JANUARY 2010

# virus BULLETIN

Fighting malware and spam

## CONTENTS

## IN THIS ISSUE

### TESTING THE WATER

Since its inception, the Anti-Malware Testing Standards Organization (AMTSO) has outlined its charter, held regular meetings, produced a range of standards documents and continues to work towards raising the overall standard of testing. However, there is still confusion as to what the organization does and stands for. David Harley provides his take on what AMTSO has achieved so far, and what might lie ahead.

**page 9**

### NEWSMAKERS

2009 was filled with security stories involving spam, malware and cybercrime in general. Terry Zink takes a look back at the ten biggest newsmakers.

**page 11**

### VBSPAM CERTIFICATIONS

The first VBSpam comparative review of the new decade saw 15 products on the test bench and some changes to the award criteria. Martijn Grooten has all the details.

**page 23**

vb VERIFIED SPAM virusbtn.com

ISSN 1749-7027

virus BULLETIN

# COMMENT

> *'There are now over 100 times more infected websites on the Internet than three years ago.'*
>
> **Costin Raiu, Kaspersky Lab**

## THE WEB OF DARKNESS

One of the marked trends in the world of cybercrime is the distribution of malware via the World Wide Web. While email worms such as Melissa wreaked havoc in the early years of the last decade, in recent years, the web has become the main distribution point for malware. Malicious programs are hosted on websites; users are then either tricked into running these programs manually, or exploits are used to execute the malware automatically on victim machines.

At *Kaspersky Lab*, we've been monitoring this trend with growing concern. In 2006, we designed and deployed a project called PatroKLes. PatroKLes monitors for infections that are hitting high-profile websites.

Based on everyday browsing, one might think it is rather unlikely for a user to hit an infected website at random. This is not true. It happened to me in 2008 when I was browsing a banking website and got an alert from my security solution. My first thought was that it was a false positive, but careful analysis showed that the website was indeed trojanized with a malicious iframe that led to a website in China that was packed with exploits.

There has been a sharp rise in the number of infected websites from roughly one in every 20,000 or so in 2006 to one in every 150 at the beginning of 2009. The number of infected sites now fluctuates around this number. This may mean that saturation point has been reached – all the websites that can be infected have been infected. However, the number rises and falls as new vulnerabilities and tools are discovered that allow attackers to take over new hosts.

In practice, one infected site in every 150 means that a new computer user will hit an infected website after only a few days of regular browsing. Sometimes it will happen even sooner, as search engine optimization (SEO) is often used to drive traffic to malicious websites.

In 2008, the malware most commonly detected on infected websites was Trojan-Clicker.JS.Agent.h, closely followed by Trojan-Downloader.JS.Iframe.oj. There were two very interesting cases in 2009, the first of which was Net-Worm.JS.Aspxor.a. Although this malware was first back in July 2008, it became far more widespread in 2009. It uses a kit which finds SQL injection vulnerabilities in websites which are then used to insert malicious iframes.

Another very interesting case is Gumblar, named after the Chinese domain that was used as an exploitation point. The 'gumblar' string, visible in the obfuscated JavaScript which is added to websites, is a clear sign that a website has been compromised. The 'gumblar.cn' domain, which was originally used in these attacks, has been taken down, but the bad guys have since switched to new domains.

Once an infection is identified, we attempt to inform the owners. We provide assistance with identifying the malicious code in the page, as well as suggestions on how to secure the server in the future. Unfortunately, we rarely hear back from the owners of these sites. Moreover, there are cases when the owners reply, but do not clean the infection.

Over the past three years, the number of legitimate websites that have been infected with malware has grown at an alarming rate. There are now over 100 times more infected websites on the Internet than three years ago. High-profile, high-traffic websites are a valuable commodity for cybercriminals, as there will be large pools of potential victims that can be infected via such sites. Our experience indicates that the owners of these websites are rarely aware of the infections, and when they are aware, they seldom know how to handle them: in some cases, sites have remained infected for years.

A lot of infections seem to arise through vulnerabilities in old versions of various CMS packages, ranging from PHPBB to *WordPress*. Yet, based on feedback we have received, the majority of website infections occur via stolen account credentials. Web developers or others with login credentials for the website get infected with a password-stealing trojan and the details are used to inject malware into the website. The sad fact is that most of these people are either using an outdated/pirate security suite, or are not running one at all.

In the end, it all comes down to the same basic points: most people are not running security solutions and most people do not really care when they get infected.

# NEWS

## VB2011 DETAILS ANNOUNCED – VIVA ESPAÑA!

*Virus Bulletin* is pleased to announce that VB2011, the 21st International Virus Bulletin Conference, will be held 5–7 October 2011 in Barcelona, Spain. Reserve the dates and start making your travel plans now!

vb 2011 BARCELONA 5-7 October 2011

If you are interested in becoming a sponsor, or require any more information about VB2011, please contact us by emailing conference@virusbtn.com.

## CHILLY CELEBRATIONS FOR KASPERSKY AT SOUTH POLE

There were double celebrations for key members of the *Kaspersky Lab* management team on 31 December as they welcomed in the new year and greeted the Kaspersky Lab Commonwealth Antarctic Expedition at the South Pole. CEO Eugene Kaspersky, Managing Director for the APAC Region Harry Cheung, and Director of *Kaspersky Lab*'s Global Research and Analysis Team Alexander Gostev travelled to the Patriot Hills base camp in Antarctica on 30 December before travelling on to the Pole itself to congratulate the expedition team.

Marking the 60th anniversary of the Commonwealth, the aim of the all-female expedition – which received funding from *Kaspersky Lab* – was to demonstrate the potential for greater intercultural understanding and exchange, while also highlighting the achievements of women across the world. The team – comprising seven women from six Commonwealth countries – braved blizzards, crevasses and temperatures below -30°C as they skied over 900km from the coast of Antarctica to the Geographic South Pole.

Kaspersky, Cheung and Gostev joined the women in planting a Kaspersky Lab Commonwealth Antarctic Expedition flag at the South Pole to mark their achievement. More about the expedition can be found at http://www.kasperskycommonwealthexpedition.com/.

## 'TWAS THE SEASON TO GO PHISHING

There was a significant rise in phishing activity in the run up to Christmas, according to figures released by managed security firm *Network Box*. According to the firm, just over 57% of all web-based threats seen in December were phishing attacks, compared to 28.3% in November – indicating that phishers were poised to take advantage of the seasonal increase in online shopping.

### Prevalence Table – November 2009[1]

| Malware | Type | % |
|---|---|---|
| Conficker/Downadup | Worm | 9.08% |
| Autorun | Worm | 7.63% |
| Virtumonde/Vundo | Trojan | 6.08% |
| OnlineGames | Trojan | 4.54% |
| Virut | Virus | 4.43% |
| Agent | Trojan | 4.40% |
| Adware-misc | Adware | 3.97% |
| Delf | Trojan | 3.85% |
| VB | Worm | 3.79% |
| Heuristic/generic | Misc | 3.75% |
| Alureon | Trojan | 3.63% |
| Heuristic/generic | Trojan | 3.04% |
| FakeAlert/Renos | Rogue AV | 3.00% |
| Suspect packers | Misc | 2.97% |
| Crypt | Trojan | 2.82% |
| Downloader-misc | Trojan | 2.69% |
| Heuristic/generic | Virus/worm | 2.10% |
| Inject | Trojan | 2.08% |
| Small | Trojan | 2.02% |
| Zbot | Trojan | 1.99% |
| Istbar/Swizzor | Trojan | 1.65% |
| FakeAV | Rogue AV | 1.55% |
| WinWebSec | Rogue AV | 1.41% |
| Wimad | Trojan | 1.37% |
| Tanatos | Worm | 1.26% |
| Zlob/Tibs | Trojan | 1.16% |
| BHO/Toolbar-misc | Adware | 1.00% |
| Sality | Virus | 0.91% |
| Cinmus | Adware | 0.88% |
| Mdrop | Trojan | 0.88% |
| Hupigon | Trojan | 0.81% |
| Exploit-misc | Exploit | 0.71% |
| Others[2] |  | 11.52% |
| **Total** |  | **100.00%** |

[1] This month's prevalence figures are compiled from desktop-level detections.

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

# MALWARE ANALYSIS

## IT'S A BIRD, IT'S A PLANE, IT'S FOOPERMAN!

*Peter Ferrie*
Microsoft, USA

It is sometimes said that one man's trash is another man's treasure. In this case, we might say 'one man's data is another man's code'. What we have here is a virus that uses the FPU to magically transform a block of data into executable code, but the secret is in the details of W32/Fooper.

### EXCEPTIONAL BEHAVIOUR

The virus begins by walking the Structured Exception Handler chain to find the topmost handler. At the same time, it registers a new exception handler which points to the host entrypoint. The reason for this will be described below. Once the topmost handler has been found, the virus uses the resulting pointer as the starting location in memory for a search for the MZ and PE headers of kernel32.dll. Once the headers have been found, the virus parses the export table to find the APIs that it needs for infection.

This leads us to the first bug in the code. The problem with the SEH walking method is that in *Windows Vista* and later, the topmost handler no longer points into kernel32.dll but points into ntdll.dll instead. The result is a crash on these platforms, because the virus assumes that the APIs will be found, and falls off the end of a buffer because they do not exist.

### HAPI HAPI, JOY JOY

If the virus finds the PE header for kernel32.dll, it resolves the required APIs. The virus uses hashes instead of names, but the hashes are sorted alphabetically according to the strings they represent. This means that the export table needs to be parsed only once for all of the APIs instead of once for each API, as is common in some other viruses. Each API address is placed on the stack for easy access, but because stacks move downwards in memory, the addresses end up in reverse order in memory. This becomes important later.

After retrieving the API addresses from kernel32.dll, the virus attempts to load 'sfc_os.dll'. If this attempt fails, then the virus attempts to load 'sfc.dll'. If either of these attempts succeeds, then the virus resolves the SfcIsFileProtected() API. The reason the virus attempts to load both DLLs is that the API resolver in the virus code does not support import forwarding. The problem with import forwarding is that while the API name exists in the DLL, the corresponding API address does not. If a resolver is not aware of import forwarding, then it will retrieve the address of a string instead of the address of the code. In the case of the SfcIsFileProtected() API, the API is forwarded in *Windows XP* and later from sfc.dll to sfc_os.dll.

### CULTURAL AWARENESS

The virus retrieves both the ASCII and Unicode versions of the required APIs. One minor detail exists here, which is that because of the way in which the virus uses the APIs, it must swap the address of the CreateFileW() API and the CreateFileMappingA() API on the stack, even though this goes against the alphabetical ordering. The reason for the swap is because the virus requires the ASCII and Unicode versions of any given API to be sequential on the stack. This allows for transparent use of the appropriate API.

Specifically, the virus calls the GetVersion() API to determine the current *Windows* platform, and uses the result to select the appropriate API set (ASCII for *Windows 9x/Me*, and Unicode for *Windows NT* and later). Yes, this virus still supports *Windows 95*! This is because the infection engine used here is the same as the one we first saw the virus author use in 2002. In fact, the only update to the code is the support for Data Execution Prevention (DEP), but setting the executable bit in the section characteristics when appropriate.

The GetVersion() API returns a bit that specifies whether the platform is *Windows 9x*-based (1) or *Windows NT*-based (0). The virus multiplies this value by four, adds the stack pointer value to it, and places the result in a register. Now, whenever the virus wishes to use an API which exists in the two forms, it simply calls the function relative to the register. As such, there is no need ever to check for the platform again. For example, the virus can call '[ebp+CreateFile]', where ebp contains the platform-specific value. If ebp is zero, then the CreateFileW() API is called, and if ebp is four, then the CreateFileA() API is called. This is why the reverse alphabetical order is important for the API addresses on the stack, and why the CreateFileW() and the CreateFileMappingA() API addresses had to be swapped.

### LET'S DO THE TWIST

After finishing with the API trickiness, the virus initializes its Random Number Generator (RNG). The RNG is interesting in itself, since it is neither the usual GetTickCount()-based randomizer, nor the Knuth-inspired algorithm. Instead, the virus uses a complex RNG known as the 'Mersenne Twister', named after the kind of prime number at its heart. The virus author has used this RNG in each of his viruses for which he requires a source of random numbers. Curiously, only one virus created by a different virus author has ever used the same RNG.

The virus then searches for files in the current directory and all subdirectories, using a linked list instead of a recursive function. This is important from the point of view of the virus author, because the virus infects DLLs, whose stack size can be very small. The virus avoids any directory that begins with a '.'. This is intended to skip the '.' and '..' directories, but in *Windows NT* and later, directories can legitimately begin with this character if other characters follow. As a result, those directories will also be skipped.

## FILTRATION SYSTEM

Files are examined for their potential to be infected, regardless of their suffix, and will be infected if they pass a very strict set of filters. The first of these filters is the support for the System File Checker that exists in *Windows 98/Me*, and *Windows 2000* and later. Since the directory searching on the *Windows 9x/Me* platforms uses ANSI paths, and since the SfcIsFileProtected() API requires a Unicode path, the virus converts the path from ANSI to Unicode, if appropriate, before calling the API.

The remaining filters include the condition that the file being examined must be a *Windows* Portable Executable file, a character mode or GUI application for the *Intel* 386+ CPU, that the file must have no digital certificates, and that it must have no bytes outside of the image. Additionally, if the file is a DLL, then it must have an entrypoint.

## TOUCH AND GO

When a file is found that meets the infection criteria, it will be infected. The virus resizes the file by a random amount in the range of 4 to 6KB in addition to the size of the virus. This data will exist outside of the image, and serves as the infection marker.

If relocation data is present at the end of the file, the virus will move the data to a larger offset in the file and place its own code in the gap that has been created. If no relocation data is present at the end of the file, the virus code will be placed there. The virus checks for the presence of relocation data by checking a flag in the PE header. However, this method is unreliable because *Windows* ignores this flag, and relies instead on the base relocation table data directory entry.

The virus increases the physical size of the last section by the size of the virus code, then aligns the result. If the virtual size of the last section is less than its new physical size, then the virus sets the virtual size to be equal to the physical size, and increases and aligns the size of the image to compensate for the change. The virus also changes the attributes of the last section to include the executable and writable bits. The executable bit is set in order to allow the program to run if DEP is enabled, and the writable bit is set because the RNG writes some data into variables within the virus body.

