SEPTEMBER 2009

virus BULLETIN

Fighting malware and spam

## CONTENTS

## IN THIS ISSUE

### MISSING IN ACTION?

How can the AV industry collectively miss a virus for as long as 12 months, asks Roel Schouwenberg on the subject of the Delphi-targeting Win32/Induc.A.

**page 2**

### FLIPPING THE COIN

Peter Ferrie reports on a virus that uses an interesting variation on cavity infection.

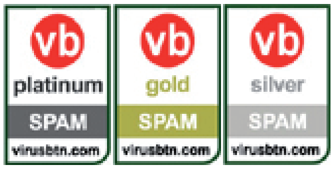
**page 4**

### G DATA TOTAL SECURITY

VB's test team decided to test drive G Data's latest complete suite product and were thoroughly impressed by the breadth and quality of the product.

**page 10**

### VBSPAM AWARDS



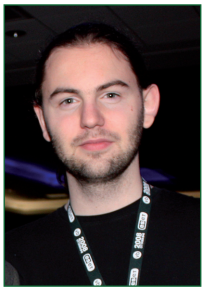
The latest round of anti-spam comparative testing sees an increase in the size of the field of competitors for the third time in a row. This month a total of 12 products line up on the test bench. Martijn Grooten details the performance levels and the award winners.

**page 15**

ISSN 1749-7027

virus BULLETIN

COMMENT



*'If there was ever an argument to be made against a whitelisting-only approach to security, this is it.'*

**Roel Schouwenberg, Kaspersky Lab**

## HOW CLEAN IS CLEAN?

During the middle of August the world suddenly became aware of a new file infector that had been found in the wild: Win32/Induc.A is a virus that targets certain versions of the Borland Delphi compiler. It manages to compile an infected version of SysConst.dcu, an essential Delphi system library.

This means that any Delphi program that depends on this library for compilation will contain the self-replicating code. What it comes down to is that virtually every program compiled will carry the virus.

The idea behind compiler malware is not new. Almost exactly 25 years ago Ken Thompson described a similar and slightly more complex situation than we see with Win32/Induc.A.

Putting this idea into practice also isn't very new. At the end of 1997, the Russian malware writer 'Z0mbie' released two innovative file infectors. One virus targeted TPU files – libraries used by Borland's famous Turbo Pascal compiler. The other targeted BGI files, which stands for Borland Graphic Interface, a primitive form of video driver.

However, what is more interesting is the fact that Win32/Induc.A has been in the wild for quite a while. Initially it was rumoured to have been in the wild since 2005. Such claims have not been substantiated, but confirmed cases date back to almost 12 months ago. In reality this most likely means that it has been in the wild for more than a year.

**Editor:** Helen Martin
**Technical Editor:** Morton Swimmer
**Test Team Director:** John Hawes
**Anti-Spam Test Director:** Martijn Grooten
**Security Test Engineer:** Simon Bates
**Sales Executive:** Allison Sketchley
**Consulting Editors:**
Nick FitzGerald, *Independent consultant, NZ*
Ian Whalley, *IBM Research, USA*
Richard Ford, *Florida Institute of Technology, USA*

How can the AV industry collectively miss a virus for such a length of time? Unless the file is put through an obfuscator or protector after compilation the viral code is highly visible.

The reason most likely has to do with the huge amounts of incoming (malicious) files. It's rather unlikely that automated processing of these infected files will reveal very much, as the virus needs Borland Delphi installed in order to start its malicious payload.

Even five years ago it would have been unlikely for Win32/Induc.A to have gone unnoticed for such a long period of time. It seems clear that we've reached an era where rare dependencies, such as having a compiler on the system, or logic bombs can thrive.

According to some reports Win32/Induc.A is only a proof of concept as it does nothing more than replicate. Clearly, I must have missed the announcement declaring file infection a non-malicious deed. Or perhaps this was it. In this regard it looks like we're becoming too focused on behaviour such as password stealing. In fact, if Win32/Induc.A hadn't been causing problems in some cases it would have taken even longer for us to become aware of its existence.

We have seen many thousands of supposedly clean files that turned out to be infected. I'm certain that the clean collection of every AV vendor out there has contained at least some infected files at some point in time. If there was ever an argument to be made against a whitelisting-only approach to security, this is it. Just as there is no 100% detection of malicious code, there is also no 100% guarantee that supposed clean files really are clean.

Malware authors may have done their own analysis of the success of Win32/Induc.A, and I'm inclined to think that from now on we should trust our clean files even less. Not surprisingly, next to all those thousands of apparently clean files there are also many malicious files infected with this virus.

What is more surprising is that Win32/Induc.A-infected trojans were being seeded even several days after the majority of AV vendors had released detection for it. This tells us that these malware authors are not running any AV solution, which is not very surprising. More surprising is that they are apparently releasing new variants without checking them against (up-to-date) AV solutions.

Most likely this means that malware authors have grown so confident in the fact that making minute changes to the source will be enough to evade detection that they are not even bothering to scan the newly created malware any more. Perhaps that is the most worrisome conclusion we can draw from the Win32/Induc.A situation.

# NEWS

## PLENTY MORE PHISH IN THE SEA?

Phishers appear to be changing tactics, according to a number of reports from security firms last month.

*Kaspersky Lab* reported a drop in the number of phishing emails seen, from 0.78% of email traffic in the first quarter of 2009 to 0.49% in the second half of the year. Meanwhile, *Symantec*'s monthly *MessageLabs* report highlighted a decrease of 0.01% in the number of phishing emails as a proportion of all email traffic, while the percentage of phishing mails as a proportion of the total number of email-borne threats decreased by 6% from July to August (dropping to 86.9%). *IBM* also reported a decrease, saying that the number of phishing emails as a proportion of total spam messages fell in the first six months of this year to 0.1%, with the figure at the same time last year having been between 0.2% and 0.8%.

With losses from online banking fraud still high, the suspicion is that cybercriminals are favouring the use of banking trojans to obtain victims' banking credentials rather than using scam emails and fake websites – indeed, the German Bundeskriminalamt (Federal Criminal Police) has reported that only 10% of online banking fraud can now be traced back to fake banking sites.

*IBM* also found a shift in the type of businesses targeted by phishers: last year 90% of phishing targets were banks, but this year that figure has fallen to 66%, while online payment services such as *PayPal* have seen an increase in phishing attacks.

## CELEBRITY WATCH

Security firm *McAfee* has revealed a start-studded list of the most dangerous celebrities in cyberspace in 2009. This year's most dangerous celebrity online was revealed as Jessica Alba, with users searching on her name having a one-in-five chance of coming across a malicious or compromised website.

Cybercriminals look to exploit sites that receive a large volume of traffic, and this makes celebrity websites a prime target. Cybercriminals pay close attention to the popularity of different sites and Internet search terms, thus the more a celebrity is in the public eye (or the more popular the celebrity), the greater the chance of them being used to hook in vulnerable users.

Reflecting the fickle nature of the general public, this year's top ten list included just three of the same celebrities as last year's list: Beyonce retained her number two position, while Brad Pitt slipped from the top spot in 2008 to the number ten position this year, while his partner Angelina Jolie jumped from the number ten position last year to the number eight position this year.

### Prevalence Table – July 2009

| Malware | Type | % |
|---|---|---|
| Agent | Trojan | 28.27% |
| OnlineGames | Trojan | 20.50% |
| Kryptik | Trojan | 15.36% |
| Heuristic/generic | Misc | 5.74% |
| NetSky | Worm | 4.70% |
| Mytob | Worm | 3.86% |
| Virut | Virus | 3.66% |
| Zbot | Trojan | 2.56% |
| Mydoom | Worm | 2.33% |
| Encrypted/Obfuscated | Misc | 2.25% |
| Bredolab | Trojan | 1.54% |
| Iframe | Exploit | 1.47% |
| Clicker-misc | Trojan | 0.89% |
| Stration/Warezov | Worm | 0.74% |
| Basine | Trojan | 0.70% |
| Lineage/Magania | Trojan | 0.44% |
| Bagle | Worm | 0.43% |
| Zlob/Tibs | Trojan | 0.42% |
| Buzus | Trojan | 0.39% |
| Small | Trojan | 0.35% |
| Backdoor-misc | Trojan | 0.33% |
| VB | Worm | 0.31% |
| Dropper-misc | Trojan | 0.28% |
| Alman | Worm | 0.23% |
| Sality | Virus | 0.19% |
| Downloader-misc | Trojan | 0.18% |
| Mywife/Nyxem | Worm | 0.15% |
| FunLove/Flcss | Worm | 0.15% |
| FakeAV | Trojan | 0.14% |
| Fujacks | Worm | 0.12% |
| Murlo | Trojan | 0.09% |
| Autorun | Worm | 0.09% |
| Delf | Trojan | 0.08% |
| Others[1] | | 1.05% |
| **Total** | | **100.00%** |

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

# MALWARE ANALYSIS

## HEADS OR TAILS?

*Peter Ferrie*
Microsoft, USA

The flip side to the section replacement technique described last month (see *VB*, August 2009, p.4) is the segment alignment technique. This technique is used by a virus which was named 'Coin' by its author, and is described here.

### MISPLACED TRUST

The virus begins by searching for files within the current directory. The virus attempts to open and map each file that is found. If the mapping fails, the virus closes the file without attempting to unmap anything. However, as with the other viruses that were written by the same author, this virus is very trusting of the contents of the file. The virus checks for an ELF signature and several fields within the supposed ELF header, but without checking that the file is large enough to support the presence of these fields. A sufficiently small file will cause the virus code to crash. A truncated ELF file, or a file with a sufficiently large value in the e_phnum field, among other things, will also cause the virus to crash, since the code contains no bounds checking of any kind.

