DECEMBER 2007

virus BULLETIN

Fighting malware and spam

CONTENTS

ISSN 1749-7027

IN THIS ISSUE

MULTI-PLATFORM

The author of MSIL/Yakizake claimed that 'very few implementations of multi-platform malware exist up until now'. Peter Ferrie explains why Yakizake, for one, does not qualify for the category.

page 5

MULTIMEDIA

Today's Internet users are accustomed to seeing and downloading interactive multimedia content on websites. Unfortunately such content can include more than one might expect. Christoph Alme and Dennis Elser present a round-up of recent multimedia vulnerabilities.

page 13

MULTIPLE PROBLEMS

Fewer than half the products submitted for this month's VB100 comparative review on Windows 2000 made the grade, largely thanks to some pesky polymorphic file-infecting viruses and a rash of false positives. John Hawes has the details.

page 16

vbSpam supplement

This month: Reza Rajabiun considers the implications of spam for developing countries and the persistence of the digital divide.

'The main trend I have observed this year has been the spread of malware activity across several forms of technology and applications.'

Eddy Willems, EICAR

A YEAR OF THREATS ACROSS SEVERAL TECHNOLOGIES

While waiting in the departure hall of a Russian airport on my return from an IT conference I reflected on the year that has nearly passed and noted that it has been interesting in every security aspect.

The main trend I have observed this year has been the spread of malware activity across several forms of technology and applications. It appears that the parties that are orchestrating security attacks are gaining an increasing foothold to build a stronger, more sustainable commercial economy based on carefully crafted security attacks.

Social engineering reached a high level of sophistication this year via the 'Zhelatin-Stormworm' gang, named after the trojan it circulated. This gang was responsible for what started out as the 'Storm worm'.

First spotted in the early part of the year, the spread of the Storm worm started via emails purporting to provide information on some severe storms that had struck parts of Europe at the end of January. Users who fell for the trick were directed to a website containing malicious code aimed at turning *Windows* PCs into spam bots. Over time, emails containing links to the Storm worm took on many different forms, with subjects ranging from supposed missile strikes to reports of genocide and other socially engineered trapdoors. The worm even got into users' blog accounts and created new blog entries with links to the trojan itself. Several million computers were infected worldwide as part of this massive botnet until it was broken down into smaller parts. And still the story continues.

Spammers took a step ahead in their ongoing battle against anti-spam measures by using images to defeat hash filtering and string matching. They also used malware-infected computers (e.g. the Storm worm botnet) to launch spam emails to defeat network/sender reputation filtering. *Excel*, RTF, PDF, RAR and even MP3 spam are just some of the other next-generation techniques spammers have used this year to avoid detection.

The banking industry continued to be a key target for phishing scams and highly sophisticated targeted attacks. As trojans became more technically complex, the malware writers implemented new techniques in their attacks, including filters that keep a closer track of users' online banking activity. Such tracking methods make it easier and more effective for fraudsters to collect account details using a variety of methods. I have seen very advanced dedicated phishing and spyware attacks against several large banks, but also some against smaller regional banks, which demonstrates the keen interest of organized criminals in this approach.

Cybercrime and real-life political unrest came together as a form of 'cyber war' causing general unrest in Estonia earlier in the year. Disputes over the relocation of a Russian Red Army monument not only led to arrests in the real world, but several Estonian government and other public sector and media websites were heavily targeted via Distributed Denial of Service (DDoS) attacks by an extremely active network of hackers. Several key sites were rendered unreachable.

The mobile malware industry has also been very active this year. 'Personalized' SMS spam, financial lottery scams, and several new items of spyware were reported for mobile devices.

It is concerning to see complex mobile trojans and spyware being developed by growing commercial entities, with the aim of making solid profits to support further development of the malicious economy. However, the increase in the volume of malware for mobile devices seems to be slowing (though it could be the calm before a storm). The rise of adware also seems

to have stagnated – of course this does not necessarily indicate that these threats will stop.

The *Mac* seems to be becoming increasingly appealing for malware writers, with several trojans appearing this year, such as DNSChanger which hijacks DNS settings and then redirects the user to malicious websites.

So what is the next step for viruses and information threats? Despite the emergence of new operating systems such as *Windows Vista*, new mobile content and devices like the *iPhone*, cyber criminals are still using tried and tested ways of attacking Internet users.

Furthermore, we have seen a significant return of DDoS attacks and attacks that use browser vulnerabilities to penetrate the system. The most significant thing that distinguishes the present situation from that of several years ago is the fact that email is not being used as the primary vehicle for spreading malware. Instead, instant messaging services and web exploits are two of today's key means of distribution.