The virus alters the host entrypoint to point to the last section, and changes the original entrypoint to a virtual address prior to storing the value within the virus body. This act will prevent the host from executing later if the host is built to take advantage of Address Space Layout Randomization (ASLR). However, it does not prevent the virus from infecting files first. The lack of ASLR support might be considered a bug unless we remember that ASLR was not introduced until *Windows Vista*, which, as noted above, the virus does not support. What is strange, though, is that changing the entrypoint in this way affects DLLs in the same way. Thus, if an infected DLL is relocated because of an address conflict, then it, too, will fail to run. This is despite the fact that in other viruses the virus author has demonstrated the ability to infect DLLs correctly, by calculating the virtual address of the entrypoint dynamically. Since this method is equally applicable to ASLR-aware files, the same method could have been used in both cases.

## ROOT BEER FLOATS

At this point, the virus generates a new decryptor for the virus body. It begins by choosing a random CPU register (with the exception of the ESP register), whose purpose depends on whether or not the decryptor was using it previously. In the .A and .C variants, the virus examines the register initialization code in the decryptor (the .B variant has no such section) and makes a note of which registers are in use. At the same time, it checks whether the chosen register is already in use (there is one register which is not used in the register initialization code – this is used as the base register for the memory accesses). This is a very elegant routine.

The decryptor contains three sections (in the .A and .C variants; two in the .B variant) where the chosen register might have been used: it might have been used in the register initialization code, it might have been used as the base register for the memory accesses, and it might have been used as the counter register. In any case, if the chosen register is used already, then the virus replaces it with the unused register. The virus always replaces the scale register for the memory accesses with the chosen register. The construction of the decryptor then proceeds differently for each of the variants.

## .A DECRYPTOR

The .A variant replaces any references to the chosen register in the arithmetic instructions with the unused register. It swaps the register initialization lines randomly. The decryptor is a set of simple arithmetic operations, but it uses all of the registers, and it is sufficiently complex that it cannot be x-rayed.

The virus generates an FPU ‘fsave’ instruction using the unused register, and assigns random initial values to all of the registers except for the counter register. The virus also generates a series of FPU ‘fld’ (Floating-point LoaD) instructions using the unused register: one fld instruction for each ten bytes of the decryptor, for a total of 80 bytes. The offset for the fld instructions is a random multiple of ten within that 80-byte block, but since the FPU registers (known as ‘stn’, where ‘n’ is the slot number, from zero to seven) exist on a stack, the lines are loaded in a fixed order. That is, the first ten bytes that are loaded correspond to the last ten bytes in the decryptor; the second ten bytes that are loaded correspond to the second-to-last ten bytes in the decryptor, and so on.

## .B DECRYPTOR

The .B variant uses a decryptor that is a set of simple arithmetic operations that use immediate values, but it is sufficiently complex that it cannot be x-rayed.

The virus generates an FPU ‘fsave’ instruction using the unused register, and assigns random values to all of the arithmetic operations. The virus also generates a series of MMX ‘movq’ (MOVe Quadword) instructions using the unused register: one movq instruction for each eight bytes of the decryptor, for a total of 64 bytes. The offset for the movq instructions is a random multiple of eight within that 64-byte block, and since the MMX registers (known as ‘mmn’ – strangely, not ‘mmxn’ – where ‘n’ is the register number, from zero to seven) can be assigned explicitly, the order of the loads is also random. That is, the first register that is loaded might be any one of the eight MMX registers, however the bytes that the register holds will always correspond to the same eight bytes in the decryptor.

The decryptor of the .B variant is somewhat weaker than the decryptor in either the .A or the .C variant, partly because of the way in which the MMX registers are used within the CPU. Specifically, the MMX registers share the same slots within the FPU as the standard FPU registers. However, since the FPU registers are each ten bytes long, while the MMX registers are only eight bytes long, the FPU automatically fills the last two bytes of the slot with the value 0xFF. Because of the way in which the decryptor works (see below), these 0xFF bytes must be skipped. The virus achieves this by further shortening the contents of the slots (hence the simple arithmetic instructions accepting only immediate values), and placing a jump instruction at the end of the slot.

The virus author could have used an instruction that would incorporate the 0xFF bytes, which would have avoided the jump, and thus would have increased the usable size of the slots by one byte. There are two candidate values that would serve the purpose: 0x80 and 0x82. Both values decode to the same instruction when followed by 0xFF 0xFF: CMP BH, FF. It seems likely that the virus author knew this, but given the style of the decryptor, the additional bytes might not have seemed sufficient to insert any further instructions (the result of the compare could have been used, for example, but the decryptor would look quite different).

## .C DECRYPTOR

The .C variant replaces any references to the chosen register in the arithmetic instructions with the unused register. It swaps the register initialization lines randomly. The decryptor is a set of simple arithmetic operations, but it uses all of the registers, and it is sufficiently complex that it cannot be x-rayed.

The virus generates an FPU ‘fxsave’ instruction using the unused register, and assigns random initial values to all of the registers except for the counter register. The virus also generates a series of SSE ‘movdqu’ (MOVe Double-Quadword Unaligned) instructions using the unused register: one movdqu instruction for each 16(!) bytes of the decryptor, for a total of 128 bytes(!). The offset for the movdqu instructions is a random multiple of 16 within that 128-byte block, and since the XMM registers (known as ‘xmmn’, where ‘n’ is the slot number, from zero to seven) can be assigned explicitly, the order of the loads is also random. That is, the first register that is loaded might be any one of the eight XMM registers, however the bytes that the register holds will always correspond to the same 16 bytes in the decryptor. Further, since the XMM registers occupy their own space within the FPU, the entire slot is available for use, and the virus takes advantage of this. The virus places a jump instruction after the last movdqu instruction in order to reach the fxsave instruction.

## FSAVE THE WORLD

The virus uses the fsave instruction (or the fxsave instruction in the .C variant) in order to do something special with the loaded registers.

Prior to the execution of the f[x]save instruction, the registers exist essentially in isolation. While the registers can be manipulated individually, they exist as separate data items. However, when the f[x]save instruction is executed, the registers are stored in a particular order to the memory location that is specified by the instruction. The order is first to last (st0 or [x]mm0, then st1 or [x]mm1, then ... st7 or [x]mm7). There is no padding between the stored registers, allowing them to form a block of executable code if the contents are valid instructions. That is the case here. However, the virus goes further, by specifying an address for the f[x]save instruction such that the next instruction to execute comes from the first of the stored registers, and execution proceeds from there. This act is self-modifying in an interesting way, since the f[x]save instruction is overwritten by the data that the f[x]save instruction causes to be stored.

## APPENDICITIS

After constructing the decryptor, the virus will append its body and encrypt it with a routine that performs the reverse actions of the decryptor.

Once the infection is complete, the virus calculates a new file checksum, if one existed previously, before continuing to search for more files.

Once the file searching has finished, the virus will allow the host code to execute by forcing an exception to occur. This technique appears a number of times in the virus code and is an elegant way to reduce the code size, in addition to functioning as an effective anti-debugging method.

Since the virus has protected itself against errors by installing a Structured Exception Handler, the simulation of an error condition results in the execution of a common block of code to exit a routine. This avoids the need for separate handlers for successful and unsuccessful code completion.

## CONCLUSION

Causing the FPU to reorder some data, such that it then becomes meaningful in a different context, is an interesting idea. It’s a bit like a word puzzle, where the letters have been arranged randomly. Who knew that the FPU could solve anagrams?

# CALL FOR PAPERS

## VB2010 VANCOUVER

*Virus Bulletin* is seeking submissions from those wishing to present papers at VB2010, which will take place 29 September to 1 October 2010 at the Westin Bayshore hotel, Vancouver, Canada.

The conference will include a programme of 30-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.

A list of topics suggested by the attendees of VB2009 can be found at http://www.virusbtn.com/conference/vb2010/call/. However, please note that this list is not exhaustive, and the selection committee will consider papers on these and any other anti-malware and anti-spam related subjects.

## SUBMITTING A PROPOSAL

The deadline for submission of proposals is **Friday 5 March 2010**. Abstracts should be submitted via our online abstract submission system. You will need to include:

- An abstract of approximately 200 words outlining the proposed paper and including five key points that you intend the paper to cover.
- Full contact details.
- An indication of whether the paper is intended for the technical or corporate stream.

The abstract submission form can be found at http://www.virusbtn.com/conference/abstracts/.

One presenter per selected paper will be offered a complimentary conference registration, while co-authors will be offered registration at a 50% reduced rate (up to a maximum of two co-authors). *VB* regrets that it is not able to assist with speakers’ travel and accommodation costs.

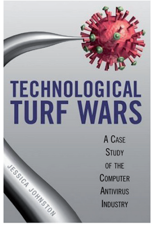
Authors are advised that, should their paper be selected for the conference programme, they will be expected to provide a full paper for inclusion in the VB2010 Conference Proceedings as well as a 30-minute presentation at VB2010. The deadline for submission of the completed papers will be Monday 7 June 2010, and potential speakers must be available to present their papers in Vancouver between 29 September and 1 October 2010.

Any queries should be addressed to editor@virusbtn.com.

# BOOK REVIEW

## SOCIAL SCIENCE MEETS COMPUTER SCIENCE

*Paul Baccas*
Sophos, UK



**Title:** Technological Turf Wars: A Case Study of the Computer Antivirus Industry

**Author:** Jessica Johnston

**Publisher:** Temple University Press

**ISBN:** 1-59213-882-9

This book is a socio-politico-economic analysis of the anti-virus industry based on interviews with associated parties (researchers, vendors and corporate end-users) through the first six years of the 21st century.

## UNDER THE COVER

The book is divided into an introduction and six chapters.

### Chapter 1: Naming the threat

Despite its title, the first chapter is not about the turgid arguments that rumble on over malware nomenclature, but rather about why and from where the term ‘virus’ came into our lexicon and why the metaphor of virus endures. It then moves on to describe the nature of the threat, how it has changed and how our defences against it have been commodified.

### Chapter 2: Security transformations

In this part of the book the standard security triple of confidentiality, integrity and availability is used to underpin the argument that the terms ‘security’ and ‘threat’ form a loop because they are defined, in the minds of the interviewees, with reference to each other.

The author goes on to address the role of governments and how their interests in the field have changed, particularly since 9/11. Finally, there is a case study looking at how spam has changed from a mere annoyance in the early ’90s to a real security threat, and how detection has been added to the standard suite of security products. This transformation is mirrored elsewhere, specifically the move in the mid-90s to add trojan detection to anti-virus products.

### Chapter 3: Trust, networks, and the transformation of organizational power

The way in which people and groups communicate and interact is the meat and drink of this book, with chapters 3 and 4 looking at grouping within the industry. The majority of researchers interviewed in the book were CARO members, so it is not surprising that chapter 3 is dominated by an analysis of CARO. Describing what the author calls ‘the mythic past’ through to the present, CARO is used as a standard against which to compare all other AV industry interactions. Detailing the contretemps that arose among vendors over REVS and how, as a result of that, industry interactions changed, this chapter touches upon the exclusions inherited by CARO.

### Chapter 4: IT corporate customers as end-users

In this chapter the author discusses how corporate users felt that they needed a more coherent and louder voice in the industry. The corporate end-users’ perception was that they were the ones working in the trenches and that the vendors/researchers could make use of their expertise. This led to the formation of AVIEN (the Anti-Virus Information Exchange Network), exclusively for end-users, and AVIEWS (the Anti-Virus Information Early Warning System), which included vendors and researchers. However, researchers soon realized that the data received via AVIEWS was also being provided via other avenues, and this diluted the usefulness of the resource from their point of view.

### Chapter 5: Marketing service

Chapter 1 talks about the commodification of anti-virus. Commodities are purchased through word of mouth, advertisement, or a mixture of the two. In short, commodities are marketed, and the anti-virus industry has had a chequered history with rogue and not so rogue marketeers/press agents. Here the author discusses the dichotomy of the researchers’ desire for truth and accuracy and the marketeers’ desire to cultivate press links and drive sales.

### Chapter 6: Situated exclusions and reinforced power

Race and gender studies are the bread and butter of social science research. The AV industry, at least as far as conference attendees go, is a fairly homogeneous group (white, male, middle-aged, degree-educated, North American or Eastern/Northern European) and provides rich picking for an analysis of race and gender.

## A GOOD READ?

The book is an interesting, if stylistically convoluted read. My main problem with it is that six years is a long period to cover in an industry such as this. I felt that several books covering shorter periods or a longer book split into sections (e.g. 1970–1990, 1990–2000, 2000–2005 etc.) would have presented the subject matter in a more coherent fashion. As it stands, there is certainly scope to add to this work – and it was fun attempting to guess the identities of the anonymous interviewees.

# SPOTLIGHT

## AMTSOLUTELY FABULOUS

*David Harley*
ESET, USA

*The Anti-Malware Testing Standards Organization (AMTSO) was formed following a 2007 CARO workshop aimed at discussing 'best practice' and common flaws in anti-virus testing methods. A selection of participants from the workshop decided to join forces and in January 2009 more than 40 security software experts and anti-malware testers from around the world met to formalize the charter of the Organization. Since its inception, AMTSO has outlined its charter, held regular meetings, produced a range of standards documents and continues to work towards raising the overall standard of testing. However, there is still confusion as to what the organization does and stands for. David Harley provides his take on what AMTSO has achieved so far, and what might lie ahead.*

Does Figure 1 represent your perception of the Anti-Malware Testing Standards Organization (AMTSO) [1]? Many people with an interest in anti-malware testing are now watching the organization with keen interest, but in a state of some confusion.

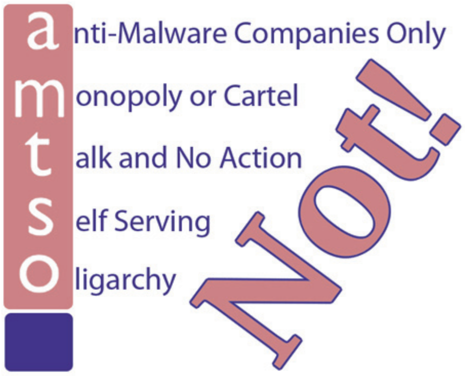


*Figure 1: AMTSO: the view from the T-shirt.*

### WHINE AND DINE

Some see AMTSO as a group of anti-virus vendors meeting to whine about how awful testing is; others have a clearer view of who is participating, but think of it as a testing organization in its own right – or expect it to transform into a full-blown standards development organization like ISO, or a full-time compliance monitoring agency. I don't presume to speak for AMTSO, but let me give you my own views on what has been achieved so far and what might lie ahead.