### MISSING THE MARK

The virus is interested in executable ELF files for the *Intel* x86-based CPU, and whose ABI is not specified. The virus does not check for an infection marker, because the marker is actually the absence of something instead of the presence of something. This will be explained below.

For each such file that is found, the virus searches within the Program Header Table entries for two PT_LOAD entries in a row, with special characteristics. The virus requires that the first PT_LOAD entry has a physical address of zero, which is the file header, and which corresponds to the image base address. The second PT_LOAD entry must have a size in the file which is equal to the size in memory.

If a file is found to be infectable, the virus calculates the amount of slack space between the end of the first loadable segment and the start of the next page in memory. The virus also calculates the amount of slack space between the start of the next page in memory and the start of the second loadable segment. The file will be skipped if the space is too small for the virus body, if the first loadable segment is aligned exactly, or if the second loadable segment is purely virtual. The alignment condition also corresponds to the infection marker. That is, when a file is infected, the first loadable segment will be aligned exactly, thus leaving no room for a virus to be inserted using this technique.

### MERRY-GO-ROUND

The virus rounds up the values for the physical and virtual sizes of the first loadable segment. If the second loadable segment does not start at the start of a page, then the virus rounds down the memory offset of the second loadable segment to the start of the page. The virus increases the physical and virtual sizes of the second loadable segment by the rounding amount applied to the memory offset of the second loadable segment. It then increases the file offset of the second loadable segment by the rounding amount that was applied to the physical and virtual sizes of the first loadable segment.

If any segment entries exist after the second loadable segment, and if any of the segment entries contain a physical offset which is greater than or equal to the file size of the first loadable segment, then the virus adjusts the physical offset of the segment by adding the combination of the rounding amounts that were applied to the physical and virtual sizes of the first loadable segment, and to the memory offset of the second loadable segment. After adjusting the Program Header Table, if necessary, the virus increases the file size by the combination of the rounding amounts that were applied to the physical and virtual sizes of the first loadable segment, and to the memory offset of the second loadable segment. The idea here is that if there is already enough space in memory to hold the virus body, then it is a simple matter to create a hole in the file that is also large enough to hold the virus body.

Then the virus attempts to remap the file. The assumption is that the operation will succeed, and the variable that holds the previous mapping is overwritten by whatever result is returned. In the event of a failure to map the file, the previous mapping still exists but the virus cannot unmap it because the original pointer has been lost. This is a minor bug in the code.

If the mapping is successful, then the virus moves all of the file contents that appear after the end of the first loadable segment to a new offset. The new offset is the old address plus the combination of the rounding amounts that were applied to the physical and virtual sizes of the first loadable segment, and to the memory offset of the second loadable segment.

### DESTRUCTION PHASE 1

If the Section Header Table appears after the first loadable section, then the virus adjusts the pointer to the Section

Header Table by adding the combination of the rounding amounts that were applied to the physical and virtual sizes of the first loadable segment, and to the memory offset of the second loadable segment.

If any of the section header entries has a physical offset greater than or equal to the file size of the first loadable section, then the virus adjusts the physical offset of the section by adding the combination of the rounding amounts that were applied to the physical and virtual sizes of the first loadable segment, and to the memory offset of the second loadable segment.

While parsing the section header entries, the virus watches for a section header entry that is named '.dtors'. The '.dtors' section contains an array of functions to call during process termination. The list is terminated by a DWORD of zero. If the virus finds a section header entry that is named '.dtors', and if the first two bytes of the tail address are zero, then the virus assumes that all four bytes are zero.

## TERMINAL SERVICES

The virus wants to replace the terminator entry with the address of the virus code. Of course, it is possible to have a destructor whose address happens to be on a 64KB boundary. This would result in the lower two bytes of the address being zero. In that case, the virus will overwrite that entry instead of appending an entry to the list. This is the most potentially serious bug in the code, but the condition seems so rare that it might almost never be encountered in the real world.

Furthermore, by simply replacing the terminator with the address of the virus code, an assumption is made that another zero can be found immediately afterwards, so that the process won't crash because of a bad pointer. If the terminator entry is found, then the virus copies itself into the cavity in the file and replaces the terminator entry with the address of the virus code.

Note that if no section header entry exists that is named '.dtors', the created (and empty) cavity will still remain in the file.

## CONCLUSION

When we think about cavity infection, most of us probably think of existing cavities in the file, such that an infection results in no file size increase. 'Forcing' the cavity in this way is an interesting variation on the theme, but let's hope that the virus author has finished with his file format tricks now.

FEATURE

# DATA TAINTING FOR MALWARE ANALYSIS – PART ONE

*Florent Marceau*
CERT-LEXSI, France

Malware technologies are becoming increasingly advanced and the use of compression and cryptographic ciphering is common. Flexible design allows for capabilities such as dynamic downloading of configuration files over the network. These practices have increased considerably over the last few years. The aim, among other things, is to make analysis of the malicious file more complex and time-consuming, as well as to hide its presence.

At the same time, the use of virtualization technologies is becoming increasingly common. Desktop virtualization (*VMware*) and virtual shared hosting (*XEN*) make great use of hardware-assisted virtualization such as hypervisors in order to improve performance. These kinds of technologies will soon become standard in PCs, and could be applied directly in the BIOS (cf. *Phoenix HyperCore*).

From a security point of view, these practices generate new angles for new types of attack. Many previous studies have been published on this subject. We have seen, among other things, the use of hypervisors in rootkits/anti-rootkit techniques [1]. In this three-part series we present a different view of this contextual technical evolution in order to take advantage of full virtualization (without a hypervisor) from a security point of view, and more specifically with the aim of helping malware researchers. We will first study the use and advantages of full virtualization, and then we will describe a concrete implementation: a way to dump character strings loaded from the network and manipulated by malware in RAM. The objective is to obtain the malware configuration file in clear text, in order to understand its impact and the risks involved with it.

## 1. INTRODUCTION AND CONCEPTS

### 1.1 Full virtualization

Generally speaking, we use a hypervisor to accelerate emulation. When the host system and the emulated guest use the same instruction set (which is usually the case with *VMware* or *XEN*, from x86 to x86) a major acceleration can be achieved by executing some part of the guest machine code directly on the host processor. Meanwhile, full emulation is slower, and will simulate the guest processor behaviours for each opcode.

All reverse engineers know how important it is to have good tools. A good user-mode debugger for *Windows* like *OllyDbg* is very useful, but is not as powerful as a kernel-mode debugger like *SoftICE*, which is more complete and allows both kernel- and user-mode debugging. Unfortunately, when working in certain debugging contexts – like working on an MBR (Master Boot Record) rootkit such as BootRoot or Mebroot/MaOS [2] – a kernel-mode debugger will not be sufficient. Most will load too late. Moreover, most kernel debuggers are OS-dependent. In the case of *WinDbg*, to debug an MBR, we need two machines with a null modem link (which can be achieved with a virtual machine). It became apparent to us that the most generic and efficient debugging platform is not the kernel debugger but the virtual machine itself. The biggest constraint here is the lack of debugging symbols.

The ultimate debugging platform is an in-circuit emulator (ICE). This piece of hardware is clearly the most efficient, but it has disadvantages: it is expensive and it is fully hardware-dependent. Thus, it seems that debugging directly via a virtual machine is the cheapest solution. Cost-free from the hardware point of view, it can be applied to any architecture – given that you can emulate it. This kind of technique provides debugging capabilities that are sometimes even better than an ICE. Indeed, hardware in-circuit emulators are directly supported by the architecture's debugging capabilities. The x86 architecture, for example, has four debug addresses that point to a maximum of a DWORD; using a virtual machine we can monitor a maximum of 4 x 4, 16 octets of memory with memory watchpoints. Finally, in a special debugging context such as reversing a BIOS, using a virtual machine is so difficult that the cost of an ICE becomes irrelevant: indeed, the BIOS code is the most hardware-dependent that exists and is difficult to virtualize correctly.

Using full virtualization we can easily modify the internal state of the CPU. It then becomes easy to modify an opcode interpretation, to break the code execution on a chosen mnemonic or to obtain a complete execution flow. Moreover, by modifying the CPU's Memory Management Unit (MMU), we act directly between the RAM and the CPU; we can then monitor any RAM access arbitrarily. This allows us to create a memory watchpoint on the two first *Go* of RAM if needed. We refer here to the use of *Qemu* in full virtualization mode on the *Anubis* sandbox [5]. *Anubis* applies instrumentation to the emulated CPU in order to monitor the call and int opcodes to watch all API and system calls emitted from the monitored code. This monitoring technique is hidden since it takes place directly inside the emulated hardware and cannot be detected as it would be on hook-based solutions like *Capture HPC* or *Microsoft Detours*.

A lot of research concerning automatic and generic unpacking methods has been carried out and also uses the full virtualization concept for code instrumentation.

It is based on the fact that the intersection between the deobfuscation code and the host code can be singularized by execution of a piece of code that was previously a data zone used by the deobfuscation code. The *Pandora's Bochs* [6] and *Renovo* [7] engines equip the emulator to follow data propagation in order to detect this intersection. Unfortunately, these implementations have detectable parts so they aren't fully hidden (*Renovo* in particular uses a kernel module), and generally they use an abstraction level that is too high – the use of virtual addresses, for example, allows evasion (cf. Skape [8]). Note that the need to track and differentiate data pages from code pages is quite similar to what is implemented to emulate the NX bit (non executable) that is missing on old processors (cf. the PAGEEXEC implementation in PaX [9]). Indeed, we can use the desynchronization between the data TLB (Translation Lookaside Buffers) and the code in order to differentiate data from code pages for the page fault handler (Interrupt 14) and eventually detect the execution from a data page. This mechanism is used by SAFFRON [10] rather than using a virtual machine. As we'll see later, our own implementation will use full virtualization to apply instrumentation in order to monitor data flow.

Obviously, this technique isn't perfect – we'll see later that it has some constraints due to the nature of full virtualization.

