildList. Thus, despite a lack of false positives, *PCTools* does not receive a VB100 award for its efforts.

Quick Heal Quick Heal AntiVirus Lite 9.50

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	95.23%
File infector	97.64%	**Macro**	98.23%
Polymorphic	81.86%	**False positives**	5

Quick Heal is one of the few products to scan the system prior to installation, but the setup process is nevertheless speedy and efficient, offering a friendly 'Welcome' message flashing in the system tray. The interface is visually appealing and seems very stable and solid, but again configuration is kept to a minimum.

On-access settings can barely be adjusted at all, with no way of forcing files such as my renamed EICAR file to be watched for, and archives left unprobed. On-demand scanning is a little more thorough, with a few items delved into lightly by default and slightly more depth available for those who want it.

This lightness of scanning may contribute somewhat to the speed of the product, which was uniformly excellent. Detection rates were a little below average over the older sets but the WildList was covered without difficulty. In the clean set, a few items were incorrectly flagged as malicious, mostly identified as 'I-Worm.Sohanad.T', suggesting some overzealousness in the detection of this item. This inaccuracy is enough to deny *Quick Heal* a VB100 award this time.

Redstone Redprotect Anti-Virus Plus 0.4.2.1

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.97%	**False positives**	0

Redstone returns for a second attempt at the VB100, having been denied last time by a small technicality in the settings of the *Kaspersky* engine on which it is based. This

Archive scanning		ACE	CAB	EXEZIP	JAR	LZH	RAR	TGZ	ZIP	EXT*
AEC Trustport Antivirus	OD	X	√	√	√	√	√	√	√	√
	OA	X	√	√	√	√	√	√	√	√
Agnitum Outpost	OD	X	√	√	X	√	√	√	√	√
	OA	X	X	X	X	X	X	X	X	√
AhnLab V3Net	OD	√	√	X	√	√	√	X	√	√
	OA	X	X	X	X	X	X	X	X	√
Alwil avast!	OD	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√
	OA	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
Avira AntiVir	OD	√	√	√	√	√	√	√	√	√
	OA	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
BitDefender Security	OD	√	√	√	√	√	8	√	8	√
	OA	√	√	√	√	√	8	√	8	√
CA eTrust	OD	1	√	1	√	√	√	√	√	√
	OA	X	X	X	1	X	X	X	X	√
Doctor Web Dr.Web	OD	X	√	√	√	√	√	√	√	√
	OA	X	X/√	X/9	X/√	X/√	X/√	X/5	X/√	√
ESET NOD32	OD	X	√	√	√	√	√	X	√	√
	OA	X	X	X	X	X	X	X	X	√
Fortinet Forticlient	OD	X	√	√	√	√	√	√	4	√
	OA	X	√	√	√	√	√	√	4	√
Frisk F-PROT	OD	X	√	√	√	√	√	√	√	√
	OA	X	X	2	2	X	X	X	2	√
F-Secure Anti-Virus	OD	X/√	5	5	5	5	5	2	5	X/√
	OA	X/√	X/5	X/5	X/5	X/5	X/5	X/2	X/5	X/√
Grisoft AVG	OD	X	√	√	1	X	√	X	√	X
	OA	X	X	X	X	X	X	X	X	X/√
Ikarus Virus Utilities	OD	2	3	1	3	3	3	X	3	√
	OA	2	3	1	3	3	3	X	3	√
Kaspersky Anti-Virus	OD	√	√	√	√	√	√	√	√	√
	OA	X/4	X/4	X/1	X/4	X/5	X/5	X/1	X/2	√
Kingsoft Anti-virus	OD	X	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	√	√	√	√	√	√	√	√	√
	OA	X	X	1	1	X	X	X	1	√
MWTI eScan	OD	√	√	√	√	√	√	√	√	√
	OA	√	√	√	√	√	√	√	√	√
Norman Virus Control	OD	X	X	X	√	√	X	√	√	√
	OA	X	X	X	X	X	X	X	X	√
PCTools AntiVirus	OD	1/2	1/√	1/√	1/√	X	1/√	X/√	1/√	√
	OA	X	X	X	X	X	X	X	X	X
Quick Heal AntiVirus Lite	OD	X	2/5	X	2/5	X	2/5	1	2/5	X/√
	OA	X	X	X	X	X	X	X	X	X
Redstone Redprotect	OD	√	√	√	√	√	√	√	√	√
	OA	√	√	√	√	√	√	√	√	√
Sophos Anti-Virus	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 Protection	OD	X	3/√	3/√	3/√	3/√	3/√	X/5	3/√	√
	OA	X	X	X	X	X	X	X	X	√
VirusBuster for Windows Servers	OD	2	√	√	X/√	X	√	√	√	X/√
	OA	X	X	X	X	X	X	X	X	X/√
Webroot SpySweeper with AntiVirus	OD	X	X	5	6	X	X	6	X	√
	OA	X	X	X	X	X	X	X	X	√