AMTSO represents a productive (and open – more members are always welcome) alliance between the anti-malware industry, mainstream testers and publishers, all of whom have had to invest much time and effort into adjusting to new threat trends. Vendors have done so by working on enhanced approaches to detection; testers, reviewers and publishers have done so by developing realistic criteria and methodologies for evaluating and comparing re-engineered technologies. The organization also benefits from the input of an advisory board [2] consisting of people who, despite their knowledge and experience of the anti-malware industry, have no vested interest in promoting it. AMTSO has always felt that the presence of this group is essential to the purpose and functioning of the organization, defending the interests of the community at large against AMTSO's becoming a clique of self-interested vendors.

### ENLIGHTENED SELF INTEREST

Of course, it is inevitable that vendors will be (self-)'interested', but most believe that they have as much to gain from a higher standard of testing across the board as anyone else [3]. In any case, an organization without the accumulated experience of the anti-malware research community would be hard-pressed to maintain credibility, as it would lack input from the people who know the most about the technology under test.

Central to AMTSO's purpose is the recognition among the security community (including mainstream professional testers) that traditional static testing (a.k.a. throwing every available malicious program at an on-demand scanner to see how many it detects) is no longer a fully effective measure of a product's capabilities – if it ever was.

### GLUT... GIVE ME GLUT AND NOTHING BUT...

In a threat landscape where tens of thousands of new samples [4] – most of which are non-viral (that is, trojans of some sort rather than self-replicating malware) – are seen on a daily basis, the leisurely testing methodologies of yesteryear are of limited use. However, more dynamic testing methods that reflect the complexities of a constantly shifting threatscape (and increasingly, cloud-based technologies) are themselves complex and resource-intensive. Even testers who are aware of the need to move towards dynamic testing are often deterred by the resource implications and the technical difficulties.

## INFORMATIONAL LOAD BALANCING

Clearly, there is a need for information sharing and discussion if the testing industry is to progress. AMTSO offers a forum for expert discussion as the testing industry moves towards more relevant testing methods, with input from the anti-malware sector of the security industry as well as from some of the most experienced mainstream testers.

Perhaps AMTSO's core function is to raise the overall standard of testing, through discussion (and there's been plenty of that!), by developing standards and documenting good practice, by encouraging the provision of tools and other resources, by providing analysis and review of tests, and, perhaps most significantly, through education. One way to raise awareness is by providing sound information from authoritative sources. To this end, AMTSO members have put together a repository of documents [5]. A particularly significant item is a testing principles document [6] that provides a high-level view of the basic rules of sound anti-malware testing. These are not tablets of stone, but documents that may be amended over time to adapt to changing circumstances and technologies.

## TESTING, TESTING, 1, 2, 3

Software testing has never been a particularly easy discipline, and security product testing poses particular challenges, since the technical aspects of attacks and countermeasures are not always well understood (or, indeed, communicated by the industry). For a long time, it seemed as though the anti-virus industry was eager to complain about bad tests, but unresponsive when asked 'so how would you *like* us to do it?' [7].

AMTSO has made moves to overturn this perception by providing copious documentation on basic principles and on specific testing issues. These don't provide the wannabe tester with everything he could ever need in order to become a credible tester, but they do at least provide a basis for communication between AMTSO and testers (and other interested parties) currently outside the organization. It is no longer possible for anyone to claim that there is no useful, impartial information on testing to be found outside the charmed circle of vendors and mainstream testers. Even testers outside the relatively close-knit security community can draw on this resource to enhance the value of their testing, and in turn, their audiences can gain better understanding of how testing works.

## NEGATIVE POLARITY

Inevitably, some people have anticipated a more negative and authoritarian approach from AMTSO, and in some cases there has been disappointment that it has not been more ready to wield the stick than the carrot. There seems to be a common perception that AMTSO either has or *should have* set itself up as the AV industry's police force, to monitor and enforce good testing methodologies – after all, the name of the organization includes the word 'standards' (not 'guidelines', 'suggestions', or even 'good practice'). However, AMTSO is *not* the AV industry, and though that industry's interests are certainly represented, they are secondary to the interests of the community at large. While there's a place for both the carrot and the stick, AMTSO's best course right now is to establish dialogue and consensus across the community, rather than to be the ultimate authority on good and evil in testing.

amtso self-assessed

amtso member

amtso supporter

## CONCLUSION

Anti-malware testing is not as easy as most people think it is (in fact *all* product testing is harder than most people think): it takes skill, knowledge, care and significant resources to perform a test that offers good guidance on a product's capabilities rather than subjective opinion based on misinterpretation and unrealistic assumptions. All too often, a single data set is used by different groups to support very different conclusions.

Perhaps the best services AMTSO offers to the community in the short term can be summarized as follows:

- Testers are not (only or primarily) accountable to the security industry, but to their audiences, who expect (sometimes naively) to be guided by objective, informed evaluation, not to be misled by personal prejudice. Of course, it's not only testers who need to know this, but also the public, who are often prepared to believe anything anyone says about a product as long as that person claims to have no connection with the industry [8].
- AMTSO has already made a difference simply by providing a platform for debate and a public resource of which testers can make good use, but it also has the potential to provide a better yardstick for the evaluation of tests. While it's not yet clear exactly what AMTSO compliance is, let alone how to measure it, many groups seem to want it, either so that they can use it as a metric, or so that they can demonstrate compliance.
- AMTSO does not certify tests or testers, and is not really in a position to adopt a compliance enforcement

> role until there are formal standards against which to apply certification. However, AMTSO's good practice guidelines describe the principles that a sound tester would normally be expected to follow. These principles form a viable basis for a number of approaches to improving accountability: AMTSO's Review Analysis Board [9] is starting to use them as a measure of the accuracy of a test or review, while a self-assessment process has also been proposed. This would enable a testing group to demonstrate its intent to comply with AMTSO guidelines and willingness to undergo some form of verification. In the longer term, there are certainly arguments for a formal certification process: hopefully AMTSO would participate or perhaps even initiate such a process.

By stressing the constructive aspects of AMTSO's mission to improve specific methodologies (notably, hybrid and dynamic testing) and community awareness of the problems and of the solutions, the organization hopes to encourage all interested parties to follow and participate in the debate. After all, it's to be expected that as AMTSO gains traction, it will be harder for testers and reviewers to claim credibility without demonstrating awareness of the organization's aims and making a verifiable effort to follow them. Now is the time for other players wishing to be heard in the debate to raise their hands and voices.

## REFERENCES

[1] http://www.amtso.org/.

[2] http://www.amtso.org/amtso---boards---advisory-board.html.

[3] Harley, D.; Lee, A. Who will test the testers? Proceedings of the 18th Virus Bulletin International Conference, 2008.

[4] Harley, D.; Bureau, P-M. A Dose By Any Other Name. Proceedings of the 18th Virus Bulletin International Conference, 2008.

[5] http://www.amtso.org/documents.html.

[6] http://www.amtso.org/en/amtso---download---amtso-fundamental-principles-of-testing.html.

[7] Harley, D. The Game of the Name: Malware Naming, Shape Shifters and Sympathetic Magic. 3rd Cybercrime Forensics Education & Training Conference, 2009.

[8] Harley, D. I'm OK, You're Not OK. Virus Bulletin, November 2006, p.6. http://www.virusbtn.com/pdf/magazine/2006/200611.pdf.

[9] http://www.amtso.org/pr-090519.html.

# FEATURE

## THE TOP TEN SPAM, MALWARE AND E-SECURITY STORIES OF 2009

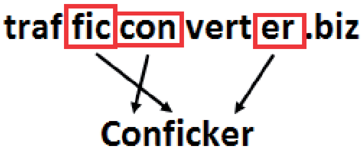
*Terry Zink*
Microsoft, USA

2009 was a year filled with security stories involving spam, malware and cybercrime in general. It was a jam-packed year, so let's take a look at the ten biggest newsmakers.

### 1. COME TOGETHER, RIGHT NOW

Conficker is a piece of malware that first appeared in late 2008, but its story didn't really heat up until 2009. The story is noteworthy not because of the impact of Conficker, which in itself was large; instead, the story is important because of the way in which the industry responded to the problem.

Conficker is a worm that uses flaws in the *Windows* operating system to add its hosts to a botnet and execute remote instructions. A patch was released for the *Windows* vulnerability in October 2008, but Conficker appeared in November 2008 and began to exploit the vulnerability. The worm used a number of advanced techniques and became the most prevalent piece of malware detected in 2009. Conficker is thought to have been named by rearranging the letters of trafficconverter.biz, which was a site used by early versions of the worm to download updates:



In spite of the advanced techniques used by Conficker, security researchers managed to discover an update mechanism through which infected computers could download additional instructions. The researchers reverse engineered an algorithm that would generate 500 new domain names per day, which the malware would then use to connect to its command and control centres. The researchers began a process of manually registering each of the domain names in advance, so that any attempt by Conficker to contact them would fail. However, it soon became clear that registering so many domain names would be a very expensive undertaking.

Thus was born the Conficker Working Group (CWG). In January 2009, representatives from various security companies, along with the anti-botnet Shadowserver Foundation, got together and designed a strategy to counteract Conficker. One month later, the group had a plan to register as many domains as possible and assign them to a sinkhole – a server designed to capture and analyse malware traffic. ISPs were able to use this data to analyse traffic in order to identify infected systems. At about the same time, ICANN[1] invited representatives from the group to present their findings to the ICANN board and expressed that it would help where it could. The Conficker authors still managed to register some domains, but members of the CWG had computed a year's worth of Conficker domains in order to direct them to the sinkholes. It looked like Conficker may be thwarted.

However, in March a new Conficker variant appeared: Conficker.D was scheduled to register as many as 50,000 new domain names *per day* across more than 100 top-level domains (TLDs), starting on 1 April 2009. This looked like too big a task to tackle, but the CWG managed it. They secured the cooperation of all of the owners of the TLDs to register or block the new domains in question. April 1 came and went, but by that point the domains had already been blocked.

The fight against Conficker is not yet over. Estimates of the number of computers infected by Conficker are in the region of five to ten million. And once in a while, legitimate domains collide with Conficker domains and are blocked. New variants of Conficker may yet be released, and no one really knows for sure what Conficker's purpose is, other than perhaps acting as a conduit to serve other botnets. However, the rapid, effective response by industry to the threat that this malware posed serves as an example of what can be accomplished when interested parties work together. It *is* possible to stop malware from spiralling out of control.

## 2. WHY CAN'T I TWEET TODAY?

In August, users of the *Twitter* social-networking site discovered that their favourite 140-character messaging service was offline. 'What's going on?' they asked. 'People need to know what I had for breakfast!'

It turned out that a co-ordinated Distributed Denial of Service (DDoS) attack had been launched against a number of social networking sites including *Twitter*, *Facebook*, *LiveJournal*, *YouTube* and *Blogger*. But whereas the other sites were able to repel the attacks, *Twitter* was not.

On further analysis the disruption appeared to be the result of a targeted attack against one particular blogger by the name of Cyxymu, or Сухуми (Sukhumi), which is the capital of the Georgian breakaway region of Abkhazia.

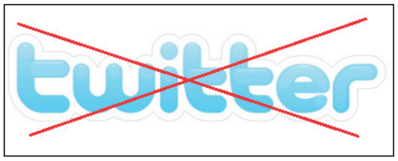


*Figure 1: Twitter didn't cope with the DDoS attacks as well as its social networking cousins.*

Cyxymu had accounts on each of the aforementioned services and posted messages on the topic of civilian suffering during the war – he was decidedly anti-Russian. In August, a large spam run containing links to Cyxymu's various social networking pages coincided with the DDoS attacks on the sites. It resembled an attempt to make it look as if Cyxymu had sent out a huge spam run to drive visitors to his pages to support his cause, and that his campaign had backfired when all of the new traffic shut down the sites, and then backfired again when *Facebook*, *Twitter*, *et al.* all found out he was spamming to get people to come to his pages. Perhaps the attackers thought that this unethical behaviour (spamming) would get him suspended. It almost sounds like the plot of a Hollywood movie.

In reality, while *Twitter* did go offline for a time, the other sites did not. In fact, they became adamant that they would not shut down Cyxymu as he was entitled to his rights to free speech. Cyxymu, so it is thought, was innocent and was the target of a cyber attack in an attempt to silence him. In the United States, an attempt to shut somebody down for exercising their rights to free speech is almost always met with contempt.

This was not the first time that politics had been mixed with cyber riots. In 2007, the Estonian government was hit with cyber attacks that shut down its infrastructure after it had attempted to remove a Russian World War II monument from downtown Talinn. In 2008, during the Russian/Georgian war, DDoS attacks took Georgian and Azerbaijani sites offline.

In the 21st century, politics and cyber attacks have become increasingly intertwined. And August's *Twitter* attack wouldn't mark the last time that hacktivism would make a splash on the political scene in 2009.

## 3. THE SHUTDOWNS CONTINUE

One of the top stories of 2008 was when Californian ISP *McColo* was taken offline after a story was published in *The Washington Post* describing how it acted as a command-and-control centre for botnets that send spam and host fast-flux or server malware. Almost immediately after the shutdown, global spam levels plummeted. This showed that, given enough motivation, there was a mechanism to fight

[1] ICANN is the Internet Corporation for Assigned Names and Numbers, the group responsible for managing the globe's IP addresses and Domain Name System (DNS).

back against spam if the spotlight was shone on the public sector.

*Figure 2: Pulling the plug on McColo.*

2009 also saw its share of ISPs taken offline. In June, the US Federal Trade Commission filed a motion of complaint to have *Pricewert LLC*, an American ISP, taken down. In August, Latvian ISP *Real Host*, which was responsible for numerous botnet command-and-control centres, was similarly disconnected. But the major talking point of these two disablements was not how much spam volumes decreased, but how little impact there was on the global spam volume.

*Pricewert*'s removal saw spam levels drop slightly – less than 10% – but within days everything had returned to normal. The takedown of the Latvian ISP registered no discernable change in spam levels at all. Indeed, what we learn from the *Real Host* outage is that spammers learned from the *McColo* outage: they no longer place all of their eggs in one basket. They have adapted and evolved so that they are no longer solely reliant on a single point of failure, and seem now to be building some redundancy into their networks. The short-lived elation of seeing *McColo* taken down has worn off and we are left with the grim reality that spammers are coming back more resilient than before.

## 4. THE LITTLE EMPIRE STRIKES BACK!

In November, the small security company *FireEye* was able to disable a botnet that at one point was responsible for perhaps a third of the world's spam. Security researchers from the company analysed the workings of the huge botnet known as Mega-D (or Ozdok) and managed to infiltrate its command-and-control structure. They were able to send a new set of instructions to all of the zombie hoards that make up the Mega-D botnet. After doing this, spam from Mega-D slowed to a crawl. *FireEye* had succeeded.