This concludes the theoretical part. Many open source emulators are available; to emulate an x86 platform we can use the *Bochs* [11] emulators that provide many instrumentation capabilities but which are slower than *Qemu*. While *Qemu* is faster its optimization mechanisms make it quite difficult to instrument.

## 1.2 Context

Nowadays, many pieces of malware have banking credential-stealing capabilities. To this end, they use regular expression keywords for each targeted bank. For flexibility, such malware downloads its configuration files over the network. In this way the configuration can easily be upgraded.

These configuration files are compressed and/or cryptographically ciphered in order to remain hidden from the network flow. Moreover, for flexibility, some malware uses different executable modules for each of its functionalities that can then easily be upgraded through the network.

Our objective here is to automatically process these pieces of software in order to obtain the clear text configuration file, and the process must be independent of the cryptographic cipher or compression algorithm used.

Observation shows that malware will download its ciphered configuration file from the Internet and will then uncompress/decipher it for use. There is necessarily a period of time in which the malware applies a transformation to the ciphered data and then stores it in memory as clear text (on the deciphering algorithm). We need simply to dump this data during this brief period of time.

To achieve our goal we need two things:

- The ability to track the full propagation of the monitored binary code (malware);
- The ability to dump all data originating from the network and that is manipulated by our tracked malicious binary.

By fulfilling these two conditions, we can force the dump of the clear text configuration file (among other data), and this is the case even if the analysed malware doesn't keep any instance of its clear text configuration file (for example if it re-encodes or destructs the configuration file after using it). To achieve this, we use data tainting.

## 1.3 Presentation of data tainting

Briefly, data tainting is a mechanism that allows us to track the full propagation of a given set of data on an information system.

Let's take a simple example of data tainting in RAM. For a memory zone named A of x tainted octets (to be tracked), a simple memcpy of x octets from zone A to zone B means that zone B will be marked as tainted too. A simple implementation is to use a RAM mirror called a taintmap, which contains for each RAM octet a 'tag' octet that keeps the tainting information. In the previous example, during the memcpy from zone A to zone B in RAM, there will be a similar memcpy on the taintmap from the tainted information corresponding to zone A (of x octets) to the corresponding zone B.

Let's examine a more concrete scenario. We want to track the propagation of data originating from the network (the classic case of a downloader that loads its payload onto the hard drive before executing it). This data came from the network and is stored on the network card cache. The kernel will load this data via IO or via the DMA, and copy it into the user-mode buffer of the application that requested this network resource. Finally, our application will request the kernel again in order to create a file to store this data.

In such a scenario, since we need to track all the incoming network data without filtering, we simply have to hook the emulator part that handles the network card cache in order to mark all incoming data as tainted as it is loaded into RAM (via the IO or the DMA). Our tainted data in RAM will then be propagated through the taintmap during all the processing that the data goes through. When the data is copied to the user-mode buffer it will retain its taint marks.

Note that we work here at the hardware level, since we are OS-independent. This means that when the OS frees a heap buffer that contains some tainted data, the data will continue to be resident in RAM and consequently the tag will persist. It is only when the buffer is reallocated that the data is overwritten and the tainted tag will also be overwritten with the new tainted data values.

A problem may appear when malware stores information on the hard drive. This requires that we extend the data tainting mechanism and propagate tainting information through the hard drive. The mechanism is exactly the same as for the network card cache: we only have to propagate tainting information (tags) for each exchange between the RAM and the hard drive via the IO or the DMA. Obviously, unless a very low capacity drive is used, we can't mirror the hard drive as we did for the RAM. In normal conditions, there is a low volume of tainted data compared to the drive size. Moreover, if this data is stored on the hard drive before any arithmetic processing, there is a low risk of loss of these tainted marks (more details on this later). We can then consider this data as mostly contiguous. From this observation we decided to store the hard drive tainting mark as a table of offset and size, using an offset similar to the LBA (Logical Block Addressing).

Let's examine the data tainting internals in more detail.

For previous examples we just used simple hooks on different data channels (IO/DMA) in order to propagate tainted data, but the RAM propagation mechanism is more complex. The simple memcpy of tainted data can itself take several different forms.

A memcpy implemented with a simple repz movsd will be different from a memory loading and storing via the register repeated on a loop. Indeed, the second case also implies a register-level propagation (anyway, registers can easily be mirrored). But the problem is really more complex. Indeed, in many cases we do not simply move data from one place in RAM to another; the data will be loaded, go through a lot of arithmetical processing and comparison before being stored in RAM.

Let's consider an example where we use data tainting to track the propagation of packed binary code that injects itself into other processes. The code must keep its tag even after the unpacking operation so that we can continue to monitor its propagation.

Therefore, we have a binary mapped image in RAM that is tagged; this image will read itself as a data sequence and decipher it to generate the unpacked code. Since many packers use several cryptographic layers, the tainting mark can get lost in the heavy arithmetic process involved. Indeed, while it is logical to say that during the execution of a mnemonic 'add REG, IMM', REG will keep its tainted tag, what would happen during a bit permutation? It is in those kinds of cases that the propagation becomes increasingly complex. As you can see, the tainted tag propagation between the RAM and CPU requires instrumentation of each virtual CPU mnemonic to identify the potential propagation of the tag for a given mnemonic. The two most common open source data tainting implementations are *Taint Bochs* [12] for *Bochs* and *Argos* [13] for *Qemu*.

In the next part of this series (next month) we will study the inherent limitations of an efficient propagation and the overall limitations of this type of solution.

## REFERENCES

[1] Rutkowska, J. Subverting Vista Kernel for Fun and Profit. http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Rutkowska.pdf.

[2] Florio, E.; Kasslin, K. Your Computer is Now Stoned (...Again!) The Rise of MBR Rootkits. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/your_computer_is_now_stoned.pdf.

[3] JTAG BDI2000. http://www.abatron.ch/products/bdi-family/bdi1000-bdi2000.html.

[4] Arium ECM-700 JTAG Emulator. http://www.arium.com/product/?prod_id=56.

[5] Bayer, U.; Kruegel, C.; Kirda, E. TTAnalyze: A tool for analyzing malware.

[6] Bohne. L. Pandora's Bochs: automatic unpacking of malware. http://www.damogran.de/PandorasBochs.pdf.

[7] Kang, M.G.; Poosankam, P.; Yin, H. Renovo: A hidden code extractor for packed executables. Proceedings of the 5th ACM Workshop on Recurring Malcode (WORM 07), October 2007.

[8] Skape. Using dual-mappings to evade automated unpackers. http://uninformed.org/?v=10\&a=1.

[9] PaX Team. Design of PAGEEXEC. http://pax.grsecurity.net/docs/pageexec.txt.

[10] Quist, D. Covert debugging circumventing software armoring techniques. http://www.offensivecomputing.net/bhusa2007/dquist-valsmith-covert-debugging-paper.pdf.

[11] Bochs IA-32 Emulator. http://bochs.sourceforge.net/.

[12] Taint Bochs Understanding Data Lifetime via Whole System Simulation. http://www.stanford.edu/~blp/papers/taint.pdf.

[13] Argos: An emulator for capturing zero-day attacks. http://www.few.vu.nl/argos/.

# CONFERENCE REPORT

## BLACK HAT 2009 CONFERENCE REPORT

*Andrew Lee*
K7 Computing

Black Hat USA is really two things: first, a series of workshops and training sessions, followed by a two-day technical conference, which also has an exhibition floor where vendors can display their wares. It draws some of the top security practitioners and speakers in the world, and the keynotes and sessions are usually of a high quality. It is certainly one of the most well-attended conferences in the security industry, and is followed by the cheaper, scruffier (but some would argue, more useful) DefCon hacker conference. Black Hat takes place each year in Las Vegas at the huge Caesar's Palace casino complex – a place so wonderfully bizarre that, upon visiting, one is reassured that the rest of the world is not quite as crazy a place as one suspected.

The advantage of Caesar's Palace is that it's big – really big. It's big enough to have its own shopping street complete with animated fountains and an aquarium in which you could sink a large warship, and it's big enough to accommodate the 4,000+ attendees and 15 tracks of conference programme that make up the Black Hat conference. The disadvantage of Caesar's Palace (at least to someone who is currently mobile only with the assistance of crutches) is that it's big – really big. Big enough to mean a five- to ten-minute walk between sessions (not to mention the 20-minute walk from one's room – in the same hotel), for which task one will also need to consult the map supplied in the programme.

Fortunately, the sessions are of reasonable length, meaning that most finish within their allotted time, leaving sufficient time to get to the next session (although you may, even then, be too late to find a space inside the room of a popular session). What's more, the entire conference proceedings can be bought on DVD, as good quality audio/video recordings showing both the speakers and their presentations. My suggestion to anyone attending is to purchase the DVD to give you the flexibility to pick the sessions you really want to attend (and that might be within reasonable walking distance), while not having to worry about conflicts.

Before any of it starts, though, you have to navigate the very long queues to pick up your delegate badge. This year an RFID badge was supplied along with the printed badge, and was duly hacked in the RFID workshop. As I was handed mine it was suggested that a few seconds in a microwave would prevent any problem. However, since 'Andrew' and 'K7 Computing' was the extent of the information contained in my badge I didn't spend much time on that solution, preferring instead to risk the consequences of having my name exposed via RFID (if, for some strange reason, it couldn't simply be read from the printed badge itself).

Although it was tempting just to place my DVD order and go and sit by the wonderfully tasteless Roman-themed pool, I resisted this urge (the 40°C+ heat was also a deterrent) and bravely plunged into the programme.