is another .NET product, again at a fairly early stage in its development, and some flakiness is evident in the running of the interface, with occasional unexpected shutdowns and the odd error message, particularly when trying to access

logs. Configuration is extremely minimal here, with the controls accessible from the system tray icon limited to running a scan and shutting down the on-access scanner. With the 'default' settings provided in the form of a series of registry keys it is here that adjustments must be made if needed – changing the default on-access behaviour (which seems to be to prompt users with a message offering not to delete if they respond within 30 seconds) seems not always to respond as expected, interrupting a few scans with its warnings.

After some struggles extracting scan data from a series of XML files and allowing the on-access scanner to delete most of the infected test set, results were obtained. The results proved as excellent as those achieved by other products using the *Kaspersky* engine.

With detection almost impeccable and false alarms completely absent, *Redstone* qualifies for its first VB100 award.

Sophos Anti-Virus 7.0.6

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	99.80%
Polymorphic	100.00%	**False positives**	0

The entire *Sophos* product line has a resolutely corporate focus, and thus the offering for this test seems identical to those that have appeared in previous comparatives. With the usability never too taxing, the installation and configuration of the product slid by without any trouble.

Testing proved just as simple a process, although the progress bar proved as errant as ever (which proved to be a common issue in this test in cases where an attempt was made to estimate the remaining scanning time), and the logging seemed rather strangely organised and confusing.

The deep configuration available did not extend to scanning archives beyond five levels deep, but most types were covered, and scanning speeds – excellent with the default settings – were fairly good.

Detection rates were splendid, and although the switching on of a wider range of suspicious detection flagged up a number of unusually packed files in the clean set, alongside a handful of 'adware/PUA' and 'Hacktool' alerts, no full false positives arose and *Sophos* is able to claim another VB100 award after a couple of unlucky months.

Symantec Endpoint Protection 11.0.780.1109

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	0

Symantec's corporate desktop product has undergone a considerable change recently, and still seems to be suffering a few teething problems.

Although the installation was impressively speedy, the automatic attempt at online updating took some time and effort to put a stop to (including a warning that it may take a few minutes to 'clean up'), followed by a reboot.

Logging of scan results also proved problematic, with attempts to open the logs via the interface causing some nasty freezes for the on-access data, and simply a blank page for on-demand data, despite several scans and several tens of thousands of items detected. The freezes were resolved by killing the process with the Task Manager, which brought up an increasing number of alert messages from *Symantec*'s anti-tamper system, informing me that attempts to shut it down had been 'blocked' – in one instance, after several dozen of these messages protection was in fact stopped and the interface restarted.

These minor issues, likely due to the generation of a log exceeding 150MB, did little to affect the results themselves however. Scan times were fairly good, with on-demand defaults delving three levels deep into most archives and more available. The on-access scanner seemed to offer only limited configuration but did identify disguised file types. Parsing the enormous log showed superb detection rates and a complete absence of false positives, and *Symantec* also qualifies for a VB100 award.