What is extraordinary about this is that a relatively small company was able to take down a very large spam operation that has a vested interest in keeping its infrastructure up and running. Mega-D had built in redundancy to guard against this very thing by reserving a long list of domain names for its command-and-control centres, as well as using hard-coded DNS servers. It also had software to dynamically generate new domain names on the fly [1].

It remains to be seen whether Mega-D will remain offline for long. As we saw with *Pricewert* and *Real Host*, their takedown had merely short-term effects on the spam problem. When the anti-spam community comes up with a technique to disable the spamming infrastructure, spammers react by building a better one. The battle continues.

## 5. COLONEL MUSTARD IN THE BALLROOM WITH THE CANDLESTICK…?

As Americans celebrated the 4th of July weekend with backyard barbecues and fireworks, various government employees had to put their hamburgers and potato salads to one side.

That weekend, a large-scale DDoS attack hit the Federal Trade Commission, the US Department of Transportation and the US Treasury. The US Secret Service, Department of Homeland Security and the State Department were also hit. So were several government websites in South Korea. The attacks were particularly severe, taking up 40GB of data per second – much larger than a typical attack.

As the attacks began to wane [2], various trojans that had infected the PCs used in the attacks started to overwrite data in the hard drives with a message that read 'memories of independence day', attempting to write over every physical drive of the compromised systems. Thus, the trojans had a self-destruct feature that was designed to inflict maximal damage.

So who was behind these attacks? Shortly after they occurred, South Korean officials blamed North Korea, or at the very least, pro-Pyongyang forces. North Korea, of course, denied involvement. What clouds the issue is that the attacks need not have been government sponsored. They could equally have been the work of pranksters or industrial spies. Were the North Koreans responsible for the attacks? Maybe they were, maybe they weren't. Certainly, the IPs used in the attacks were located in the Far East, but that doesn't mean that the people responsible for controlling them were.

What is more worrying is the fact that the DDoS attacks actually succeeded in disabling the government websites. These types of attacks are things that private ISPs see every day and repel every day. Yet, the governments had a single attack against them and just like that, their sites were taken down. This illustrates the current vulnerability of governments in the cyber arena – they can't defend against the sort of attacks that industry has been handling for years.

Perhaps the US and South Korean governments need to join up with *Twitter* and form a support group.

## 6. THE LONG ARM OF THE LAW

2009 saw some pretty heavy hitting in the legal arena in the spam world. In June, 'spam king' Alan Ralsky pleaded guilty to a stock fraud case where he pumped up Chinese penny stocks.

In 2004 and 2005, Ralsky, along with a small group of other people, conspired to manipulate stocks using spam messages to 'pump-and-dump' their value. In other words, they would pick a stock, buy shares in it, send out a huge number of spam messages claiming that the stock was poised to go through the roof, and then wait for the rest of the world to buy it in droves. This buying surge would send the price up, at which point Ralsky and his group would sell their shares and collect the profit. Once the buying surge was over, the stocks would return to their previous value. The stocks were typically low-priced 'pink-sheet' stocks for US companies owned by people in Hong Kong and China.

Ralsky used all sorts of spamming techniques to get his message across, including the falsification of email headers and extensive use of botnets.

Ralsky was one of the world's most prolific spammers. He reportedly once admitted to sending more than 70 million spam messages per day. At 70 million per day, even a hit rate of 0.01% equates to 7,000 actions being taken. Eventually, though, his crimes caught up with him and he pleaded guilty to wire fraud, mail fraud, money laundering and violating the United States' CAN-SPAM Act. In exchange for lighter sentencing, he agreed to provide assistance in the prosecution of other spammers.

Yet Ralsky did not get off lightly. In November 2009, he was fined $250,000 and sentenced to four years in jail. However, many anti-spam advocates doubt that this is enough.

Across the ocean, another spammer was also hit with a huge fine. In November, the US Federal Trade Commission (FTC) fined Lance Atkinson $15 million. Atkinson is believed to have been behind the spam affiliate Affking, the folks who brought you such delights as the Canadian Pharmacy's cheap drugs and Herbal King's wonderful line of weight loss pills.

Atkinson is a New Zealander now living in Queensland, Australia. If the FTC is able to collect, it will be a significant victory against spammers. While spammers do make a substantial amount of money from their illegal activities, $15 million is an extremely large amount to be taken out of their coffers. Spamming is about the money, and huge fines like these are a deterrent – if the spammer gets caught. The FTC was fortunate in this case because Herbal King flagrantly violated the CAN-SPAM Act by faking headers and not providing valid unsubscribe links. Next time they may not be so lucky.

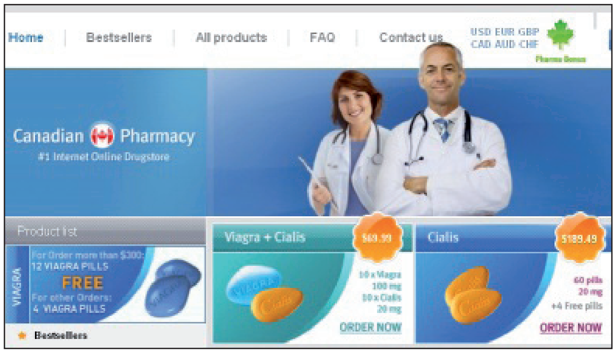


*Figure 3: The two most hated doctors on the Internet.*

Even *Facebook* got into the game this year. In October, a judge in San Jose, CA, awarded *Facebook* a $711 million judgment against alleged spammer Sanford Wallace. Filed in February, the suit alleged that Wallace sent misleading messages to *Facebook* users with malicious links, tricking them into giving up their credentials. In addition to the fines, Wallace and his gang face possible prison time for their actions. Wallace declared bankruptcy in an attempt to avoid payment of the various suits he has been saddled with (and lost).

So, while in general spammers get away with what they are doing, sometimes it does catch up with them. And we, in the anti-spam and e-security community, can enjoy a little bit of schadenfreude, if only for a little while.

## 7. BLACK SEO

One of the biggest trends in spam over the past two years has been black search engine optimization, or black SEO. 2009 was not the year it started, but it certainly was the year in which it really accelerated.

Black SEO comes in two main flavours:

1. Malvertising. When you perform a search on *Bing*, *Google* or *Yahoo!*, you will often see sponsored search results on the side of the screen. In legitimate cases, sponsors pay for their ads to be displayed on the side, in the hopes that the user will click on them and be directed to their websites. This is the advertising model that provides so much income for *Google*, *Microsoft* and *Yahoo!*.

   However, spammers and distributors of malware have exploited this facility by also purchasing ad space; when a user performs a search and clicks on the sponsored result, the page is not a legitimate web page but instead is a link to malware. The user has been lulled into a false sense of security, assuming that all paid-for advertisements are legitimate – after all, who would pay to advertise malware? Doesn't it destroy the attacker's cost model? The answer is no, it doesn't, and malvertising is something that strikes at the very heart of *Google*, *Microsoft* and *Yahoo!*'s revenue model.

2. Page rank optimization. This is similar to malvertising, but in this case the attacker doesn't need to buy advertising space in order to infect a user's computer. Instead, all he needs to do is make sure that his page is near the top of the search ranking.

   To accomplish this, a spammer will create a malicious page containing links to malware, or perhaps their spammy product, or perhaps a phishing page. They will then utilize a variety of techniques to make sure that users see their page. One such technique is to look for what the current most popular search term is, such as 'Jessica Biel' (thank goodness we have finally moved on from Britney Spears and Paris Hilton). They will then tag their web pages with 'Jessica Biel' meta-tags. At the same time, they will send out armies of botnets to plague discussion forums and blog comment forums with pointers to these web pages. With so many web pages and pingbacks pointing to their malicious page, the trap is set.

   *Google*, *Bing* and *Yahoo!*'s web crawlers crawl the Internet, indexing popular pages. Because spammers know that Jessica Biel is a popular search target, and because so many pointers on everyone else's pages now point to their spammy landing pages, the landing pages end up near the top of an Internet search. The result? Somebody legitimately searching for Jessica Biel will see the spammer's page near the top of an Internet search. They click on the link and voilà – free traffic is driven to the malicious site.

Black SEO in each flavour destroys the confidence of the end-user and, among spammers, is the growth industry of 2009.

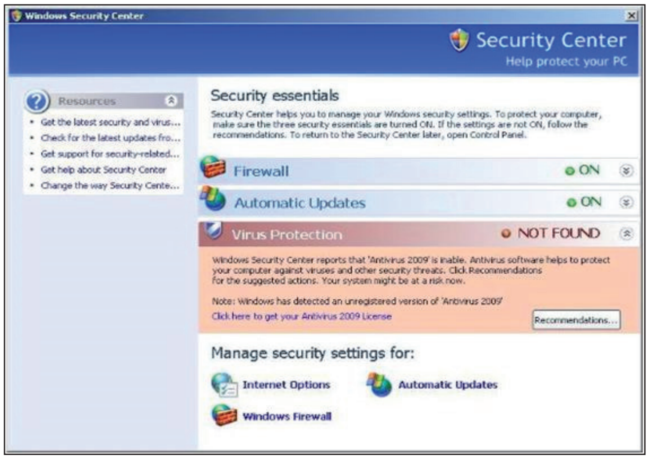
## 8. GOING ROGUE

Reputation hijacking continues unabated in the world of spam and malware, and social engineering is the tactic of choice.

Social engineering is the process by which an attacker will prey on a person's emotions in an attempt to get them to do something that they otherwise might not do. Two of the primary emotions that are targeted are the same as those that drive the stock market: fear and greed.

The general public has a vague notion that they need to run security software in order to keep their systems clean. They understand that there are computer viruses out in the wild that mean to do their computers harm; they have a genuine concern. Malware writers understand this. They know that people want to avoid getting infected, so now they provide a solution – cheap, or sometimes even free, anti-virus software!

To accomplish this, a well-known piece of software such as *Microsoft*'s *Windows Security Center* will be spoofed (see Figure 4). The user, recognizing *Microsoft*'s splash page reminding them that they have no anti-virus protection, can't resist the lure of cheap or free software to protect them from the nefarious world out there. They are also fearful of becoming the next virus victim and want to prevent this. Not necessarily thinking things through (when our emotions interfere with the logical parts of our brains they usually win), they download the advertised cheap or free software and install it. Unbeknownst to the user, their system is infected and more vulnerable than ever.



Security essentials

Security Center helps you to manage your Windows security settings. To protect your computer, make sure the three security essentials are turned ON. If the settings are not ON, follow the recommendations. To return to the Security Center later, open Control Panel.

Windows Security Center reports that 'Antivirus 2009' is inable. Antivirus software helps to protect your computer against viruses and other security threats. Click Recommendations for the suggested actions. Your system might be at a risk now.

Note: Windows has detected an unregistered version of 'Antivirus 2009'

Click here to get your Antivirus 2009 License

Manage security settings for:

*Figure 4: Screenshot of FakeXPA, the sixth most prevalent piece of malware detected by Microsoft in the first half of 2009.*

The story of rogue anti-virus software is not new to 2009. It has been going on for a while. What makes the rogue anti-virus story newsworthy for 2009 is that it is still a big problem and is getting worse.

## 9. MICROSOFT SECURITY ESSENTIALS FOR FREE

Long criticized for its insecure software, or rather the perception of insecure software, *Microsoft* made a splash into the home-user market in 2009 when it released *Microsoft Security Essentials* [3], a free anti-virus program for registered users of *Windows*.

*Microsoft* has already made significant inroads into keeping software secure:

1. The *Malicious Software Removal Tool* [4] is a free tool for scanning and removing unwanted and malicious software for registered and unregistered copies of *Windows*.
2. *Microsoft Update* is an automated process for downloading critical updates to keep your software secure. The more up to date you keep your system, the less likely it is to be exploited.
3. In 2002, *Microsoft* launched its Trustworthy Computing [5] initiative to improve public trust in its own commercial software. In addition, the company participates in a number of industry collaborative groups such as the Anti-Phishing Working Group (APWG), the Messaging Anti-Abuse Working Group (MAAWG) and the National Cyber Security Alliance (NCSA).
4. *Microsoft*'s Software Development Lifecycle requires that all of its products go through the Secure Windows Initiative, a process where potential security risks are identified and mitigated.

What makes the new home-user anti-malware product different is that it is free; the company now offers consumers anti-spam and anti-malware services, putting it on a par with other traditional security vendors such as *McAfee* and *Symantec*, and joining the ranks of free anti-virus providers including *AVG*, *Avira*, *Alwil*, *PC Tools* and others.

There's now no reason for a licensed user not to run anti-virus software. There are many choices out there, and *Microsoft* recommends you run something. So does the anti-spam and anti-malware community.

## 10. LOTS AND LOTS OF HACTIVISM

In October, an unusual article was posted on the technology blog *Neowin* – it was a large posting containing approximately 10,000 usernames and passwords belonging to *Hotmail* users. Many theories floated about. Whose usernames were these? What were they used for? How did the hacker gain access to them? Is my username and password at risk? Are these victims of a phishing scam? Did they get past *Hotmail*'s spam filter? The problem was complicated further by the fact that *Yahoo! Mail* and *Gmail* (*Google Mail* in Europe) accounts were also compromised, with various account details from those services also posted [6].

Regardless of how the hacker gained access to the accounts, what became painfully clear was that users in general do not follow good security practices. The most common password was '123456'. The second most common password was '123456789'. Armed with information like this, an attacker wouldn't necessarily need to know someone's password to break into their account. All they would need would be a lot of usernames and then they could try those two passwords and see which ones turned the key in the lock. Given enough usernames and passwords, some of them will undoubtedly unlock the doors that seal shut financial records. Who needs lock picks (other than magicians and locksmiths)?

While some hacker somewhere broke into a bunch of people's email accounts, in December, another news story broke. One of the hottest stories of the past decade is that of global warming. On the one hand, groups of scientists have published mountains of evidence indicating that the earth's global and atmospheric temperature is increasing and that this will change weather patterns, which will lead to decreased living standards in most of the world and negatively impact the prosperity of humanity. The scientists believe that this change in the world's climate is primarily the result of human activity.

Meanwhile, sceptics claim either that the evidence for global warming is overstated, or that its potential impacts are exaggerated, that the links between human activity and global warming have a weak or unknown correlation, or that the economic costs of preventing climate change outweigh the benefits of attempting to reverse it.