Anti-virus and security vendors have improved their technologies considerably and introduced several new ones. Presently, end points or PCs are protected much more effectively than they were several years ago. The average length of time that most new malicious programs survive in the wild has been cut to a number of hours.

Company data is worth a lot of money on the dark side of the web and criminals will go to significant lengths to harvest it. But let's predict what will happen next. Malicious users will attempt to reach beyond the current security solutions – a task that is a shift from 'getting around' anti-virus programs or security devices and implies more action in fields that have not yet been mastered by normal security and anti-virus protection, or areas in which protection is not an option for any number of reasons. This is more than likely where the new front will be in the information war.

We will face more botnet problems, threats to Web 2.0 sites, *Windows Vista* malware, malware targeting online games, along with attacks on IM software and more problematic rootkits. I think that hackers will also turn their attention to virtualization software because companies are increasingly looking into virtualization for their defence.

I was so deep in thought at the airport that I nearly missed my chance to have one last chat with Irishka, a student from Rostov University whom I had met on my trip and who had helped me a lot in communicating with the locals. It occurred to me that we should all make the effort to invest more time in real life than in our virtual one before it's too late. Maybe it's time that malware writers considered this as well.

Prevalence Table – October 2007

Virus	Type	Incidents	Reports
W32/Netsky	Worm	1,985,492	34.61%
W32/Mytob	Worm	1,358,652	23.68%
W32/Bagle	Worm	699,466	12.19%
W32/MyWife	Worm	347,694	6.06%
W32/Virut	File	272,344	4.75%
W32/Zafi	File	151,562	2.64%
W32/Mydoom	Worm	143,486	2.50%
W32/Bagz	Worm	106,980	1.86%
W32/Stration	Worm	78,441	1.37%
W32/VB	Worm	74,208	1.29%
W32/Grum	Worm	59,226	1.03%
W32/Sality	File	55,037	0.96%
W32/Rontokbro	File	41,565	0.72%
W32/Autorun	Worm	31,506	0.55%
W32/IRCbot	Worm	29,357	0.51%
W32/Parite	File	28,197	0.49%
W32/Klez	File	27,433	0.48%
W32/Rjump	Worm	26,636	0.46%
W32/Sdbot	File	22,453	0.39%
W32/Bugbear	Worm	17,579	0.31%
VBS/Small	Worm	17,465	0.30%
W32/Rbot	Worm	14,060	0.25%
W32/Fujacks	File	13,162	0.23%
W32/Sohanad	Worm	10,565	0.18%
W32/Jeefo	File	10,008	0.17%
W32/Looked	File	8,666	0.15%
VBS/Butsur	Script	7,977	0.14%
W32/Tenga	File	7,116	0.12%
W32/Perlovga	Worm	6,133	0.11%
W32/Feebs	Worm	5,820	0.10%
W32/Mabutu	Worm	5,667	0.10%
W32/Fleming	Worm	5,519	0.10%
Others[1]		67,260	1.17%
Total		5,736,732	100%

[1]The Prevalence Table includes a total of 67,260 reports across 140 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

NEWS

YULETIDE GREETINGS

Festive greetings from VB: (clockwise from top left) Helen, John, Martijn and Allison.

The members of the *VB* team extend their warm wishes to *Virus Bulletin* readers for a very happy holiday season and a healthy and prosperous new year.

This Christmas *Virus Bulletin* has made a donation of clothing and other items to UK-based charity for the homeless *Crisis* (http://www.crisis.org.uk/).

VISTA FAILS TO REASSURE WEB USERS

According to a recent poll, 50% of visitors to the *VB* website do not believe that *Windows Vista* has made the Internet any safer.

On its release, *Microsoft*'s most recent operating system was hailed by chairman Bill Gates as being 'dramatically more secure' than other operating systems, but a year after its initial roll-out only 25% of visitors to www.virusbtn.com say they think it has made a positive impact on web security. One reader summed up: '*Windows Vista* does an OK job of protecting itself and its users, but virus writers will find a way around it and in the end security all comes down to the human factor.'

VB's second comparative review of anti-malware products for *Vista* will be conducted in spring 2008.

BOTNETS ROASTING ON AN OPEN FIRE

The FBI has revealed that eight individuals have been indicted, pled guilty or been sentenced for crimes related to botnet activity since the start of its 'Operation Bot Roast' in June. The operation, now in its second phase, has also seen the serving of 13 search warrants both in the US and by the FBI's overseas law enforcement partners. According to the FBI the operation has uncovered more than $20 million in economic loss and more than one million victim computers to date.