VirusBuster VirusBuster for Windows Servers 5.3 b.57

ItW	99.82%	**Worms & bots**	99.91%
ItW (o/a)	99.82%	**DOS**	99.77%
File infector	99.21%	**Macro**	100.00%
Polymorphic	85.91%	**False positives**	0

VirusBuster's server product again seems much the same as the home-user version, with the addition of an MMC-based console for some extra configuration. This included options which seemed to imply archives would be scanned

internally on access, but apparently only cover normal executables renamed as archives to conceal their intentions (which would be ignored in the default modes).

The interface itself is pleasant if a little fiddly when setting up scans, and suffers a tendency to linger a little over saving its logs, even those with minimal content. This did little to dent a good performance in terms of both speed and detection, with no false alarms and the W32/Virut samples missed by the other products using the same engine causing no difficulties here – presumably due to a slightly later version of the detection data. However, one remaining item, a W32/VB worm variant, was missed, and although we are advised that detection was added to the product a week or so after the submission deadline, the missed detection prevents *VirusBuster* from attaining a VB100 award this month.

Webroot SpySweeper AntiSpyware with AntiVirus Corporate Edition 3.50.3578

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	100.00%	**Macro**	99.93%
Polymorphic	100.00%	**False positives**	0

Last on the list of products comes *Webroot*'s *SpySweeper*. This is *SpySweeper*'s second visit to the *VB* test bench, having made its debut – and gained VB100 status – in the June 2007 *XP* review (see *VB*, June 2007, p.10). The corporate version submitted here was considerably different from the home-user edition submitted previously.

After a rather drawn out installation and startup process, the product offers a fairly comprehensive interface with some apparently well-populated configuration pages. Unfortunately, these are initially greyed out, as the client system submitted is designed to cede all control to a management server. Some changes to the registry allowed access to the settings (after providing a password) and testing continued.

Problems did not end there however, as the on-demand scanner seemed to provide no option to scan only a given folder and the entire system had to be scanned – no small job in this case. On returning after leaving the scan running overnight I found that the test sets had been covered pretty thoroughly, and they were then replaced before attempting the on-access tests. These were again hampered by the product's rather unusual implementation, with on-read scanning deactivated by default and only functioning rather flakily once enabled. This rendered any speed results gathered somewhat suspect, and only detection results were obtained by copying all test sets to the system across the network.

As far as can be judged by feeling alone, the protection did seem to slow the machine's response time down noticeably, especially during the five or so minutes after a reboot when the system tray icon is whirring and the interface unavailable (presumably doing some sort of boot-up checks.) After several attempts yielded a usable log of detection, results turned out to be pretty good – close to the high level expected of the *Sophos* engine used in the product – bar a few file types not scanned with these settings. Without false positives either, *Webroot* earns another VB100 award this month.

CONCLUSIONS

After the deluge of problems detecting a handful of nasty polymorphic viruses in the last round of testing, it was good to see far better coverage of the WildList this time. Most products seemed to have resolved their issues with these items, with a small handful of the latest worms causing the majority of difficulties this month.

False positives hit a cluster of other products, but few suffered any major issues with false alerting, most only flagging single files. With only a small number of packages added to the clean test set this month, this was to be expected. Many of the problems were with files that have been in the set for some time without causing any problems, which suggests that adjustments to heuristics are the main cause of the niggles.

The addition of the archive scanning test, intended as an adjunct to the speed test to indicate how speed times are affected by the depth of scanning, has also provided some information on the breadth of configuration available in products. Running a server-based test, we expected to draw in mostly enterprise-level products, which one would expect to offer considerably more flexibility than home-user offerings. Enterprise admins have far more complex and varying requirements than the simpler needs of the home user, with marked differences in network layout and system uses from company to company, widely varying company policies to comply with and so on. By limiting the choices offered to their users and admins, some products may risk limiting their usefulness in the corporate arena.