### PRESENTATIONS

As mentioned, there are 15 conference tracks; eight tracks on the first day, seven on the second, and a keynote each day. Past keynotes have been very good, and are always well attended, and this year was no exception. I couldn't get in the door for David Merrill's (*EMI*, *Google*) speech but there were relays to several other rooms with screens. The crux of the presentation was that most companies get security wrong because they assign the responsibility to the security team, rather than making security the responsibility of each person. The second keynote was by Robert Lentz from the US Department of Defense, but unfortunately I was unable to attend due to a prior appointment with a pillow and warm bed. But, back to day one…

Unfortunately, the first session of the conference proper turned out to be a rather damp squib. Although the speaker has since garnered quite a bit of publicity for some hacking activities in Zurich airport (http://peterkleissner.com/?p=34) and for getting fired from his position in an AV company (http://web17.webbpro.de/index.php?page=peter-kleissner), Peter Kleissner's talk on the 'Stoned' (yes, he really did call it that) bootkit, was pretty uninteresting in terms of new information. While the fact that he showed that you can bypass disk encryption (at least, pass information across) and infect files from outside the OS was an interesting twist, there was little else new in this. It is pretty obvious that you can patch things offline, and as Kleissner admitted, the technique he is using is 20 years old. The fact that technical problems meant that his demos didn't work was also a factor, though one can't really blame the speaker for that. It is also unfair to criticize an 18-year-old for his presentation technique, particularly in a second language, but his inexperience generally made for a very difficult-to-follow presentation. The paper itself is clearer, and can be found on the Black Hat website.

After that, things picked up considerably and I managed to hit a run of great presentations for the rest of the day. Andrea Barisani and Daniele Bianco, two Italian students, gave what was probably the funniest presentation of the conference, including a video performance. Their boundless enthusiasm for the subject was refreshing, and the audience responded warmly as they explained how you could use cheap electronics to monitor keystrokes remotely via the power grid. Not content with one cool hack, they then showed how they could use cheap lasers to remotely read what someone is typing on a laptop – a feat which gains extra points for the use of laser beams.

After lunch I took in a smattering of a couple of different presentations, including the rather good 'Netscreen of the Dead' by Graeme Neilson, whose presentation included screenshots from many classic zombie movies, providing a good accompaniment to his discussion of creating a trojaned OS for *Juniper*'s *Netscreen* appliances.

Heading back to the Rootkits stream, Jeff Williams (not the Jeff Williams many of us know from *Microsoft*, but rather the CEO of *Aspect Security*) gave an excellent talk on the dangers of Java applications in the enterprise environment; particularly financial institutions that are highly reliant on such applications. He showed how little code it would take to steal data, cause damage or install other malicious programs inside Java applications – something that could be achieved by bribing or coercing a dissatisfied developer.

The second day started out with a great presentation on attacking SMS by Zane Lackey and Luis Miras. It seems I wasn't the only one interested in their exploits of SMS on *iPhone* and *Google*'s *Android*, as the room was packed with people standing five deep outside the doors trying to catch what was being said. Their talk centred around problems caused by the phone operating systems failing to validate the source of SMS service messages, meaning that they were able to set up their own servers, and have phones pick up the messages from there rather than the legitimate servers. This was only one of several excellent presentations in the 'Mobile' track.

Throughout the two days there is also a Panels track, which is always worth a quick look. I took in a few minutes of the 'Hacker Court' which examined a fictitious but legally accurate case in a mock trial, which was entertaining if nothing else for the use of a rather amusing British pejorative as the nickname of one of the defendants.

Unfortunately, I had to leave for the airport at lunchtime, so missed the afternoon sessions, but rather fittingly, the conference was rounded off with Mikko Hyppönen speaking on Conficker in the Turbo Talks stream. If nothing else, Conficker has taught us that the security industry will be around for a long time, as we're still suffering the same problems as we have faced for the last 20 years. I'm sure Black Hat will be around for a long time too, and I hope to see some of you there next year.

The papers from this year's conference (as well as audio and video material) can be found at http://www.blackhat.com/html/bh-usa-09/bh-usa-09-archives.html.

# PRODUCT REVIEW

## G DATA TOTAL SECURITY 2010

*John Hawes*

*G Data Software* was founded in Bochum, Germany in 1985 and has been in the anti-virus business for over 20 years. Alongside its long-time flagship product formerly known as *AntiVirusKit* (*AVK*), the company now produces a range of security offerings including business-oriented protection software, and has an expanding global presence.

The company may not have the highest profile among security firms, outside its native Germany at least, but the name will be well known to regular *VB* readers thanks to a consistent record of excellence in VB100 testing. The company's products have built up an impressive chain of passes, with only two fails since 2003 – one for a fairly minor false positive, the other from an issue with scanning floppy disks, back in the days when such things were still important. In recent comparative reviews our new RAP rating system has highlighted the superb detection capabilities provided by the company's dual-engine approach, placing it consistently among the leading performers in our RAP quadrants. Similarly impressive scores have been achieved in other independent tests.

With the latest complete suite product promising a wide range of extras in addition to the top-class anti-malware protection, we decided it was time to take a closer look.

### WEB PRESENCE, INFORMATION AND SUPPORT

The company operates a number of websites in a variety of languages, with the main English-language hub to be found at gdatasoftware.com. From here, users can navigate to a localized site for their region, most of which seem to offer much the same experience. The site is simple, clear and responsive, and is happy to run with scripts disabled – something which far too many security firms seem to consider an unacceptable impediment to their marketing efforts. The home page is heavily product-focused, with a list of the full product range given pride of place beneath a colourful advertisement for the latest 2010 version. This is accompanied by details of recent favourable reviews and the latest upgrade opportunities, and followed by a selection of news items on recent security issues. A selection of links lead to the main subsections of the site, top of the list being the inevitable online shop and access to free trials, which seem to be offered for most of the home-user product range.

This is followed very sensibly by the support section, which continues the plain and simple layout of the rest of the site; a large and clear search box is the main item, and contact telephone numbers – so often these days buried out of sight to prevent any chance of human contact – are displayed prominently on every support-related page. An online contact form is also provided, along with detailed contact information for local and global offices. For more standard support issues, a well-stocked FAQ covers a range of common issues with clear and sensible solutions, and is easily searchable. A downloads section provides access to more detailed manuals, although these are not yet available for the new 2010 edition, as well as a selection of additional tools, including a bootable CD image for those tricky cleaning jobs.

The 'Security Labs' section provides a more general range of advice and assistance, with information on malware and malware issues, a well-stocked library of news stories and alerts, and a set of handy tips and tricks to improve security in general, including advice on password selection and backing up of data. Some fun statistics are also available, with maps and graphs showing malware and spam outbreaks (including an amusing 'massiveness' meter), number of active zombies and so on.

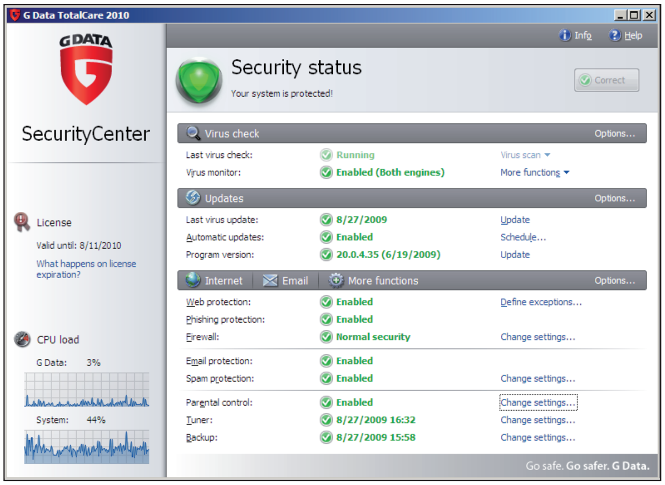
Finally, a selection of data is provided on the company and its business partners, including an impressive list of recommendations from existing customers, predominantly in Germany. Technology partners also include a roster of collaborations with leading security firms.

Having perused the information available online, and not yet having thought of any issues which might warrant testing the support system, we took our copy of the top-of-the-line *Total Security* product into the lab.

## INSTALLATION, CONFIGURATION AND ASSISTANCE

Installation of the product is fairly straightforward, with a few pauses at various stages on lower-powered systems but generally fairly speedy. Options are provided to install additional components, including parental controls and a data shredder, and the now standard community system can also be joined (or not) at this stage. With little further ado things are up and running, but of course an update is required to bring things up to speed.

Somewhat surprisingly, the build provided for download seems not to include any detection data added in the last few months, so a rather lengthy initial update is required. It would seem that the extra labour involved in keeping the standard online build reasonably up to date would be balanced out by the reduced strain on update servers, and the better immediate protection provided to users, but doubtless there are other factors involved too.



Security status

Your system is protected!

License

Valid until: 8/11/2010

What happens on license expiration?

Last virus check: ✓ Running

Virus monitor: ✓ Enabled (Both engines)

Last virus update: ✓ 8/27/2009

Automatic updates: ✓ Enabled

Program version: ✓ 20.0.4.35 (6/19/2009)

Web protection: ✓ Enabled

Phishing protection: ✓ Enabled

Firewall: ✓ Normal security

Email protection: ✓ Enabled

Spam protection: ✓ Enabled

Parental control: ✓ Enabled

Tuner: ✓ 8/27/2009 16:32

Backup: ✓ 8/27/2009 15:58

Once up and running, the product presents a very appealing interface, reflecting the unfussy feel of the company web presence. The design is fairly standard, with a list of the various components and modules along with status information and links to configuration and control, with a fairly large pane down the left-hand side dedicated to licensing information and some nice little graphs of system and scanner load. Checking quickly through the various options menus showed immediately that a commendable depth of control is available. This all seemed at first glance to be logically laid out and accessible; we decided to look at each section in greater depth later on, pausing only for a brief skim through the help system.