As the world prepared for its leaders to converge in Copenhagen to discuss potential solutions to the problem, a story broke. A hacker had broken into a server used by the Climatic Research Unit (CRU) at the University of East Anglia in Norwich, England. The hacker stole and disseminated over a thousand emails and other documents that had been compiled over the course of 13 years. The website RealClimate was then hacked and portions of the emails were uploaded to the site. What was particularly damaging, depending on how you look at it, was the way in which the emails could be interpreted. The sceptics claimed that the emails and documents were proof of a massive conspiracy to hide or manipulate data in order to support their case for global warming. One excerpt, written by Kevin Trenberth, a climatologist at the National Center for Atmospheric Research, discussed gaps in the understanding of recent temperature variations: 'The fact is that we can't account for the lack of warming at the moment and it is a travesty that we can't.' [7]

Of course, there are different ways to interpret the emails. Trenberth told the Associated Press that the phrase was actually used in reference to an article he authored calling for improvement in the measuring of global warming to describe unusual data, such as rising sea surface temperatures. The word 'travesty' refers to what Trenberth sees as an inadequate observation system.

The fallout from all of this is entirely political; the emails can be interpreted in different ways by different people.

Once again, the politics is driving cyber attacks in an attempt to get people to support a certain set of beliefs. We need to be careful what we say and do because somebody with an army of botnets, or a certain set of computer skills, might not be on our side and might have the mechanism to do us a great deal of harm.

*Figure 5: What, if anything, will be the fallout of Climategate?*

## CONCLUSION

Well, that's the way I saw the security world this year. There were other notable stories that didn't make my cut: Canada finally got around to passing an anti-spam bill (almost), ICANN is set to release a bunch more top-level domains, and URL-shortening services were abused in droves. But I think the stories above are the ones that made the greatest impact on the world in general.

As we enter the new year, I look forward to seeing what stories unravel in 2010.

## REFERENCES

[1] http://www.theregister.co.uk/2009/11/10/fireeye_takes_out_ozdok/.

[2] http://voices.washingtonpost.com/securityfix/2009/07/pcs_used_in_korean_ddos_attack.html.

[3] http://www.microsoft.com/security_essentials.

[4] http://www.microsoft.com/security/malwareremove/default.aspx.

[5] See http://www.microsoft.com/mscorp/twc/default.mspx for more details on this initiative.

[6] See Virus Bulletin, December 2009, p.12.

[7] http://en.wikipedia.org/wiki/Climategate.

# PRODUCT REVIEW

## ALWIL AVAST! 5

*John Hawes*

*Alwil*'s *avast!* product has a pretty enormous user base, as evidenced by the company's recent 100 millionth user celebrations. Its renown and popularity are assisted no end by the free edition which seems to have taken up permanent residence in the top-ten lists of most freeware download sites. Version 4 has been with us for over six years now, and the prospect of a major new release has brought growing levels of excitement and anticipation among the product's huge legions of fans.

When rumours of a new edition first reached the *VB* lab in the summer, it went straight to the top of our must-review list, but delays in issuing the final release have dragged on, and with a new release deadline extended to sometime in the new year, we just couldn't wait any longer. Late beta versions of the product have been made publicly available, so we'll be taking a quick look at both the free edition and the main *Internet Security* suite version, which features a number of interesting extras. As both versions are still in beta, we'll be skimming briefly over any minor bugs observed, on the assumption that these wrinkles will be ironed out by the time the full release is finalized. We will mainly be focusing on the suite edition in this review, but where applicable will note any differences observed between the two.

## COMPANY, INFORMATION AND SUPPORT

*Alwil* is one of the veteran brands in anti-malware, and its products have been regular entrants in our VB100 comparative reviews since official records began in January 1998 (see *VB*, January 1998, p.10). The company's first standalone review in these pages was in 1995, when the product included detection for a 'massive' 3,103 viruses (see *VB*, February 1995, p.21). Things have come a long way since then.

The Czech company has been in business since 1991, steadily building up a broad product range around its core desktop anti-malware offering. Solutions for multiple platforms including *Mac*, *Linux* and PDAs, consumer and business versions including management systems are all available. Details of the full product range are available on the company website (www.avast.com), and most are offered with a generous 60-day free trial period. One of the most interesting offerings is the *BART* (*Bootable Antivirus and Recovery Tool*) CD, which promises a complete bootable environment for comprehensive and secure cleaning and removal – something we've not had time to look into properly, but which possibly merits a review of its own. A full online purchasing system is available for most solutions and market segments, with easy access provided to a comprehensive network of local resellers and distributors for bulk corporate orders.

Best known of all the products, of course, is the free home-user version, the users of which must make up the bulk of the 100 million figure boasted of on the company's website. The current official version of this is pretty similar to the professional editions we have seen in VB100 reviews for the past several years, with its distinctive *Winamp*-style simplified interface (a more advanced version is provided for those, like us, with more specialist needs), and it seems to have served its users pretty well.

One of the most useful features of the company's web presence is a bustling support forum (forum.avast.com), where a huge and highly active community of enthusiasts swap tips, help out newbies, discuss new and desired features, and generally revel in their fondness for the product. Most questions, even on obscure and complex issues, seem to be met with a flurry of responses within moments. Company representatives also seem to be pretty active, moderating and providing expert assistance where required.

A more official knowledgebase of standard articles is also provided. This is kept pretty well stocked with answers to common issues, as well as a simple step-by-step troubleshooting process, but the forum is a much richer and more powerful support resource. For more specific or sensitive issues, registered users can also submit full support tickets using an online system.

Elsewhere on the website, a small section of malware information seems somewhat neglected and behind the times (the latest WildList displayed was from early 2008 when we checked), but does include an interesting and highly granular statistics page under the heading 'Summary of Virus Reports', which seems to be much more scrupulously maintained.

The final section of the website carries the standard information on the company, news items and awards, press releases on major company events, testimonials from customers around the world, information on partners and affiliates, and of course the now obligatory company blog. *Alwil*'s blog is perhaps more interesting than many, with the recently installed CEO Vince Steckler frequently posting strident and occasionally controversial opinions on a range of issues surrounding malware and security.

A recent blog post provided links to the beta versions of the new product range, which we keenly downloaded for this review. With ample information on the products within easy reach, it was time to install them and see what they had to offer.

## INSTALLATION AND CONFIGURATION

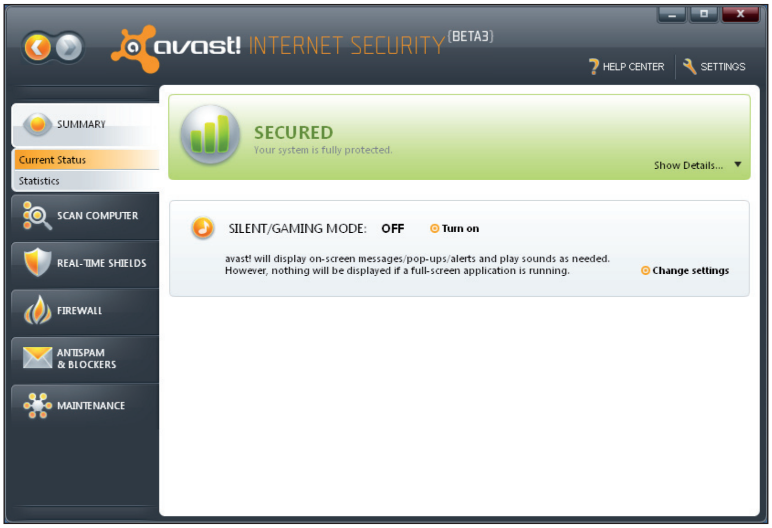
Both products were made available as full downloads with recent updates rolled in. This made for a good start for us, having observed in our most recent comparative that many vendors seem intent on only allowing products to be installed directly from their websites, or in some cases providing product downloads with extremely aged data inside, requiring a lengthy update process once up and running. It seems to us that it is far better, for most purposes, to have a product which can be installed to a state that provides a decent level of protection before venturing online with a new machine – this is exactly what is provided here.

The installation process itself is fairly standard, and in its plain and simple appearance differs little from the process of installing version 4.8 (a very familiar task in the *VB* lab thanks to its participation in numerous comparatives). The main item of note during the process is the community membership scheme. This is pretty standard in most products these days and allows companies to closely monitor just what is hitting the systems on which their products are installed. This not only allows for interesting statistics to be generated, such as those displayed on the website, but also helps focus the attention of analysts on the types and vectors of attack which really matter. The installation process completes fairly speedily, and for the suite product at least requires a reboot to finalize.

With installation completed, we finally got to feast our eyes on the new-look product. The previous version, as we have commented regularly in the past, had a rather distinctive look which was somewhat past its best. We have often found the layout, even in the advanced mode, somewhat unappealing, confusing and occasionally a little slow to respond, so we had been looking forward to seeing what changes had been wrought. We were not disappointed. A quick straw poll of the lab resulted in a unanimous victory for 'wow – that looks fantastic'.

The GUI has a very slick, clean, stylish and modern look, with a nice, simple layout. A row of tabs is arranged down the side of the screen in the manner which is becoming something of a standard in quality security solutions. The tabs are split into a sensible selection: a main summary page, an on-demand scanner, settings for the real-time shields, controls for the firewall and anti-spam filter (these last two are absent from the free version), and product maintenance. The summary offers a nice, subtly green bar to indicate that everything is operating properly, the controls for the silent/gaming mode, and a space for additional messages such as warnings about reboots being required.

Each of the other sections has a main screen and a series of sub-tabs for controlling and configuring different aspects of the feature under consideration, with most offering a



SECURED
Your system is fully protected.

SILENT/GAMING MODE: OFF

avast! will display on-screen messages/pop-ups/alerts and play sounds as needed.
However, nothing will be displayed if a full-screen application is running.

link to ‘expert settings’ for more detailed tuning. A general settings button in the top right leads to a wide selection of additional controls, while a ‘Help Center’ button next to it not only opens the help file, but in most cases picks out the appropriate page for the section the user is currently viewing. A few more such links might be useful in the expert control areas, but in general enough information is provided with each control for users to figure them out with minimal effort.

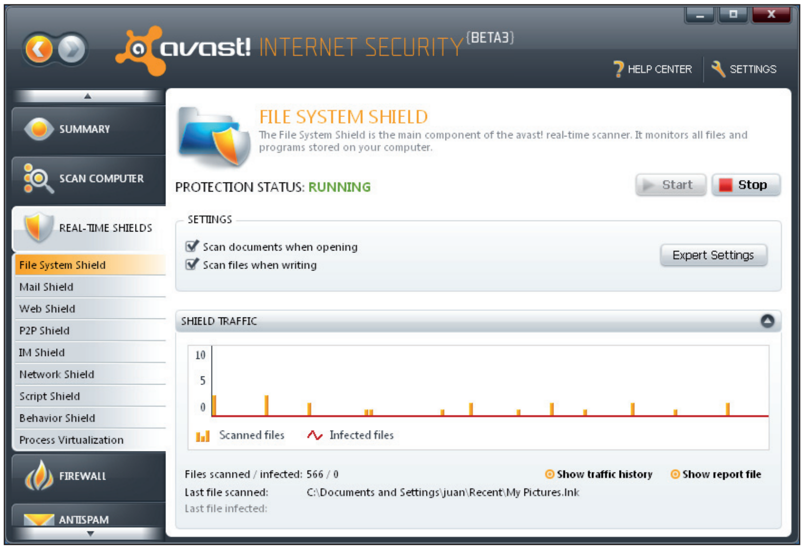
Overall, the layout is excellent, very simple to navigate, while providing splendid depth and breadth of configuration throughout. We will look a little more deeply at each separate area as we put them through their paces.

## SYSTEM PROTECTION AND MALWARE DETECTION

Protection against malware attack is, of course, the main function of the product, and *avast!* approaches this in a number of different ways. First off is the standard on-demand scan. The controls for this offer a selection of standard jobs. A ‘quick scan’, which checks the system memory and system drive and runs fairly speedily, completes in around 10 minutes on a fairly low-powered but well-used netbook. The ‘full scan’ is a little more thorough, while the ‘removable media’ option is designed to check items not usually connected to the system – USB thumb drives for example. There are a few other pre-set areas, such as configuration of scanning from the context menu within *Internet Explorer*, or scanning while the screensaver is active. The final option is to create a custom scan, which provides a nicely thought out set of stages to design and implement a scan.

Each of these can be adjusted in all manner of ways, including running on a schedule – although, somewhat unusually, no scan is set to run regularly by default. For most users – at least users of always-on desktop systems rather than laptops and netbooks – the on-demand scanner will mainly be used for a once-a-week check-up, probably run in the middle of the night. The fact that *Alwil* has chosen not to offer a suggested time for this is interesting – perhaps a sign of the growing use of mobile systems and sensible implementation of power saving – but users should probably try to run occasional thorough scans to check for nasties buried in their machines.

The real-time set-up is much more useful in terms of keeping one’s system safe from penetration in the first place of course, and *avast!* offers a pretty comprehensive selection of filters watching all conceivable points of access. The standard filesystem scan, usually the main point of contention for those users who find security products upsettingly intrusive, is fairly light on system resources thanks to its avoiding on-read scanning for most file types by default. A more comprehensive selection of file types are only checked when being written to the system in the first place, or when being executed (when the real danger is likely to arise). Most of the settings are fairly sensible, and a huge array of fine tuning is provided for every aspect of the monitoring; one of the more unusual sets of controls in here is the automatic checking of autorun files on removable media and boot sectors on floppy disks – again, a wise decision for ensuring safety.

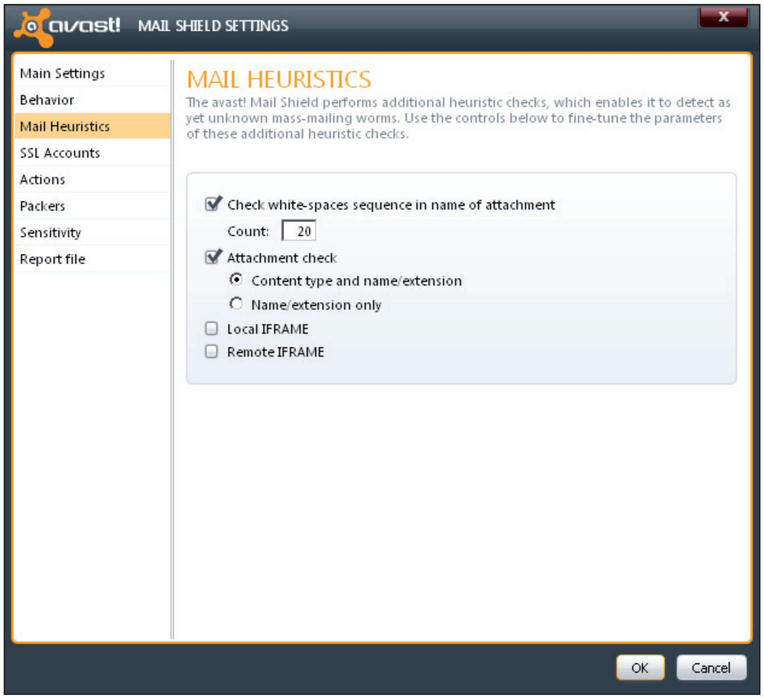


FILE SYSTEM SHIELD
The File System Shield is the main component of the avast! real-time scanner. It monitors all files and programs stored on your computer.