Technical details

Tests were run on identical machines with *AMD Athlon64 3800+* dual core processors, 1GB RAM, 40GB and 200 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive, all running *Microsoft Windows Server 2003 Enterprise Edition R2 SP2*.

END NOTES & NEWS

Black Hat DC 2008 Briefings and Training take place 18–21 February 2008 in Washington, DC, USA. For full details and registration see http://www.blackhat.com/.

The SecureLondon Conference on emerging threats will be held 4 March 2008 in London, UK. Attendees will be given an overview of the interaction between web, spam and malware, with a focus on specific campaigns. For further information see https://www.isc2.org/cgi-bin/events/information.cgi?event=48.

Black Hat Europe 2008 takes place 25–28 March 2008 in Amsterdam, the Netherlands. Registration is now open. See http://www.blackhat.com/.

Forrester's Security Forum will be held 2–3 April 2008 in Amsterdam, the Netherlands. *Forrester* is offering *Virus Bulletin* readers a 15% discount on the registration fee, which can be claimed by downloading the brochure from http://www.forrester.com/imagesV2/uplmisc/Forrester_Virus_Bulletin_Security_Brochure.pdf or calling +31 (0)20 305 4848 and quoting the code 'Virus Bulletin reader'.

RSA Conference 2008 takes place 7–11 April 2008 in San Francisco, CA, USA. This year's theme is the influence of Alan Mathison Turing, the British cryptographer, mathematician, logician, philosopher and biologist, often referred to as the father of modern computer science. Online registration is now available. See http://www.rsaconference.com/2008/US/.

Infosecurity Europe takes place 22–24 April 2008 in London, UK. For more information and to register interest in attending see http://www.infosec.co.uk/virusbulletinevents.

The 2nd International CARO Workshop will be held 1–2 May 2008 in Hoofddorp, the Netherlands. For details see http://www.datasecurity-event.com/.

EICAR 2008 will be held 3–6 May 2008 in Laval, France. See http://www.eicar.org/conference/ for the full details.

The 5th Information Security Expo takes place 14–16 May 2008 in Tokyo, Japan. For more details see http://www.ist-expo.jp/en/.

The 9th National Information Security Conference (NISC) will be held 21–23 May 2008 in St Andrews, Scotland. For full details and registration information see http://www.nisc.org.uk/.

Hacker Halted USA 2008 takes place 1–4 June 2008 in Myrtle Beach, SC, USA. The conference aims to raise international awareness towards increased education and ethics in information security. Hacker Halted USA delegates qualify for free admission to the Techno Security Conference which runs concurrently. For more details see http://www.hackerhalted.com/.

The 20th annual FIRST conference will be held 22–27 June 2008 in Vancouver, Canada. The five-day event comprises two days of tutorials and three days of technical sessions where a range of topics of relevance to teams in the global response community will be discussed. For more details see http://www.first.org/conference/.

The 17th USENIX Security Symposium will take place 28 July to 1 August 2008 in San Jose, CA, USA. A two-day training program will be followed by a 2.5-day technical program, which will include refereed papers, invited talks, posters, work-in-progress reports, panel discussions, and birds-of-a-feather sessions. For details see http://www.usenix.org/events/sec08/cfp/.

Black Hat USA 2008 takes place 2–7 August 2008 in Las Vegas, NV, USA. Online registration is now open and a call for papers has been issued. For details see http://www.blackhat.com/.

VB2008 will take place 1–3 October 2008 in Ottawa, Canada. *Virus Bulletin* is currently seeking submissions from those wishing to present papers at VB2008. Full details of the call for papers are available at http://www.virusbtn.com/conference/vb2008.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

- Single user: $175
- Corporate (turnover < $10 million): $500
- Corporate (turnover < $100 million): $1,000
- Corporate (turnover > $100 million): $2,000
- *Bona fide* charities and educational institutions: $175
- Public libraries and government organizations: $500

Corporate rates include a licence for intranet publication.