The help system is accessed via a link from the main GUI window, but only from the main page, with few additional contextual links from within the various subsections of control, which is often a fast and useful way of accessing information on a specific subject. The information provided is pretty thorough and generally clear and lucid, with just the occasional infelicity of style or grammar hinting at translation issues. Though rather short on screenshots and links to control areas back in the main product, it covers the ground pretty thoroughly, with a very nice selection of 'tips' guiding users through performing specific tasks, rather than simply detailing what each button or checkbox is intended to do.

All in all, the product seemed pretty well designed and laid out on the surface; it was time to see if it still had what it takes under the hood.

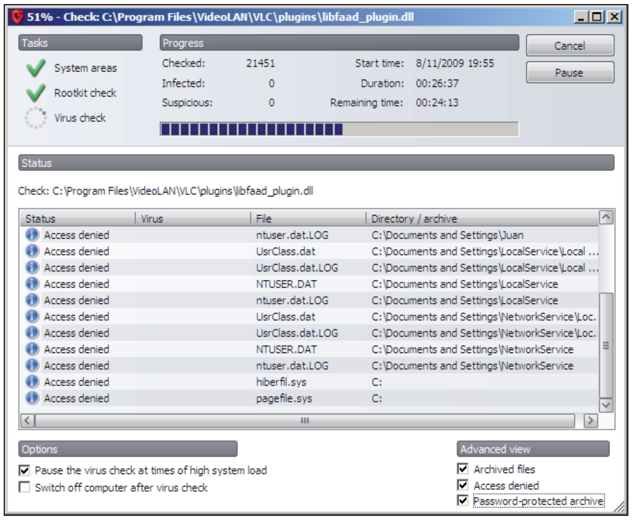
## SYSTEM PROTECTION AND MALWARE DETECTION

We have already noted *G Data*'s consistent excellence in terms of malware detection, demonstrated by superb performances in VB100 and other tests. The product's high level of detection is assured by the use of two separate engines, which have occasionally been switched in the past as different developers and labs prove themselves worthy of inclusion. In the past the *Kaspersky* engine, so popular with OEM products, has been a stalwart component, but this time *G Data* has opted to move on to newer ground, bundling together engines from *BitDefender* and *Alwil*. With both these engines doing extremely well of late, the combination promised to provide as good if not better detection rates while slightly reducing scanning times in some areas, judging by our recent measurements in VB100 comparatives.

Running the product over our sample collections proved this to be right on the money, with all sets totally destroyed by the scanner, which easily handled just about everything we threw at it. On the evidence of the range of impromptu scans we carried out, *G Data* looks set to further improve its excellent ranking in our RAP quadrants over the next few months. Scanning proved solid and stable, with no problems handling our sets of difficult malformed files which have tripped up many others in the past, and the on-access monitor held up well under extremely heavy bombardment from all directions.

No behavioural or HIPS-type blocking is included in the product, but with both the engines included showing some great scores in the reactive part of our RAP sets lately, protection against new and unknown malware should be about as good as it can be with static scanning using advanced heuristics and generic detection. We also found the overheads imposed to be fairly reasonable despite the two-pronged approach, with most systems functioning perfectly well and even the low-powered netbook barely registering any slowdown in normal operations.

On-demand scans, designed to be as fast and thorough as possible, do impose considerable restrictions on using the system for anything else while they are being carried out, but an option is provided to cede control when the user wants to get on with something else. This can even be enabled mid-scan, taking a few moments to take effect but soon returning the machine to full speed, and resuming high-power scanning once resources are made available. There is also a pretty decent caching system which causes files previously scanned or listed in whitelists to be ignored; this kind of technology is not yet accurately covered by our comparative speed measurements, although we hope to introduce an updated system in the near future. As a slightly less than scientific measure however, it seemed pretty clear that speeds picked up considerably once the scanner got to know the local system.


51% - Check: C:\Program Files\VideoLAN\VLC\plugins\libfaad_plugin.dll

Checked: 21451 Start time: 8/11/2009 19:55
Infected: 0 Duration: 00:26:37
Suspicious: 0 Remaining time: 00:24:13

Check: C:\Program Files\VideoLAN\VLC\plugins\libfaad_plugin.dll

| Status | Virus | File | Directory / archive |
|---|---|---|---|
| Access denied | | ntuser.dat.LOG | C:\Documents and Settings\Juan |
| Access denied | | UsrClass.dat | C:\Documents and Settings\LocalService\Local ... |
| Access denied | | UsrClass.dat.LOG | C:\Documents and Settings\LocalService\Local ... |
| Access denied | | NTUSER.DAT | C:\Documents and Settings\LocalService |
| Access denied | | ntuser.dat.LOG | C:\Documents and Settings\LocalService |
| Access denied | | UsrClass.dat | C:\Documents and Settings\NetworkService\Loc. |
| Access denied | | UsrClass.dat.LOG | C:\Documents and Settings\NetworkService\Loc. |
| Access denied | | NTUSER.DAT | C:\Documents and Settings\NetworkService |
| Access denied | | ntuser.dat.LOG | C:\Documents and Settings\NetworkService |
| Access denied | | hiberfil.sys | C: |
| Access denied | | pagefile.sys | C: |

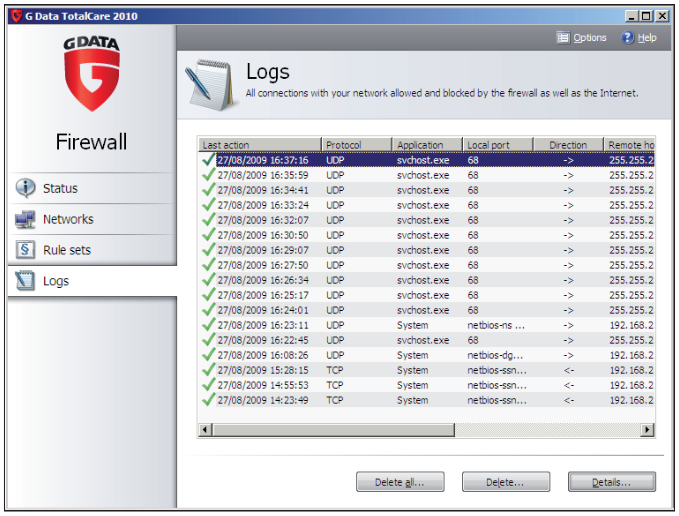
Options
☑ Pause the virus check at times of high system load
☐ Switch off computer after virus check

Advanced view
☑ Archived files
☑ Access denied
☑ Password-protected archive

Hampered as ever by a lack of time to go into too much depth, with yet another comparative to prepare for and the annual VB conference fast approaching, we weren't able to look as closely at removal and disinfection as we would have liked, but the selection of items we did manage to get installed on a system were easily and cleanly removed once those all-encompassing definitions were updated to provide detection. Throughout we found the control system to be both simple to use and impressively thorough, with no option we could think of that was either absent or even difficult to find. For both the on-access monitor and the on-demand scanner, the option to use one or other of the two engines (coyly referred to as 'Engine A' and 'Engine B') is provided, with the default being to use both in both modes. Engine A is described as having stronger detection but slightly lower performance; Engine B is recommended for faster scanning speed but not such complete detection.

The simplicity of operation extends to the web and email protection, which is given a separate section in the main interface but closely tied in with the anti-malware scanner. Traffic via HTTP, IM and email (both inbound and outbound) is scanned, with a few configurables such as size limit for scanning downloaded files and attachments, the addition of ports to scan and so on. The user can also opt to report infected sites back to base, to improve protection for the whole community.

Moving on to the other main protective element, the suite of course includes the now-obligatory firewall. In this area there are some simple measurements of success: the efficacy of the standard settings, the intrusiveness or lack thereof on the user experience, and the usability of the fine-tuning for more advanced users. In all three *G Data* scores pretty highly. By default, the firewall operates entirely on 'auto-pilot', taking a selection of standard rules and creating new sets for whatever network-enabled software is found running on the system. This all seems pretty well thought out and effective, and the whole experience is completely transparent to the user, with none of the training periods



Logs

All connections with your network allowed and blocked by the firewall as well as the Internet.

| Last action | Protocol | Application | Local port | Direction | Remote ho |
|---|---|---|---|---|---|
| ✓ 27/08/2009 16:37:16 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:35:59 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:34:41 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:33:24 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:32:07 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:30:50 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:29:07 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:27:50 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:26:34 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:25:17 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:24:01 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:23:11 | UDP | System | netbios-ns ... | -> | 192.168.2 |
| ✓ 27/08/2009 16:22:45 | UDP | svchost.exe | 68 | -> | 255.255.2 |
| ✓ 27/08/2009 16:08:26 | UDP | System | netbios-dg... | -> | 192.168.2 |
| ✓ 27/08/2009 15:28:15 | TCP | System | netbios-ssn... | <- | 192.168.2 |
| ✓ 27/08/2009 14:55:53 | TCP | System | netbios-ssn... | <- | 192.168.2 |
| ✓ 27/08/2009 14:23:49 | TCP | System | netbios-ssn... | <- | 192.168.2 |

or constant deluges of requests for permission favoured by many systems. For the average user, a very satisfactory level of protection from web-based attacks will be provided invisibly and with no effort whatsoever.

For the more experienced (or more paranoid), there is of course the option to delve deep into the settings and configure things exactly to one's liking. Such systems are often complex and bewildering, but once again *G Data* has gone to considerable efforts to provide even less skilled and knowledgeable users with some access to fine tuning. A simple and pleasant wizard system is provided to lead the user through the steps of designing and creating a rule or ruleset based on categories including applications, network connections and services, along with the direction to control and so on. The only thing missing would be the option to block specific applications and behaviours at a local level to turn it into a fully fledged and highly usable application control and HIPS system. Beyond these simplified controls, full and detailed configuration is also provided via an advanced tab, which again is clear and lucid. Logging is complete and detailed, with a nice clear summary available for every incident noted, whether blocked or allowed.