PROTECTION STATUS: RUNNING

SETTINGS
☑ Scan documents when opening
☑ Scan files when writing

Files scanned / infected: 566 / 0
Last file scanned: C:\Documents and Settings\juan\Recent\My Pictures.lnk
Last file infected:



MAIL HEURISTICS

The avast! Mail Shield performs additional heuristic checks, which enables it to detect as yet unknown mass-mailing worms. Use the controls below to fine-tune the parameters of these additional heuristic checks.

☑ Check white-spaces sequence in name of attachment

Count: 20

☑ Attachment check

● Content type and name/extension

○ Name/extension only

☐ Local IFRAME

☐ Remote IFRAME

We exercised both the filesystem shield and the on-demand scanner pretty thoroughly with the full set of tests from the most recent comparative (see *VB*, December 2009, p.16), and with the benefit of an extra month's worth of updates a small improvement was shown, even over the already excellent scores achieved by the version 4.8 product. Across all the RAP sets, now several weeks old, very little was left undetected. We compiled a small set of more recent samples, including some gathered after the product had been downloaded, to emulate a RAP test in miniature (testing the product's reactive and proactive detection abilities). We found similarly strong detection levels, declining somewhat in the most recent and retrospectively gathered items. Overall, detection seems as superb as we have come to expect from *Alwil*'s products; we look forward to seeing one or both of the products appearing in a full comparative review so that we can provide some more comprehensive detection scores.

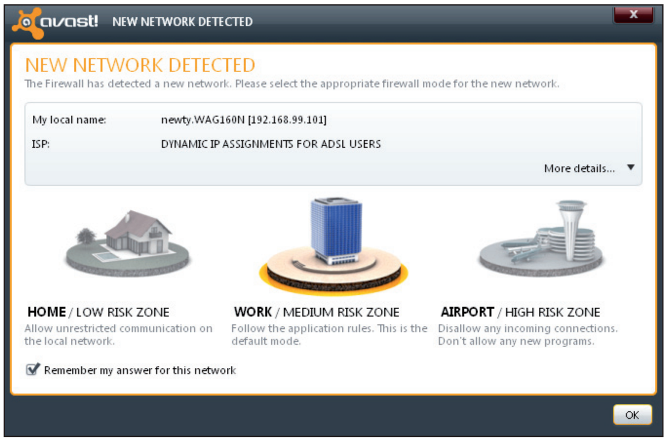
The next set of shields focuses on the main transport vectors through which malware is likely to enter a system: email, web browsing, peer-to-peer and instant messaging. Each is monitored for malcode making its way into or out of the system, with mail transport watched via SMTP, POP and IMAP, and even newsgroup traffic via NNTP. Suspect and infected mails can be marked with a warning (and a label denoting cleanliness can also be added to approved mails), while an interesting additional heuristic watches for large amounts of white space in attachment names – a technique often used to obscure file type identifiers.

The web scanner simply checks files being downloaded during web browsing, while the IM and P2P monitors do the same for various messaging and download tools – both come well stocked with a thorough list of common programs of both types.

Each of these last five shield types comes with an 'expert settings' button which provides tweaking options that are unique to the particular activities of the monitor, as well as some standard options. These include default actions when finding an infection (usually moving the item to the 'chest'), whether or not to scan inside archived or packed files, the sensitivity of the heuristics, and how much data to log to report files.

The final set of shields includes the script, network and behaviour shields. No tuning is provided for these beyond the simple on and off buttons, and indeed very little information is available on what they are up to. They seem fairly self-explanatory though, the first watching for scripts being run (although exactly what constitutes a script is not entirely clear), the second monitoring network traffic for dangerous content, and the third keeping an eye out for suspicious behaviour on the local system – all useful for stopping items not spotted by the standard detection methods. With so little information available on their operation, little testing of these shields was possible, but we did observe that some activities of items not detected by the main shields were being blocked.

All of these features will be familiar to most experienced *avast!* users, but they have been souped up considerably in terms of the fineness of the controls and, of course, in the attractiveness of their appearance. All the controls are really very well laid out and come with lots of useful comments and explanations to make their functions clear to all but the most unthinking of minds. The final entry in this section,



NEW NETWORK DETECTED

The Firewall has detected a new network. Please select the appropriate firewall mode for the new network.

My local name: newty.WAG160N [192.168.99.101]

ISP: DYNAMIC IP ASSIGNMENTS FOR ADSL USERS

HOME / LOW RISK ZONE
Allow unrestricted communication on the local network.

WORK / MEDIUM RISK ZONE
Follow the application rules. This is the default mode.

AIRPORT / HIGH RISK ZONE
Disallow any incoming connections. Don't allow any new programs.

☑ Remember my answer for this network

in the suite version at least, is a new one and is intriguingly labelled ‘Process virtualization’; we left this be, planning to look at it in more detail a little later on.

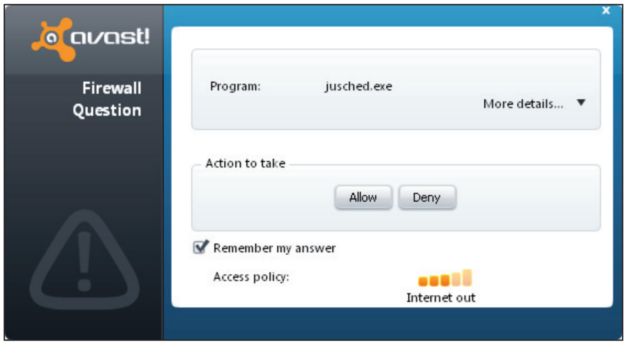
The third main part of the standard protective offerings, available only in the suite version, is a firewall. The initial settings for this are pretty basic: a colourful and attractive slider allows the user to choose from a selection of profiles depending on location. Even the ‘expert’ settings for this section are fairly limited, providing little more than a selection of choices on how to respond to new connections – the default is to decide automatically based on experience, but this can be adjusted to always block, always allow, or always ask the user; some additional options on notifications are also provided, defaulting to only inform the user if a new block rule is set up. This was something of a surprise given our past experience of enormously complicated firewall rule systems.

Looking deeper into the additional tabs, we found a great deal of information provided on network connections established and rules created for specific applications. Just about everything running on the systems we tested on was easily identified and treated appropriately, with *Alwil*’s massive user base doubtless playing a useful role in ensuring that most commonly used software is recognized and handled safely without interruptions. While the connections page seems to be informative only, the rules page allows the user to adjust the level of trust given to each product and individual executable, using a very nice, simple scale system on a by-profile basis. This rather non-standard approach is innovative and intriguing; it seems to offer a very respectable level of control without the complex and baffling tables and lists which are often relied on for firewall configuration; we thoroughly approved of it.

The final tab of the firewall section is labelled ‘network utilities’, and is another we shall look at in the next section.

## OTHER FUNCTIONALITY

Before looking at the selection of interesting and unusual extras, a quick glance over something a little more standard is called for. Anti-spam is pretty much a given these days in a suite product (it is not provided in the free edition). Once again, the configuration is pleasingly simple to use. The main part of the system uses standard spam detection rules, for which little configuration is provided beyond how often they are updated; messages can be tagged with a label if they are believed to be spam, and *Outlook* users can opt to have them moved to a dedicated junk folder. Addresses to which the user sends mails can be added automatically to a whitelist, which is simple to manage and to add to manually, with a blacklist similarly easy to operate. At some point we hope to be able to add such desktop products to our anti-spam testing set-up when running these reviews, as at the moment we can only provide measurements for server-grade solutions, and no details can be provided on how effective the filtering is.



Program: jusched.exe

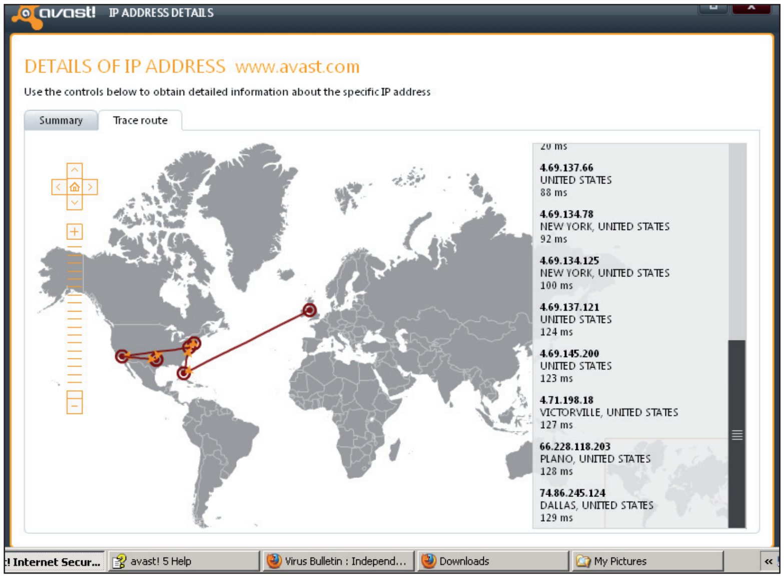
☑ Remember my answer

Access policy: Internet out

The second tab of the ‘Antispam and blockers’ section provides a system for blocking access to URLs. Disabled by default, it offers a means of populating a list with web addresses which are then completely blocked. Operating the entry system seemed a little fiddly at first, but as soon as one entry had successfully been made the rest was easy; wildcards are permitted in entries. It proved fully effective in a number of browsers, with just blank pages being displayed when an attempt was made to access the sites in question. It is not entirely clear exactly what the purpose of this is, but presumably it is intended as a sort of rudimentary parental control system.

A few other items are worthy of mention, starting with the Network Utilities option found in the firewall section. This offers a system for providing whois and traceroute details, with a pleasing graphic map display looking up a given IP address or domain and providing information about it, including details of all the steps required to reach it. A search for the route between our lab and avast.com led us, via several steps on the US east coast, to Dallas. The same section also includes an option marked ‘fix network stack’, which offers to try to reset the *Windows* network stack should networking run into problems; the option is clearly marked as potentially dangerous and only to be used in extreme cases.

The final tab on both the suite and free product is a set of maintenance tools, including the usual updater, product information and also giving access to the quarantine system, known to *avast!* users as the ‘Virus Chest’. In the updater we spotted one of the few bugs still evident in these beta products where, on occasionally demanding an update when it is not required (i.e. shortly after successfully updating), the product acknowledges that no further update is required but then insists that a reboot is required to complete the update. Doubtless this will be fixed by the time the final version is released.



DETAILS OF IP ADDRESS www.avast.com

Use the controls below to obtain detailed information about the specific IP address

20 ms

4.69.137.66
UNITED STATES
88 ms

4.69.134.78
NEW YORK, UNITED STATES
92 ms

4.69.134.125
NEW YORK, UNITED STATES
100 ms

4.69.137.121
UNITED STATES
124 ms

4.69.145.200
UNITED STATES
123 ms

4.71.198.18
VICTORVILLE, UNITED STATES
127 ms

66.228.118.203
PLANO, UNITED STATES
128 ms

74.86.245.124
DALLAS, UNITED STATES
129 ms

Another area which we were rather disappointed to find unfinished was the intriguing sandboxing system. This offers the ability to run any program in a secure environment, preventing it from making dangerous changes to the system – along similar lines to the popular *SandboxIE* utility. It is provided as a context menu option, either simply running a given item in a sandbox or setting it up to always run sandboxed in future – all very simple and easy. Unfortunately, one of the main items which one might wish to run in a secure setting, the usual desktop link to *Internet Explorer*, seemed immune to these menu options, even when an additional shortcut to it was added. Going via the expert settings and browsing all the way to the *IE* executable proved more effective, and *IE* could be opened inside the sandbox, indicated by a nice reassuring red border. It seemed pretty thoroughly blocked from making changes to the filesystem. Indeed, it seemed impossible to keep any files created or downloaded during a sandboxed session, which some may see as a little too secure; other similar utilities allow the user to browse the contents of the sandbox and fish out useful and trusted items before flushing it clean. Such functionality seemed absent, but with the documentation (and indeed the product itself) not quite finished, it was difficult to tell.

Other browsers were easier to associate with the sandbox, but were more difficult to persuade to run properly; *Firefox* simply presented an error message saying it was already running, unless an instance was indeed already running, in which case it would simply open a new, unsandboxed window. *Opera*, on the other hand, opened in a sandbox, but had much of its content blacked out and was barely usable. This could, of course, be an issue with the specific set-up of the test system, which had some other sandboxing software in place too; a few other systems we looked at had fewer problems, but also fewer browsers. Even during the course of this write-up new beta builds have been released, with many of the issues fixed being in the sandbox area, so I expect that all these little troubles will be removed when the product is deemed ready for full release.

## CONCLUSIONS

Once again we find ourselves thoroughly impressed with the latest generation of solutions. *Alwil*'s products have been on something of a wave recently, with a clean sweep of passes in the VB100 testing throughout 2009 and some reliably impressive detection rates. With this new version *Alwil* finally has an interface and a set of additional features to match its splendid detection. While some rival solutions have offered over-complicated and unfriendly systems, and others have gone for the trust-mother approach and not offered the user any control over their destiny, a*vast! 5* strikes an excellent balance between simplicity and control, with a very good depth of configuration made available without compromising ease of use. Information is vital here, and the interface designers have done a great job of providing lots of detail on what each section does, and how and why, couched in simple layman's language to enable all users to get the most out of the product without requiring a computing degree or hours of research.

Of course, being a beta, there are still a few minor hiccups which need fixing, but even with them in place the product outperforms a number of competitor products we've struggled with in our tests in recent years. That it manages to look pretty fabulous too is pure gravy; the suite version has some fun and useful extras, and the free version being available to all without charge is nothing short of a miracle. We look forward to seeing the various members of the version 5 range taking part in official VB100 tests in future, as on the evidence of this month's trials they are very worthy of the *VB* stamp of approval.