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

Editorial enquiries, subscription enquiries, orders and payments:

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

Tel: +44 (0)1235 555139 Fax: +44 (0)1235 531889

Email: editorial@virusbtn.com Web: http://www.virusbtn.com/

vb Spam supplement

CONTENTS

NEWS & EVENTS

NO TASTE FOR SPAM?

The practice of domain tasting, often used by spammers and other shady types to register tens of thousands of Internet domain names at no cost, looks set to end thanks to a new ICANN ruling.

ICANN charges a fee of 20 cents per domain name per year, but under its current rules a domain owner is able to 'return' the domain name within five days for a full refund (allowing legitimate registrants a grace period to rectify any mistakes that they may have made in their registration). This means that spammers and scammers have been able to register tens of thousands of domains for no cost – allowing spammers to hide their identity, and assisting search-engine spammers in their quest to hijack search engine rankings.

At the end of last month, however, the ICANN board voted to make the 20-cent fee non-refundable. While 20 cents may seem like small pennies, when multiplied by the tens of thousands of domains being registered by spammers on a regular basis it is likely to prove sufficiently costly to make the practice unprofitable.

Shortly before the ICANN ruling, *Google* also took a step towards curtailing the practice of domain tasting. The search engine and advertising giant announced that it would start looking out for domains that are repeatedly registered and dropped and exclude them from its *AdSense* program – thus preventing scammers from generating advertising revenue from them.

It is not clear when the change to the ICANN 20-cent fee will take effect, but industry watchers believe it could be within the next month.

SPAMMERS AND SCAMMERS IN COURT

The US Federal Trade Commission (FTC) has settled a court case with a spamming advertising company. According to the FTC, *Member Source Media* used deceptive emails and online advertising to lure customers to its websites. The settlement requires *Member Source Media* to disclose all costs and obligations associated with the products and services it advertises, bans it from sending emails that violate the CAN-SPAM Act and requires it to pay $200,000 in civil penalties.

Meanwhile, three African defendants have pled guilty in a US court to defrauding a total of $1.2 million from US citizens through a series of 419 scams. The two Nigerian defendants and one Senegalese man were charged with a combination of conspiracy, wire fraud and email fraud. A fourth defendant fled to Nigeria but is being held by Nigerian authorities pending extradition to the US. The scam was originally uncovered by Dutch authorities and the men were arrested in Amsterdam in 2006. The men face a maximum penalty for mail and wire fraud of 20 years in prison, while the conspiracy charge carries a maximum penalty of five years in prison.

MORTGAGE SPAM ROCKETS

Mortgage spam saw a significant increase last month in conjunction with the interest rate cuts announced by the US Federal Reserve. According to researchers at *Commtouch*, mortgage spam rose to 10% of all spam as spammers took note of the fact that millions of US mortgages became eligible for refinancing as a result of the lowered interest rates. According to *Commtouch,* finance-related spam accounted for a mere 2% of all spam subjects in the fourth quarter of 2007.

EVENTS

The MAAWG 12th general meeting, open to members and non-members, will be held 18–20 February 2008 in San Francisco, CA, USA. See http://www.maawg.org/.

The 2008 Spam Conference will take place 27–28 March 2008 in Cambridge, MA, USA. Potential speakers are invited to submit proposals for papers, tutorials or workshops. For the full details see http://spamconference.org/.

CEAS 2008 will take place 21–22 August 2008 in Mountain View, CA, USA. A call for papers for the event is now open. For more information see http://www.ceas.cc/2008/.

FEATURE

PREDICTIONS ABOUT THE PREDICTION SCAM

Sampson Pun, Eric Parsons, Margaret Nielsen, David Ma and John Aycock
University of Calgary, Canada

Many traditional confidence games have made their way into electronic form. Witness the humble advance fee fraud, for example, dating back to before Jack the Ripper's time [1, 2] and now flourishing in large volumes thanks to the magic of spam. One scam that is conspicuous by its electronic absence, however, is the prediction scam.