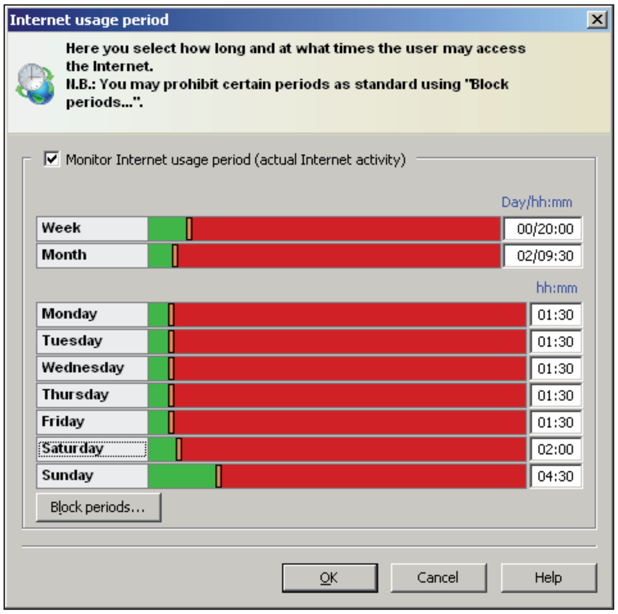
## OTHER FUNCTIONALITY

The name of the product is 'Total Security' and of course it goes some way beyond the standard components mentioned so far. There is, of course, a spam filter – another pretty compulsory component of a security suite these days. Here, the spam filter is given its own sub-GUI much like the firewall component and it is similarly kitted out with options and controls. This was something else that we weren't able to test thoroughly, our current anti-spam testing set-up being more geared towards corporate, server-level protection than the home-user end, but running through the layout we found it offered all one could want, from simple allow and block lists to detailed controls of sensitivity and response to suspected detections.

The lower entries on the main interface cover a selection of items only included in the more thorough suites. The first, and probably most common, is a parental control system to protect children from inappropriate content online. We had quite some fun playing around with this, concluding that, once again, the interface was designed extremely well. The area for defining permitted usage hours, based on specific times and/or weekly and monthly allowances, was particularly simple to implement. It even differentiates between Internet use and use of the computer for offline activities. The built-in controls offer filtering of a range of unwanted topics, based both on known bad sites and keywords by the look of things, and also a 'walled garden' approach where only a list of known-good sites can be accessed. This list seems reasonably well populated, and is fairly simple to expand with new sites and whole new categories for the committed and diligent parent to configure as they desire. The blocking system is similarly simple, with new unwanted URLs or keywords easy to add.

In implementation, things seem a little less complete however, with a few quirks of behaviour noted when exploring. Occasionally, sites on the allowed list would fail to display, then subsequent pages visited would appear to be masquerading as the missing page, and some of the sites and categories appeared to be in different languages, indicating that the localization of this section is not fully complete. It seems likely that the user's mileage with this tool will vary depending on their location. The opposite method, of blocking unwanted content but generally allowing access, also had a few oddities, failing the 'Scunthorpe' test and apparently defusing some of the blocks if an 'exception' keyword is included on the same page. So, a fairly decent stab at a control system, with some excellent configurability, but perhaps a little lacking in the sophistication of the most advanced examples of the genre. Full logging of all activities is included of course.

Moving on, we find a section labelled 'tuner', which offers a lot more than the simple clean-up of excess files provided by some of the other suites we've looked at. Not only does it clear up the various temporary and cache items collected on the hard disk of a well-used machine, it also probes through the registry for unnecessary dross and checks through the system settings to check that a selection of basic security measures have been taken. Dividing these into security, performance and privacy-related issues, all are enabled by default but can be deactivated, and a trial run can be performed which will produce a list of changes found to be necessary, but without actually implementing them. Full scheduling and logging is provided, and an undo



Internet usage period

Here you select how long and at what times the user may access the Internet.
N.B.: You may prohibit certain periods as standard using "Block periods...".

☑ Monitor Internet usage period (actual Internet activity)

| | Day/hh:mm |
|---|---|
| Week | 00/20:00 |
| Month | 02/09:30 |

| | hh:mm |
|---|---|
| Monday | 01:30 |
| Tuesday | 01:30 |
| Wednesday | 01:30 |
| Thursday | 01:30 |
| Friday | 01:30 |
| Saturday | 02:00 |
| Sunday | 04:30 |

feature can roll back any changes subsequently found to be inappropriate. It all seems pretty thorough, without including any potentially harmful activities, and works surprisingly speedily. Running on a tired old netbook which has seen a great deal of software installed and removed of late, it certainly seemed to make a discernable difference to the system performance, and cleared out all the unnecessary and potentially sensitive information we could think to look for.

The final option in the main interface is a back-up facility, again provided with its own interface which continues the uniformity of design and layout of the rest of the product. Simple back-ups, on demand or scheduled, can be set up to archive specific areas and file types as required, with the archive stored wherever the user wishes, although local hard drive partitions are not recommended; storage on network drives, including an option to post to FTP sites, is preferred, but local archives can be created and burned to CD if desired. Yet again, configuration is both highly in-depth and simple to navigate, and logging is fairly thorough. There is even a system to administer previous back-ups, to remove older or unwanted data.

Having run through such a broad range of utilities, we thought we must surely be at the end of *G Data*'s offerings, but there remains one more item worth looking at: the shredder mentioned briefly during installation. This has no entry in the main interface but simply provides a desktop icon onto which items can be dropped for secure deletion. It doesn't seem to have any sort of configuration, eschewing the choice of destruction types offered by some similar utilities, but it does its job in a simple and perfectly effective way without seeing the need to trouble the user with choices of what kind of military-grade, DoD-certified, multi-level-overwriting to perform.

## CONCLUSIONS

Somewhat overwhelmed by the breadth of this suite, we arrive at the end of this review and remain thoroughly impressed. The coupling of the exhaustive protection of the dual-engine approach with equally exhaustive additional components will doubtless appeal strongly to the more demanding user, who will find little to complain about here other than the lack of full-blown HIPS. Where *G Data* has really scored, though, is in the layout and design of the product, opening up its many wonders to a much wider audience beyond the more technically inclined. In the vast range of products we see in the *VB* lab there is a strong tendency to sacrifice configurability for usability, or vice versa, and when a product manages to combine the two effectively it stands out from the crowd.

As security suites mature as a software type and become standard items on every desktop, the range of utilities they offer continues to expand and the quality of those components increases. While a few of the lesser items included here may still lag a little behind the very best in their specific fields, the provision of such a broad range in a single package, and moreover with a single, unified approach to operation and control, will open up new horizons of safety to a wider audience.

Something else worthy of note is the improvement in speed and reduction in resource usage. While previous iterations may have been rather hefty for many users, and while increasing power in desktop systems has led many developers to believe they can get away with growing overheads, *G Data* has greatly improved the performance of its product without noticeably reducing its ability to protect users. Doing away with the issue of sluggish and unresponsive systems, which many would cite as a major reason to shun the multi-engine approach, could signal *G Data*'s emergence as a truly major player on the security scene.

**Technical details:**

*G Data Total Security 2010* was tested on:

*Intel Pentium 4* 1.6 GHz, 512 MB RAM, running *Microsoft Windows XP Professional SP2.*

*AMD Athlon64* 3800+ dual core, 1 GB RAM, running *Microsoft Windows XP Professional SP3* and *Windows Vista Business Edition SP2.*

*Intel Atom* 1.6 GHz netbook, 256 MB RAM, running *Microsoft Windows XP Professional SP3.*

# COMPARATIVE REVIEW

## ANTI-SPAM COMPARATIVE REVIEW SEPTEMBER 2009

*Martijn Grooten*

This month's VBSpam comparative review sees an increase in the field of competitors for the third time in a row. Starting out with a modest six products on the test bench in the first test (see *VB*, May 2009, p.S5), this month sees that figure doubled, with a total of 12 products lined up on the bench: eight of the products that took part in the last VBSpam test (see *VB*, July 2009, p.25) are joined by four new ones. To date, there have been few other anti-spam tests with as many participating products.

We hope that our tests will make a valid contribution to the anti-spam community, helping the community answer questions about which anti-spam methods work better than others, and helping developers find ways to improve their products. For me, the best part of conducting these tests is hearing developers say that they have made improvements to their product upon receiving our feedback on its performance.

A total of nine VBSpam awards were given out this month, but only one of these was at the Platinum level, leaving most developers with something to improve upon. But even those achieving a Platinum award have good reason to look carefully at their product's performance: with spam changing constantly, a filter that isn't kept up to date – even a very good one – will soon start to fall behind.

## THE TEST SET-UP

A few changes were made to the set-up after the last test. However, these were mostly of a technical nature and designed to help the test run more smoothly, and thus should not have affected the test itself. The full methodology can be found at http://www.virusbtn.com/vbspam/methodology/. Readers who wonder about the relatively high false positive rates measured in our tests (compared with those seen in other tests and those claimed by the developers themselves) are advised to consult the last review (see *VB*, July 2009, p.25) for explanation. Finally, as has been mentioned previously, the nature of this test is comparative, and as such it is important to note that it is not so much the absolute performance of a product that matters, but the relative performance compared to that of its competitors.

The products that needed to be installed on a server were installed on a *Dell PowerEdge R200*, with a 3.0GHz dual core processor and 4GB of RAM. Those running on *Linux* ran on *SuSE Linux Enterprise Server 11*; the *Windows Server* products ran either the 2003 or the 2008 version, depending on which was recommended by the vendor.

## THE EMAIL CORPUS

The test ran from 16:45 h on 7 August 2009 to 08:00 h on 27 August 2009. The corpus consisted of all emails – ham and spam – sent to '@virusbtn.com' addresses mixed with a spam stream provided by Project Honey Pot. Emails from both sources were sent through the products in real time.