**Technical details:**

*Alwil avast! Internet Security Beta 3 5.0.259* and *avast! free edition Beta* 3 5.0.259 were variously tested on:

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

*AMD Athlon64* 3800+ dual core, 1GB RAM, running *Microsoft Windows 7 Professional.*

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

# COMPARATIVE REVIEW

## VBSPAM COMPARATIVE REVIEW

*Martijn Grooten*

Of the many reviews of the ‘noughties’ we have seen in the media in recent weeks, few have mentioned spam as being something that defined the decade. Yet in the past ten years, spam has grown from a mere nuisance to Internet users into a major field of criminal activity.

Even the most optimistic will find little reason to believe that the spam problem will disappear any time soon, but thankfully those in the anti-spam world keep working hard to protect end-users’ inboxes.

The first VBSpam comparative review of the new decade saw 15 products on the test bench: 14 full anti-spam products and one partial solution. Developers of three of the products that took part in previous tests decided to sit this one out in order to concentrate on new versions of their products; all of them hope to be back on board for the next test. However, four new products were included in this month’s test.

### THE TEST SET-UP

No major modifications were made to the test set-up and, as usual, the full methodology can be found at http://www.virusbtn.com/vbspam/methodology/.

As before, 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. The *Linux* products ran on *SuSE Linux Enterprise Server 11*; the *Windows Server* products ran on either the 2003 or the 2008 version, depending on which was recommended by the vendor.

Some changes have, however, been made to the award criteria. First, we decided to stop using the combined average performance of the products to define the award thresholds – with the performance levels of all products continually improving, the thresholds were in danger of becoming too dependent on one or two products performing significantly more poorly than the rest. Secondly, with the thresholds for the three award levels edging ever closer to each other, the difference between the levels was becoming very small and almost more dependent on luck than on a significant difference in performance.

As a result, a product’s performance will now be measured based on the value of its spam catch (SC) rate minus three times its false positive (FP) rate. A product will earn a VBSpam award if this value (referred to as the ‘final score’) is at least 96%:

$$SC - (3 \text{ x } FP) \geq 96\%$$

The simplification of the award structure should help to reduce confusion among end-users.

This does not mean we believe there is no difference in performance between the various products, and end-users are encouraged to compare the performance figures shown in the tables and to look at the relative positions of the products plotted in the VBSpam results graph.

Our intention is not to give an absolute value to the performance measured by us: a 98% catch rate in our test does not necessarily indicate the same as a 98% catch rate in another test, and does not mean that the product will catch 98% of a customer’s spam. However, the catch rates (or false positive rates) of two products in our test can be compared against each other.

### THE EMAIL CORPUS

The test ran from 1pm GMT on 14 December 2009 until 8am GMT on 4 January 2010 – a test period of almost three weeks, which included most of the Christmas holiday period (notorious for breaking spam records). The corpus contained 249,569 emails: 2,811 ham messages and 246,758 spam messages, where the latter consisted of 224,411 messages provided by Project Honey Pot and 22,347 messages sent to legitimate @virusbtn.com addresses.

As described in the previous VBSpam review (see *VB*, November 2009, p.22), the ham consisted of all legitimate messages sent to @virusbtn.com addresses, but with the senders of emails that regularly discuss spam- and malware-related topics (for example anti-spam discussion lists) excluded. Such emails regularly contain links to malicious and/or spamvertised URLs and we believe that not only are such emails unlikely to occur in the legitimate email stream of an average organization, but also that the recipients of such emails generally have the level of knowledge and technical ability required to whitelist these particular senders. To make up for these exclusions, we added to the corpus a number of email discussion lists on a variety of other topics.

In an attempt to make the test results more realistic, we decided to count no more than four false positives per sender for each product. This change should prevent a small mistake on a blacklist from having escalating effects if a certain sender sends many emails during a test period, but more importantly, it will reflect a real situation where legitimate senders whose emails keep being blocked are eventually whitelisted.

Another small change was that emails that claimed to have been sent from @virusbtn.com addresses were removed from the corpus: given the way our test is set up, products could have valid reasons for considering these emails to have been sent from a legitimate *VB* server. While this does not appear to have affected any product's past performance, we want to avoid the possibility of penalizing filters for making such assumptions.

A more showing change was the addition of two new categories: those of 'image spam' and 'large spam'. The former consisted of all spam emails that contained at least one embedded image, and the latter consisted of all spam emails with a body size of at least 50,000 bytes. Both types of emails are considered difficult to filter, especially using content scanning methods. We measured each product's performance on these sub-sets of the spam corpus, and while these measurements do not count towards the VBSpam award, they should give developers a better idea as to which part(s) of their filters can be improved upon.

## RESULTS

Starting from this test we will distinguish between full solutions and partial solutions. The latter are anti-spam products that are unlikely to be deployed on their own but are intended to work together with other solutions. As such, the performance of these products should not be compared directly to other solutions. This test contained one such solution (*Spamhaus Zen*), but *SaneSecurity*, which participated in the previous tests, would also fall into this category.

### BitDefender Security for Mail Servers 3.0.2

**SC rate (total):** 98.14%

**SC rate (Project Honey Pot corpus):** 98.86%

**SC rate (VB spam corpus):** 90.94%

**SC rate (image spam):** 97.53%

**SC rate (large spam):** 94.84%

**FP rate:** 0.605%

**Final score:** 96.33%

Having worked hard on their spam filter since the last test, *BitDefender*'s developers were eager to see the results of this month's test. Their hard work paid off: both the spam catch rate and the false positive rate improved a little, and in an area where the devil is in the details, this is no small achievement. *BitDefender*'s *Linux* server product thus wins its fifth VBSpam award in a row.

### Fortinet FortiMail

**SC rate (total):** 98.40%

**SC rate (Project Honey Pot corpus):** 98.79%

**SC rate (VB spam corpus):** 94.57%

**SC rate (image spam):** 97.83%

**SC rate (large spam):** 94.91%

**FP rate:** 0.427%

**Final score:** 97.12%

One of the clear high achievers of the previous VBSpam test, *Fortinet*'s *FortiMail* appliance saw its performance levels drop slightly on both fronts. However, this was not enough to prevent the product from earning a VBSpam award – the company's fourth in a row – and its developers will no doubt be extra motivated to improve its score during the next round of testing.

### Kaspersky Anti-Spam 3.0

**SC rate (total):** 95.94%

**SC rate (Project Honey Pot corpus):** 97.15%

**SC rate (VB spam corpus):** 83.71%

**SC rate (image spam):** 97.54%

**SC rate (large spam):** 93.67%

**FP rate:** 0.071%

**Final score:** 95.73%

It is hard not to feel that the anti-spam developers at *Kaspersky* are a bit unlucky: while their *Linux* server product was the only one to miss out on a VBSpam award in this test, it had fewer false positives than any other full solution. An improved spam catch rate should see the product winning an award again next time around.

### M86 MailMarshal SMTP

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

**SC rate (Project Honey Pot corpus):** 99.86%

**SC rate (VB spam corpus):** 97.01%

**SC rate (image spam):** 99.60%

**SC rate (large spam):** 98.25%

**FP rate:** 0.142%

**Final score:** 99.17%

*M86*'s *MailMarshal SMTP* spam filter, which runs on *Windows Server 2003*, made its debut in the VBSpam test in November with commendable results, but did even better in this test: it saw its false positive rate reduced significantly, while barely compromising on the spam catch rate, and it was the only product in this test with a final score of more than 99%. Moreover, neither large spam emails nor those containing images proved a problem for the product.

## McAfee Email Gateway (formerly IronMail)

**SC rate (total):** 99.59%

**SC rate (Project Honey Pot corpus):** 99.84%

**SC rate (VB spam corpus):** 97.11%

**SC rate (image spam):** 99.46%

**SC rate (large spam):** 97.95%

**FP rate:** 0.640%

**Final score:** 97.67%

For the third time in a row, the *McAfee Email Gateway* hardware appliance caught more than 99% of all spam and its performance in the various categories shows that this product is a good all-round filter. The product's false positive rate is slightly on the high side, but certainly not too high for it to win another VBSpam award.

## McAfee Email and Web Security Appliance

**SC rate (total):** 98.92%

**SC rate (Project Honey Pot corpus):** 99.49%

**SC rate (VB spam corpus):** 93.23%

**SC rate (image spam):** 98.84%

**SC rate (large spam):** 94.86%

**FP rate:** 0.462%

**Final score:** 97.53%

Another of the high achievers of the previous two tests, *McAfee*'s *Email and Web Security Appliance* demonstrated a very good spam catch rate once again – a small improvement compared to the previous test even – but also saw its false positive rate increase. While certainly not a bad performance, the developers will no doubt be eager to show that the rise in false positives was a one-off incident.

## MessageStream

**SC rate (total):** 99.14%

**SC rate (Project Honey Pot corpus):** 99.61%

**SC rate (VB spam corpus):** 94.38%

**SC rate (image spam):** 99.15%

**SC rate (large spam):** 97.55%

**FP rate:** 0.605%

**Final score:** 97.33%

The *MessageStream* hosted solution is another product whose performance dropped slightly compared to the previous test (in particular, it missed more legitimate emails than during previous tests), but this didn't stop it from performing well enough to earn a fifth VBSpam award in a row.

## Microsoft Forefront Protection 2010 for Exchange Server

**SC rate (total):** 99.06%

**SC rate (Project Honey Pot corpus):** 99.32%

**SC rate (VB spam corpus):** 96.49%

**SC rate (image spam):** 99.24%

**SC rate (large spam):** 98.07%

**FP rate:** 0.249%

**Final score:** 98.31%

The publication of the previous VBSpam test report almost coincided with the official release of *Microsoft*'s *Forefront Protection 2010 for Exchange Server* but the developers certainly weren't too busy to make improvements to their product. This test saw improvements in both the spam catch rate and the false positive rate, and with a final score of over 98%, *Forefront* was among the top performers in this test.

## MXTools Reputation Suite

**SC rate (total):** 97.65%

**SC rate (Project Honey Pot corpus):** 98.81%

**SC rate (VB spam corpus):** 85.97%

**SC rate (image spam):** 98.28%

**SC rate (large spam):** 94.86%

**FP rate:** 0.178%

**Final score:** 97.12%

*MXTools* sells three anti-spam solutions, each of which can be used as an add-on to improve an existing solution, but the three can also be used together to form a standalone spam filter. Apart from *Spamhaus ZEN plus DBL*, which is described below, the suite also contained *SURBL* and *Server Authority*.

*SURBL* is a DNS blacklist against which any domains contained in the body of an email can be checked: most spam contains a link to a website, and by looking at the domain part of the URL and checking this against a database of known bad domains, a lot of spam can easily be identified. Using the DNS protocol, the *SURBL* database can be queried repeatedly, with very short response times.

*Server Authority* also checks for bad domains, but rather than checking the domain itself, it looks up the name server associated with the domain: identifying domains associated with name servers used by spammers is a proactive way of blocking email containing bad domains. *Server Authority* was not only applied to URLs but also to the EHLO/HELO domain, the reverse DNS of the sending IP address and the domain part of the MAIL FROM address.

It should be noted that, when it comes to finding domains in emails, there is no unique way of doing so. We searched the bodies of emails for strings matching certain regular expressions, but it is possible to use less strict regular expressions that would catch more URLs, to follow redirects, or even to search URLs contained inside images. This may have improved the spam catch rate, but at the cost of a higher server load, longer processing times and, possibly, more false positives.

Even with the settings used, the suite's spam catch rate was better than some traditional anti-spam solutions. Like those, however, it was not without fault and an apparently incorrectly listed *SURBL*-domain, as well as a small mistake in the way domains were read from emails, caused a total of five false positives. Still, with a final score that is higher than around half of the full solutions tested, it easily won a VBSpam award.

(Note: A small error in the way the *SURBL* server was queried, for which *VB* and *MXTools* share responsibility, meant that the suite's performance over the first four days of the testing period was slightly lower than it could have been; without this error, the final score could have been a few hundredths of a per cent higher.)

## SPAMfighter Mail Gateway

**SC rate (total):** 97.60%

**SC rate (Project Honey Pot corpus):** 98.17%

**SC rate (VB spam corpus):** 91.85%

**SC rate (image spam):** 97.15%

**SC rate (large spam):** 92.44%

**FP rate:** 0.427%

**Final score:** 96.32%

*SPAMfighter*'s developers made use of the feedback we gave them after the previous two tests not just to review the product's settings, but also to make some changes to the solution itself. These changes certainly had a positive effect: the product's performance improved and it earned another VBSpam award. There is still room for improvement though, and with the product's relatively poor performance on both large spam and image spam, the developers might want to look into these areas.

## SpamTitan

**SC rate (total):** 99.65%

**SC rate (Project Honey Pot corpus):** 99.90%

**SC rate (VB spam corpus):** 97.13%

**SC rate (image spam):** 99.60%

**SC rate (large spam):** 98.59%

**FP rate:** 0.356%

**Final score:** 98.58%

*SpamTitan*, which runs as a virtual machine under *VMware*, had the highest spam catch rate in the last test and repeated that achievement in this test. The detailed results show that neither large spam nor spam containing images are a problem for the product and, as there were few false positives, it earns a VBSpam award with the second highest final score.