The prediction scam works like this. A scammer picks an event with a typically binary result, such as a sports event: win or lose. Starting from a pool of (say) 32 people, the scammer contacts half the people and predicts one result, predicting the opposite result to the other half. The event occurs, and the scammer must have given the correct prediction to 16 people. Those 16 are now split into two groups, and the scammer repeats the process, and then repeats the process again. Now four people are really, *really* convinced of the scammer's predictive powers.

The scammer makes money by asking people in the final group to pay for the next prediction. Remember, this group has only seen correct predictions from the scammer, so the likelihood of them being willing to pay is fairly high. The victims expect to recoup their investment by betting on the event themselves. Of course the paid-for prediction, if it arrives at all, is no better than a random guess.

It is easy to imagine this scam electronically: the scammer turns spammer, and emails the predictions to the masses. The problem with a naïve conversion of the prediction scam into electronic form is time. The scammer must remain able to contact and hold the interest of their potential victims for long enough to make the predictions, and for the corresponding real-world events to occur. This can be mitigated somewhat by choosing periods when lots of events are happening, such as sports playoffs. However, the time factor still means there is a real risk to the scammer that their emails will become blocked as spam.

The answer comes in the form of *parlaying*. In gambling, a parlayed bet is made on the outcome of multiple games; the bet is only won if all the games turn out as predicted. Now, instead of having to get N prediction emails through to a victim, the scammer sends one email containing N predictions. To people who get the correct predictions, the scammer has instant credibility after just one spam run.

Now, the scammer has two choices for the victim to make contact. They can continue to use email because the situation has changed, and not in favour of anti-spam. The relatively small number of victims' emails can now be handled manually by the scammer, so there are no giveaway bulk mail indicators. Furthermore, the victim now *wants* the scammer's emails, and it is not far fetched to imagine the scammer asking a victim to adjust their anti-spam filter accordingly.

The scammer could also send the victim to a website. This means that anti-spam defences only get one chance to detect this fraud, at the outset. A clever scammer would also make the website require login, and issue unique logins for each prediction; only the people who receive correct predictions are allowed into the site.

At the scammer's website, the victim would buy the next prediction, or buy some software that they can use to make their own predictions. The latter, of course, is an opportunity for the scammer not just to make money, but to infect the victim's machine. Browser-based anti-phishing defences could block access to the scammer's website if the scam is caught in time, but again the victim wants the scammer's communication – anti-phishing defences may quickly find themselves disabled.

The prediction scam is not limited to sports. Stock predictions work equally well [3, 4], and may even lead to a new variant of the pump-and-dump scam. After all, who wouldn't listen to a STRONG BUY stock alert from someone with a proven ability to predict the stock market?

The prediction scam is unfortunately likely to be successful when it makes the transition into electronic form. To start, users will simply not be wise to the scam. Also, unlike advance fee fraud, there is no need for the scammer to build their credibility or convince the victim, because the 'proof' is already supplied and is independently verifiable. Throw away the crystal ball and Tarot cards: the future of prediction is on its way.

REFERENCES AND NOTES

John Aycock's research is supported in part by the National Sciences and Engineering Research Council of Canada. He thanks Kelly Wilson for some fact-checking.

[1] Power-Berrey, R. J. The bye-ways of crime: with some stories from the Black Museum. Greening, 1899.

[2] Letter from Foreign Office (signature illegible) to the Under Secretary of State of the Home Office, 22 February 1881. In UK National Archives, HO 45/9606/A2527.

[3] Paulos, J. A. Innumeracy: Mathematical illiteracy and its consequences. Hill and Wang, 2001.

[4] Henderson, L. Crimes of persuasion: Schemes, scams, frauds. Coyote Ridge, 2003.