While the test was running, we noticed a downside to its popularity: with so many products running on the same network, many of which perform regular Internet look-ups, the Internet connection was put under considerable strain and during some periods wasn't as reliable as we would have liked it to have been. It is interesting to see how different products react to this situation – which could easily occur in a real-world environment – and how much their performance suffers.

But of course, we do not want uncontrolled and unannounced circumstances to influence our test. Therefore we looked carefully at the network's performance while the test was running; emails that were received during periods for which we cannot be absolutely sure the network performance was reliable have been eliminated from the test. It should be noted that this has been done without looking at whether or how this affected individual products, and the final corpus used was still large enough for the results to give a good reflection of the products' performances.

This corpus contained 1,275 ham messages and 19,401 spam messages sent to *VB* addresses. It also contained 294,338 spam messages from Project Honey Pot; these emails reflect the global nature of spam and the fact that different addresses and domains do sometimes receive different kinds of spam. The total corpus thus contained 315,014 messages.

### BitDefender Security for Mail Servers 3.0.2

**SC rate (total):** 98.74%

**SC rate (Project Honey Pot corpus):** 99.29%

**SC rate (VB spam corpus):** 90.41%

**FP rate:** 0.87%

**FP rate of total VB mail corpus:** 0.053%

Romanian company *BitDefender* submitted its *Linux* product for the third time this month, and its developers were eager to improve upon the Silver VBSpam award they won in July. On that occasion the product missed out on the higher-level awards because it was eager to block some legitimate emails from countries that use different character sets – a tempting idea perhaps, as a large volume of such mail is spam, but the practice could, in fact, lead to end-users missing important messages. Fixing this issue saw the product's false positive rate drop significantly, while still retaining a high spam catch rate, and as such it is the deserved winner of a VBSpam Gold award, only narrowly missing out on a top-level Platinum award.

vb Sept 2009 gold SPAM virusbtn.com

## ClamAV using Sanesecurity signatures

**SC rate (total):** 85.40%

**SC rate (Project Honey Pot corpus):** 86.43%

**SC rate (VB spam corpus):** 69.83%

**FP rate:** 0.39%

**FP rate of total VB mail corpus:** 0.024%

From the outset, the developer of the *Sanesecurity* signatures that work with *ClamAV* did not expect to win a VBSpam award for his product, which is generally used together with other solutions, but he was eager to hear feedback on its performance. After receiving feedback from the last test, the developer made some adjustments to the product, resulting in a significantly improved performance this time around. A spam catch rate of over 85%, together with fewer false positives than all but one product, is a good score indeed; not quite enough to win an award, but nevertheless impressive for a product based on many hours of voluntary work.

## Fortinet FortiMail

**SC rate (total):** 99.04%

**SC rate (Project Honey Pot corpus):** 99.20%

**SC rate (VB spam corpus):** 96.64%

**FP rate:** 2.25%

**FP rate of total VB mail corpus:** 0.135%

*Fortinet*'s *FortiMail* achieved a VBSpam Silver award for its first performance in our tests in July. An extensive logging system enabled us to provide feedback to its developers on the types of emails that were being blocked incorrectly and which anti-spam tests these emails had failed. As a result of subsequent tweaks made to the product, *FortiMail*'s false positive rate dropped, while its spam catch rate remained high; in fact, its performance against the spam received directly by *VB* (the VB spam corpus) was better than that of any other product. The product's scores were not quite sufficient to earn a Gold award, but another VBSpam Silver award should spur the developers on to do even better next time.

vb Sept 2009 silver SPAM virusbtn.com

## Kaspersky Anti-Spam 3.0

**SC rate (total):** 98.39%

**SC rate (Project Honey Pot corpus):** 99.01%

**SC rate (VB spam corpus):** 88.92%

**FP rate:** 0.63%

**FP rate of total VB mail corpus:** 0.039%

*Kaspersky's* anti-spam product, usually referred to as *KAS* and running on *Linux,* is an excellent demonstration of the fact that using a locally installed product does not necessarily mean a lot of extra work for the system administrator: from the point at which it was first set up in our test network in June, we have had hardly any reason to look at the product. As in the previous test, the product's false positive rate was very low. The spam catch rate was slightly higher this time than in the previous test, but with the benchmarks having become stricter this month, it was not quite enough to earn a Platinum award. A VBSpam Gold award is thus earned by *Kaspersky* in recognition of a better-than-average performance on both accounts.

vb Sept 2009 gold SPAM virusbtn.com

## McAfee Email Gateway (formerly IronMail)

**SC rate (total):** 99.60%

**SC rate (Project Honey Pot corpus):** 99.87%

**SC rate (VB spam corpus):** 95.62%

**FP rate:** 1.27%

**FP rate of total VB mail corpus:** 0.077%

*McAfee Email Gateway* appliance, previously known as both *Secure Mail* and *IronMail*, was originally developed by *Secure Computing*, which was bought by *McAfee* in 2008. The product was set up easily using a web interface. I was particularly interested in seeing how well the product performed in the blocking of spam as this was the only product in the test that was configured to scan the contents of incoming email during the SMTP transaction and block those emails it was certain were spam. When done well, this can save a significant number of resources.

vb Sept 2009 gold SPAM virusbtn.com

Helped by this two-layered blocking, the product's spam catch rate of 99.60% was higher than that of any other. The cost of this was a number of false positives though, including a few non-bulk emails, although it should be added that none of these were blocked at the SMTP level and would thus have ended up in the quarantine rather than being discarded outright. Moreover, the number of false positives was still below average, which means that *McAfee Email Gateway* debuts with a well-deserved VBSpam Gold award.

## McAfee Email and Web Security Appliance

**SC rate (total):** 99.39%

**SC rate (Project Honey Pot corpus):** 99.63%

**SC rate (VB spam corpus):** 95.72%

**FP rate:** 0.24%

**FP rate of total VB mail corpus:** 0.015%

*McAfee*'s *Email and Web Security Appliance* can do a lot more than just filter spam – which perhaps isn't surprising for an appliance made by a well-known anti-virus vendor. We did not look beyond the product's email filtering capacity, but even here the extensive web interface gives the interested system administrator many settings to tinker with.

In the way in which it was set up for this test, there was little need to think about modifying the settings. Not only did the product have one of the highest spam catch rates, it combined that with the lowest false positive rate of all products. As a result of this stellar performance, the *Email and Web Security Appliance* earns a VBSpam Platinum award, the only one of its kind in this test.

## MessageStream

**SC rate (total):** 98.65%

**SC rate (Project Honey Pot corpus):** 99.01%

**SC rate (VB spam corpus):** 93.24%

**FP rate:** 0.78%

**FP rate of total VB mail corpus:** 0.048%

The developers of *MessageStream*, the hosted solution from British company *Giacom*, used some of our feedback from the last test to change the settings in their spam filter and reduce the number of false positives. As a result, *MessageStream*'s false positive rate halved compared to that of the previous test, while barely affecting the spam catch rate. Thus, despite even stricter benchmarks in this test, the product completes a hat-trick of three VBSpam Gold awards in a row.

## Messaging Architects M+Guardian

**SC rate (total):** 98.78%

**SC rate (Project Honey Pot corpus):** 99.20%

**SC rate (VB spam corpus):** 92.41%

**FP rate:** 1.11%

**FP rate of total VB mail corpus:** 0.068%

*M+Guardian*, a hardware appliance developed by Canadian company *Messaging Architects*, achieved VBSpam Platinum awards in both of the preceding tests, giving a good indication of its capabilities. Of course, the spam landscape changes over time and success in the past (or even in the present) does not guarantee success in the future. Indeed, a slightly higher false positive rate and a slightly lower spam catch rate mean that on this occasion the product wins a VBSpam Gold award. A commendable achievement, but to regain a Platinum-level award the product's developers have some work to do.

## Microsoft Forefront Security for Exchange Server v.11

**SC rate (total):** 99.51%

**SC rate (Project Honey Pot corpus):** 99.77%

**SC rate (VB spam corpus):** 95.53%

**FP rate:** 2.00%

**FP rate of total VB mail corpus:** 0.121%

The founder of *Microsoft* once famously predicted that spam would be a thing of the past within two years, but thankfully *Microsoft* is a realistic company and its *Forefront* product is one of many solutions available to protect our inboxes. The product runs on a *Windows* server, where it is an enhancement of *Microsoft Exchange*; the version we tested ran on *Windows Server 2008*. In a normal situation, email that is thought to be ham is sent to the user's inbox and email that is believed to be spam is discarded, with an in-between category stored in quarantine. As is the case with many products, the thresholds for emails ending up in each category can be adjusted by system administrators. In our test, both the discarded mail and the quarantined mail were considered to have been marked as spam.

With a spam catch rate of over 99%, *Forefront* is among the best spam catchers in this test. However, its false positive rate was slightly higher than average – the product misclassified several emails discussing spam as well as a number of newsletters – which means that it debuts in our tests with a VBSpam Silver award.

## SPAMfighter Mail Gateway

**SC rate (total):** 98.03%

**SC rate (Project Honey Pot corpus):** 98.40%

**SC rate (VB spam corpus):** 92.46%

**FP rate:** 3.16%

**FP rate of total VB mail corpus:** 0.189%

*SPAMfighter*'s free product protects the inboxes of many a home-user, but the Danish company also has a server product for businesses. It can run together with *Microsoft Exchange* or *Lotus Domino*, but the version we tested runs as a stand-alone MTA on *Windows Server 2003*. Installation is smooth and the product can be set up easily through a simple web-interface. I was charmed by a graph that showed how many spam emails had been caught and, at $0.04 per email, how much money was thus being saved: while these numbers are nothing but a rough estimate, they show how essential a spam filter is in a business setting.

Unfortunately, the product blocked what it thought was spam a little too eagerly and various legitimate emails were wrongly classified. In particular, newsletters and press releases were blocked and the number of false positives was about twice as high as that of the average product. Hence, despite a decent spam catch rate, the product failed to win an award on its first entry in the test.