## Sunbelt VIPRE Email Security

**SC rate (total):** 98.77%

**SC rate (Project Honey Pot corpus):** 99.08%

**SC rate (VB spam corpus):** 95.65%

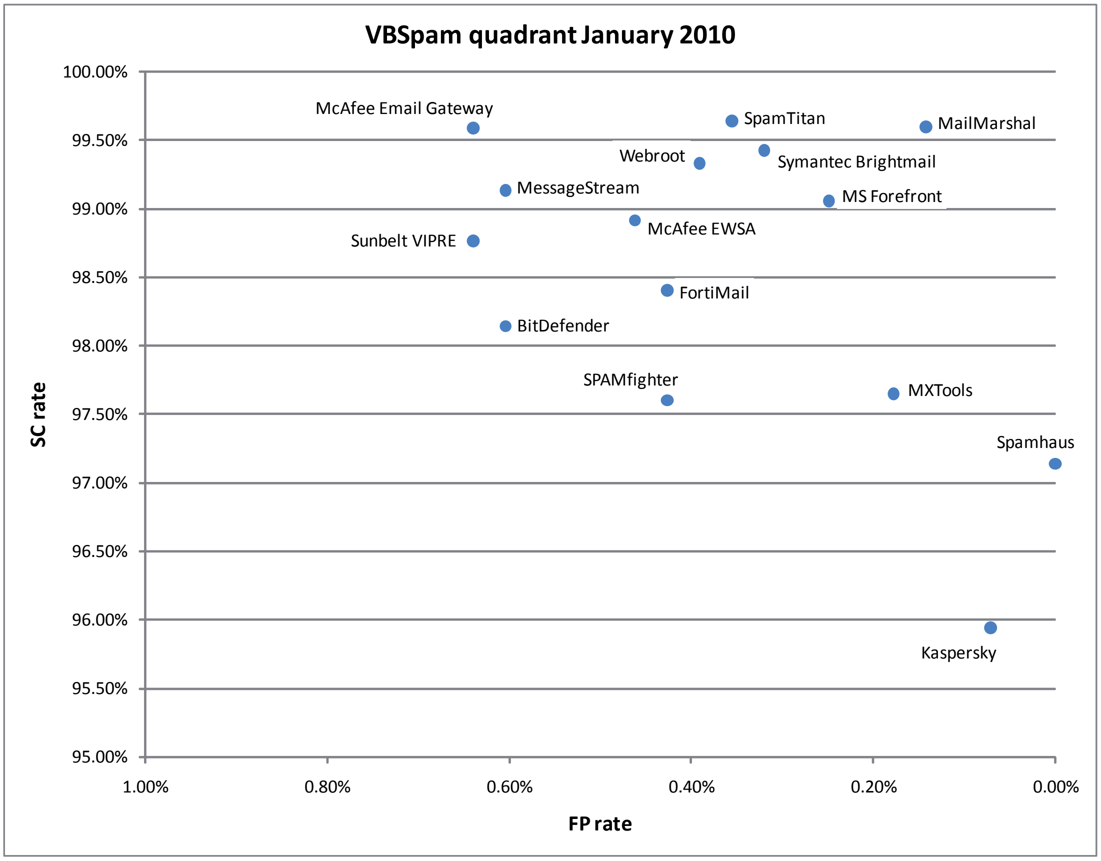
**SC rate (image spam):** 97.34%

**SC rate (large spam):** 94.56%

**FP rate:** 0.640%

**Final score:** 96.85%

Over the last few years, *Sunbelt* has become a big name in the world of computer security. Until recently, its anti-spam solution was known as *Ninja*, but, like its anti-malware solution, it is now known as *VIPRE*. The product runs alongside *Microsoft Exchange 2007* (which we ran on a *Windows Server 2003* machine), and once that has been installed,



the product is easy to set up and works almost immediately. While we ran the product mostly using its default settings, administrators have plenty of options to add, modify and disable anti-spam rules.

The product certainly had a good spam catch rate, although large spam and image spam are areas where there is some room for improvement. Its false positive rate was on the high side, but as the product was new to the test, this may well be the result of teething problems that may easily be solved by some modifications to the settings. In any case, the product won a VBSpam award with relative ease, and this should motivate the developers to perform even better next time.

## Symantec Brightmail Gateway

**SC rate (total):** 99.43%

**SC rate (Project Honey Pot corpus):** 99.88%

**SC rate (VB spam corpus):** 94.88%

**SC rate (image spam):** 99.39%

**SC rate (large spam):** 96.66%

**FP rate:** 0.320%

**Final score:** 98.47%

As the world's largest vendor of security software, it is not surprising that *Symantec* offers a range of anti-spam solutions. One of these is *Brightmail*, which was acquired by *Symantec* in 2004.

*Brightmail Gateway* is available both as a hardware appliance and as a *VMware* virtual appliance; we tested the latter.

The product works well using its default settings, but it comes with an easy-to-use web interface where it can be fine tuned to meet the needs of an organization. Like more and more spam products, it can also be used for outbound filtering and company policies can be enforced on outgoing email: given the importance of email reputation this certainly seems a good idea.

In our test, we only looked at inbound filtering and *Brightmail* certainly does an excellent job there, catching all but just over 0.5% of spam. A few mailing list emails and some newsletters were incorrectly blocked, but that didn't stop the product from debuting with a VBSpam award and the third highest final score.

| | | | | Total spam | | | Final score |
|---|---|---|---|---|---|---|---|
| | True negative | False positive | FP rate | False negative | True positive | SC rate | |
| BitDefender | 2794 | 17 | 0.605% | 4581 | 242177 | 98.14% | 96.33% |
| Fortinet FortiMail | 2796 | 12 | 0.427% | 3937 | 242821 | 98.40% | 97.12% |
| Kaspersky | 2809 | 2 | 0.071% | 10026 | 236732 | 95.94% | 95.73% |
| M86 MailMarshal | 2807 | 4 | 0.142% | 987 | 245771 | 99.60% | 99.17% |
| McAfee Email Gateway | 2789 | 18 | 0.640% | 1001 | 245757 | 99.59% | 97.67% |
| McAfee EWSA | 2795 | 13 | 0.462% | 2667 | 244091 | 98.92% | 97.53% |
| MessageStream | 2782 | 17 | 0.605% | 2130 | 244628 | 99.14% | 97.33% |
| MS Forefront | 2804 | 7 | 0.249% | 2318 | 244440 | 99.06% | 98.31% |
| MXTools | 2804 | 5 | 0.178% | 5803 | 240955 | 97.65% | 97.12% |
| SPAMfighter | 2797 | 12 | 0.427% | 5920 | 240838 | 97.60% | 96.32% |
| SpamTitan | 2801 | 10 | 0.356% | 873 | 245885 | 99.65% | 98.58% |
| Sunbelt VIPRE | 2793 | 18 | 0.640% | 3043 | 243715 | 98.77% | 96.85% |
| Symantec Brightmail | 2798 | 9 | 0.320% | 1404 | 245354 | 99.43% | 98.47% |
| Webroot | 2796 | 11 | 0.391% | 1639 | 245119 | 99.34% | 98.17% |
| Spamhaus | 2811 | 0 | 0.000% | 7064 | 239694 | 97.14% | 97.14% |

## Webroot E-Mail Security SaaS

**SC rate (total):** 99.34%

**SC rate (Project Honey Pot corpus):** 99.55%

**SC rate (VB spam corpus):** 97.23%

**SC rate (image spam):** 99.26%

**SC rate (large spam):** 97.31%

**FP rate:** 0.391%

**Final score:** 98.17%

*Webroot*'s hosted solution saw its false positive rate reduced significantly in this test, while it also caught more spam. Its performance on the difficult-to-filter *VB* spam corpus was especially striking, and with a final score of well over 98%, the product earns another well-deserved VBSpam award.

## Spamhaus ZEN plus DBL

**SC rate (total):** 97.14%

**SC rate (Project Honey Pot corpus):** 98.50%

**SC rate (VB spam corpus):** 83.47%

**SC rate (image spam):** 98.20%

**SC rate (large spam):** 94.64%

**FP rate:** 0.00%

**Final score:** 97.14%

*Spamhaus* (officially known as *The Spamhaus Project*) has been active for well over a decade and provides several DNS blacklists – databases of IP addresses known to be used by spammers. *Spamhaus ZEN* combines all three of the DNSBLs the organization provides and in this test, we combined it with *Spamhaus DBL*, which uses various heuristics to identify domains used by spammers. This DBL was checked for the domain part of every URL that appeared in the body of the emails – using the same method as used for *SURBL* and *Server Authority* – and also for the EHLO/HELO domain and the reverse DNS of the sending IP address.

*Spamhaus* has a rather conservative approach when it comes to adding IP addresses and domains to blacklists in order to minimize the number of false positives and, indeed, we did not see any false positives in this test. At the same time, the solution caught over 97% of the spam in this test, giving it a very good final score.

Still, the low catch rate for the *VB* spam corpus suggests that using *Spamhaus* on its own would lead to a fairly large number of spam messages reaching users' inboxes. This is why this is only a partial solution, the performance of

| | Project Honey Pot spam | | VB spam corpus | | Image spam* | | Large spam* | |
|---|---|---|---|---|---|---|---|---|
| | False negative | SC rate | False negative | SC rate | False negative | SC rate | False negative | SC rate |
| BitDefender | 2556 | 98.86% | 2025 | 90.94% | 419 | 97.53% | 209 | 94.84% |
| Fortinet FortiMail | 2724 | 98.79% | 1213 | 94.57% | 369 | 97.83% | 206 | 94.91% |
| Kaspersky | 6386 | 97.15% | 3640 | 83.71% | 417 | 97.54% | 256 | 93.67% |
| M86 MailMarshal | 319 | 99.86% | 668 | 97.01% | 68 | 99.60% | 71 | 98.25% |
| McAfee Email Gateway | 355 | 99.84% | 646 | 97.11% | 91 | 99.46% | 83 | 97.95% |
| McAfee EWSA | 1154 | 99.49% | 1513 | 93.23% | 197 | 98.84% | 208 | 94.86% |
| MessageStream | 874 | 99.61% | 1256 | 94.38% | 144 | 99.15% | 99 | 97.55% |
| MS Forefront | 1534 | 99.32% | 784 | 96.49% | 129 | 99.24% | 78 | 98.07% |
| MXTools | 2668 | 98.81% | 3135 | 85.97% | 292 | 98.28% | 208 | 94.86% |
| SPAMfighter | 4098 | 98.17% | 1822 | 91.85% | 483 | 97.15% | 306 | 92.44% |
| SpamTitan | 232 | 99.90% | 641 | 97.13% | 68 | 99.60% | 57 | 98.59% |
| Sunbelt VIPRE | 2072 | 99.08% | 971 | 95.65% | 452 | 97.34% | 220 | 94.56% |
| Symantec Brightmail | 259 | 99.88% | 1145 | 94.88% | 104 | 99.39% | 135 | 96.66% |
| Webroot | 1021 | 99.55% | 618 | 97.23% | 125 | 99.26% | 109 | 97.31% |
| Spamhaus | 3370 | 98.50% | 3694 | 83.47% | 305 | 98.20% | 217 | 94.64% |

* There were 16,970 spam messages containing images and 4,047 considered large; the two are not mutually exclusive.

which should not directly be compared to that of full solutions. Still, even as a partial solution, it easily earns a VBSpam award.

## CONCLUSION

This test saw several changes both to the way in which we measure results and to the make up of the email corpus. It is hoped that these changes will make it easier to translate the results to a real-world situation. We are working on some changes to the test set-up to make the next test even more realistic. In particular, we will be able to emulate a real situation where filters receive emails directly from the senders.

To achieve this, we will be able to send extra SMTP commands prior to the DATA command that inform the filter of the original sender's IP address and of their HELO/EHLO domain. For instance, this is possible in the Postfix MTA using the little known XCLIENT extension (http://www.postfix.org/XCLIENT_README.html), but we will be able to send different commands to different products. Using these commands, products will be able to block email pre-DATA (that is, before the actual email is sent) and the spam catch rate will be split into a pre-DATA rate and a post-DATA rate.

It should be noted that even in the current set-up, products have access to the original IP address and original HELO/EHLO domain. It will therefore not be mandatory for products to make use of these extended SMTP commands; we are well aware that for some products it may be harder, or even impossible, to change the way SMTP commands are dealt with. What will matter for the earning of a VBSpam award, as previously, are the total spam catch rate and the total false positive rate, regardless of how much (if anything) is blocked pre-DATA. It should, however, be an excellent opportunity for those products who want to boost their ability to block a large percentage of spam 'at the gate'.

The next VBSpam comparative review is set to run throughout February. The deadline for product submission will be 28 January 2010; any developers interested in submitting a product should contact martijn.grooten@virusbtn.com.

# END NOTES & NEWS

**Black Hat DC 2010 will be held 31 January to 3 February 2010 in Arlington, VA, USA**. Online registration is now open. For details see http://www.blackhat.com/.

**RSA Conference 2010 will be held 1–5 March 2010 in San Francisco, CA, USA**. For details see http://www.rsaconference.com/.

**The 7th Annual Enterprise Security Conference will take place 3–4 March 2010 in Kuala Lumpur, Malaysia** with the theme ‘Establishing effective strategies to secure the enterprise against new age cybercrime’. For details see http://www.acnergy.com/EntSec2010.htm.

**Security Summit Milan takes place 16–18 March 2010 in Milan, Italy** (in Italian). For details see https://www.securitysummit.it/.

**The 11th annual CanSecWest conference will be held 22–26 March 2010 in Vancouver, Canada**. For more details see http://cansecwest.com/.

**The MIT Spam Conference 2010 is scheduled to take place 25–26 March 2010**. Venue details and other information will be announced in due course at http://projects.csail.mit.edu/spamconf/.

**Black Hat Europe 2010 takes place 12–15 April 2010 in Barcelona, Spain**. For details see http://www.blackhat.com/.

**The New York Computer Forensics Show will be held 19–20 April 2010 in New York, NY, USA**. For more information see http://www.computerforensicshow.com/.

**Infosecurity Europe 2010 will take place 27–29 April 2010 in London, UK**. For more details see http://www.infosec.co.uk/.

**The 19th EICAR conference will be held 10–11 May 2010 in Paris, France** with the theme ‘ICT security: quo vadis?’. For more information see http://www.eicar.org/conference/.

**The International Secure Systems Development Conference (ISSD) takes place 20–21 May 2010 in London, UK**. For details see http://issdconference.com/.

**NISC11 will be held 19–21 May 2010 in St Andrews, Scotland**. Interest in attending can be registered at http://nisc.org.uk/.

**CARO 2010, the 4th International CARO workshop will take place 26–27 May 2010 in Helsinki, Finland**. The workshop will focus on the topic of ‘Big Numbers’. For more information see http://www.caro2010.org/.

**Security Summit Rome takes place 9–10 June 2010 in Rome, Italy** (in Italian). For details see https://www.securitysummit.it/.

**The 22nd Annual FIRST Conference on Computer Security Incident Handling takes place 13–18 June 2010 in Miami, FL, USA**. The conference promotes worldwide coordination and cooperation among Computer Security Incident Response Teams. For more details see http://conference.first.org/.

**CEAS 2010 – the 7th annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference – will be held 13–14 July 2010 in Redmond, WA, USA**. A call for papers has been issued, with a deadline for submissions of 26 March. As in previous years the conference will run a ‘spam challenge’. For details see http://ceas.cc/.

**Black Hat USA 2010 takes place 24–29 July 2010 in Las Vegas, NV, USA**. DEFCON 18 follows the Black Hat event, taking place 29 July to 1 August, also in Las Vegas. For more information see http://www.blackhat.com/ and http://www.defcon.org/.

**The 19th USENIX Security Symposium will take place 11–13 August 2010 in Washington, DC, USA**. For more details see http://usenix.org/.

**VB2010 will take place 29 September to 1 October 2010 in Vancouver, Canada**. *VB* is currently seeking submissions from those wishing to present papers at the conference, see p.7. For details of sponsorship opportunities and any other queries relating to VB2010, please contact conference@virusbtn.com.

## 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.