## Vircom modusGate

**SC rate (total):** 97.36%

**SC rate (Project Honey Pot corpus):** 97.68%

**SC rate (VB spam corpus):** 92.48%

**FP rate:** 4.42%

**FP rate of total VB mail corpus:** 0.261%

*Vircom*'s *ModusGate*, a product that runs on *Windows Server 2003*, failed to win an award in the previous test and careful investigation by the developers determined that some scripts had not been working as well as they should have been. Fixing these scripts did indeed make a difference, and the product's spam catch rate on this occasion was comparable to that of most other products. At the same time, the product's false positive rate halved, but still this was not sufficiently low for it to win an award. For this, the product would have had to block fewer newsletters as well as emails that discuss spam and/or malware.

## Webroot E-Mail Security SaaS

**SC rate (total):** 99.56%

**SC rate (Project Honey Pot corpus):** 99.81%

**SC rate (VB spam corpus):** 95.75%

**FP rate:** 1.84%

**FP rate of total VB mail corpus:** 0.111%

| | | | | | Total spam | | | Project Honey Pot spam | | | VB corpus | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | True negative | FP | FP rate | FP/total VB corpus | False negative | True positive | SC rate | False negative | True positive | SC rate | False negative | True positive | SC rate |
| BitDefender Security | 1264 | 11 | 0.87% | 0.053% | 3959 | 309780 | 98.74% | 2099 | 292239 | 99.29% | 1860 | 17541 | 90.41% |
| ClamAV | 1270 | 5 | 0.39% | 0.024% | 45808 | 267931 | 85.40% | 39954 | 254384 | 86.43% | 5854 | 13547 | 69.83% |
| Fortinet FortiMail | 1247 | 28 | 2.25% | 0.135% | 3014 | 310725 | 99.04% | 2363 | 291975 | 99.20% | 651 | 18750 | 96.64% |
| Kaspersky Anti-Spam | 1267 | 8 | 0.63% | 0.039% | 5056 | 308683 | 98.39% | 2907 | 291431 | 99.01% | 2149 | 17252 | 88.92% |
| McAfee Email Gateway | 1259 | 16 | 1.27% | 0.077% | 1244 | 312495 | 99.60% | 394 | 293944 | 99.87% | 850 | 18551 | 95.62% |
| McAfee Email & Web Security Appliance | 1272 | 3 | 0.24% | 0.015% | 1927 | 311812 | 99.39% | 1097 | 293241 | 99.63% | 830 | 18571 | 95.72% |
| MessageStream | 1265 | 10 | 0.78% | 0.048% | 4234 | 309505 | 98.65% | 2922 | 291416 | 99.01% | 1312 | 18089 | 93.24% |
| Messaging Architects M+Guardian | 1261 | 14 | 1.11% | 0.068% | 3822 | 309917 | 98.78% | 2350 | 291988 | 99.20% | 1472 | 17929 | 92.41% |
| Microsoft Forefront | 1250 | 25 | 2.00% | 0.121% | 1534 | 312205 | 99.51% | 667 | 293671 | 99.77% | 867 | 18534 | 95.53% |
| SPAMfighter Mail Gateway | 1236 | 39 | 3.16% | 0.189% | 6184 | 307555 | 98.03% | 4722 | 289616 | 98.40% | 1462 | 17939 | 92.46% |
| Vircom modusGate | 1221 | 54 | 4.42% | 0.261% | 8273 | 305466 | 97.36% | 6814 | 287524 | 97.68% | 1459 | 17942 | 92.48% |
| Webroot E-mail Security | 1252 | 23 | 1.84% | 0.111% | 1377 | 312362 | 99.56% | 553 | 293785 | 99.81% | 824 | 18577 | 95.75% |

vb Sept 2009 silver SPAM virusbtn.com

*Webroot*’s hosted solution won a VBSpam Silver award in the previous test, but the developers made it clear they wanted to do better this time.

The results indeed show a significant improvement in performance, both on blocking spam and on letting through ham, and its false positive rate would have been even higher had it not blocked several legitimate emails from the same sender discussing malware. Unfortunately for *Webroot*, other products did better this month too, resulting in stricter benchmarks for this test, and as a result *Webroot* wins another VBSpam Silver award.

## AWARDS

As in the previous test, the levels of the awards earned by products are defined as follows:

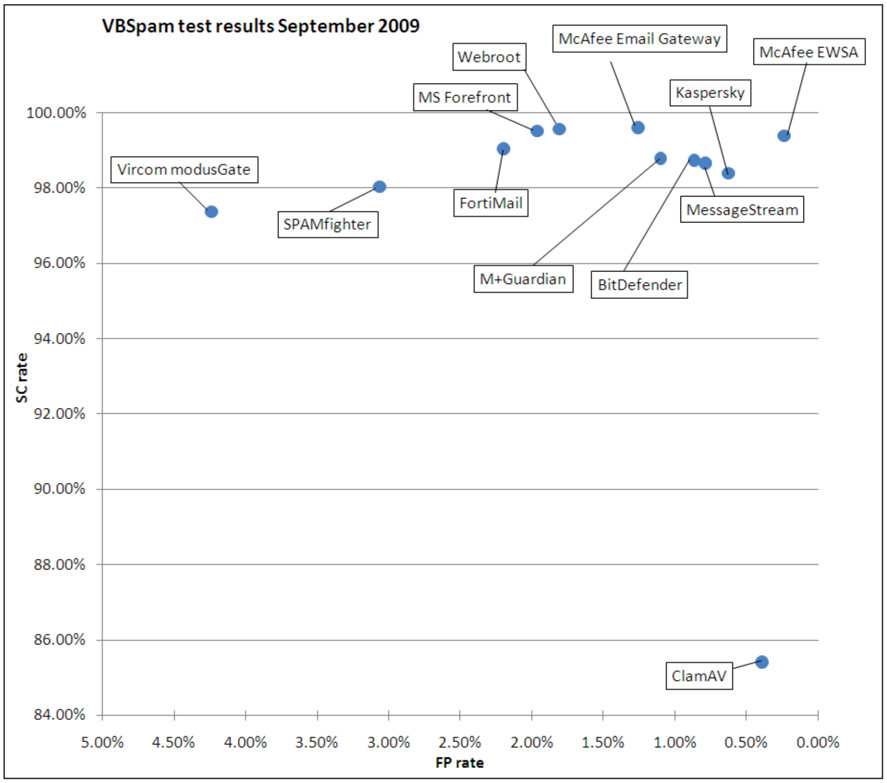
- VBSpam Platinum for products with a total spam catch rate twice as high and a false positive rate twice as low as the average in the test.
- VBSpam Gold for products with a total spam catch rate at least as high and a false positive rate at least as low as the average in the test.
- VBSpam Silver for products whose total spam catch rate and false positive rates are no more than 50% worse than the average in the test.

To avoid the averages being skewed by one or more malperforming products, the scores for any product with a false positive rate of more than 10% and/or a spam catch rate of less than 70% are removed from the computation of the averages; this did not apply to any of the products this month.

This month’s benchmarks are then as follows:

- Platinum: SC 98.85%; FP 0.79%
- Gold: SC 97.70%; FP 1.58%
- Silver: SC 96.56%; FP 2.37%

The table on the previous page shows the scores for all of the products on test. The highlighted columns show the scores used for the benchmark calculations.



## CONCLUSION

With three full anti-spam tests having been completed, a clearer picture is starting to emerge as to which products are the better performers. But, as seen in this test, those that do well cannot rest on their laurels and must work as hard as the others on keeping their products up to date.

We are already working on the next anti-spam test, which we hope will see even more products on the test bench. Just as spam filters need to be constantly updated to fight the latest threats, good anti-spam testers should always look for ways in which their tests can be improved. We welcome comments and suggestions and I hope to open the discussion on what a good anti-spam test entails in a presentation on the subject at VB2009 in Geneva, later this month (23–25 September 2009, see http://www.virusbtn.com/conference/vb2009/ for details).

The next anti-spam comparative review will run in October, with the results published in the November 2009 issue of *Virus Bulletin*. The deadline for product submission is 28 September 2009. Any developers interested in submitting a product are asked to email martijn.grooten@virusbtn.com.

# END NOTES & NEWS

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

**The 7th German Anti-Spam Summit takes place 14–16 September 2009 in Wiesbaden, Germany** (the event language will be English). Participation is free of charge, but regitstration required. For details see http://www.eco.de/veranstaltungen/7dask.htm.

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

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

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

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

**Hack in the Box Security Conference 2009 takes place 5–8 October 2009 in Kuala Lumpur, Malaysia**. Technical training will take place on 5 and 6 October, with conference sessions on 7 and 8 October. For full details see http://conference.hackinthebox.org/.

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

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

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

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

**CSI 2009 takes place 24–30 October 2009 in National Harbour, MD, USA**. For information and online registration see http://www.csiannual.com/.

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

**AVAR2009 will be held 4–6 November 2009 in Kyoto, Japan**. For more details see http://www.aavar.org/avar2009/.

**A step by step masterclass in digital forensics and cybercrime will be run by ICFE on 19 November 2009 in Kuala Lumpur, Malaysia**. The masterclass follows the launch of CSI Malaysia. See http://www. icfe-cg.com/.

**ACSAC 2009 will be held 7–11 December 2009 in Honolulu, Hawaii**. For details see http://www.acsac.org/.

## ADVISORY BOARD

## SUBSCRIPTION RATES

**Subscription price for 1 year (12 issues):**

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

Corporate rates include a licence for intranet publication.

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

**Editorial enquiries, subscription enquiries, orders and payments:**
Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England
Tel: +44 (0)1235 555139 Fax: +44 (0)1865 543153
Email: editorial@virusbtn.com Web: http://www.virusbtn.com/

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.