Meanwhile, *McAfee*'s annual Virtual Criminology Report has suggested that the biggest security threat in 2008 will be international cyber spying. According to the report governments and allied groups are already using the Internet for spying and cyber attacks, with national infrastructure network systems being targeted. The report, which draws information from NATO, the FBI, SOCA and several educational institutions, indicates that as many as 120 countries are currently using the Internet for espionage operations.

CALL FOR PAPERS

VB2008 OTTAWA

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

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

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

SUGGESTED TOPICS

A list of topics suggested by the attendees of VB2007 can be found at http://www.virusbtn.com/conference/call/. Please note that the list is not exhaustive – the selection committee will consider papers on any subjects relevant to the anti-malware community.

HOW TO SUBMIT A PROPOSAL

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

Following the close of the call for papers all submissions will be anonymized before being reviewed by a selection committee; authors will be notified of the status of their paper by email. Authors are advised that, should their paper be selected for the conference programme, the deadline for submission of the completed papers will be Monday 9 June 2008. Full details of the paper submission process are available at http://www.virusbtn.com/conference/.

LAST-MINUTE PRESENTATIONS

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

ANALYSIS

SOMETHING SMELLS FISHY

Peter Ferrie
Symantec, USA

Multi-platform malware is nothing new. In 1999 we saw the W32/W97M infector Coke and W32/HLP infectors SK and Babylonia. In 2000 we saw W32/HLP infectors Dream and Pluma; in 2001 we saw W32/Linux infector Peelf, followed by Simile in 2002 and Bi in 2006. In 2003 and 2004 we saw W32/W64 infectors MSIL/Impanate and Chiton. Three new multi-platform scripting viruses were seen in 2005 (see *VB*, November 2005, p.4) – and of course, there was the Morris worm in 1988.

These points are apparently lost on Paul Sebastian Ziegler, the author of MSIL/Yakizake. The virus author wanted to call his virus 'Akikaze' (Japanese for 'Autumn wind'), but I went with the Japanese word for grilled salmon. The virus author claims that 'very few implementations of multi-platform malware exist up until now' (despite the dozen that I've just listed), so he went ahead and wrote a 'multi-platform' virus and presented it at the DEFCON 15 conference.

MULTI-WHAT?

It's unclear why Mr Ziegler thinks that his virus is multi-platform, because the platform is the environment in which the application runs. It's not the CPU on which it is running, because it needs to interact with other hardware to survive. It's not the operating system, either, if the environment is a virtual machine of some kind, or the virus exists outside of the operating system itself (for example, a boot sector virus).

In this case, the virus runs on a particular platform that has multiple implementations – which include *Microsoft .NET Framework*, *Novell Mono*, and *DotGNU Portable.NET*. The platform is a hardware-independent virtual machine. The platform has been ported to several CPU architectures, but since it's hardware-independent, the applications running inside it can't see the CPU anyway. So it's really just the one platform. The virus is aware of the operating system, but that's irrelevant. It's still just the one platform.

There can be exceptions, of course, such as MSIL/Impanate (see *VB*, November 2004, p.6). Impanate is a file infector that understands both the 32-bit and 64-bit MSIL file formats. It's a MSIL virus, so it's not multi-platform, but it is multi-platform-*aware*. Yakizake is neither of these things.

THE VIRUS

The virus begins by looking for the *Thunderbird* address book. There is code to deal with Unix systems, *Macintosh* systems, and *Windows* systems, however due to a bug, only the Unix and *Windows* code works. The bug is that the code to check for the *Macintosh* system is identical to the code to check for all other Unix systems. As a result, the *Macintosh* code can never be reached. This means that the virus cannot replicate from *Macintosh* systems.

In the case of *Windows* systems, the virus will attempt to terminate all instances of the *Thunderbird* executable, in order to gain control over the address book.

The virus creates a list of all addresses that it can find. The first version of the virus accepts addresses in '*@*.*' format, where '*' can be any character. The second version of the virus restricts this to one or more case-insensitive alphanumeric characters before and after the '@', and no longer checks for the '.' character.

The virus also looks inside 'prefs.js' for the SMTP server information, and inside 'signons.txt' for the SMTP server password.

The virus creates different email messages, depending on certain characteristics. If the virus is sending from a German system to a German user, the subject will be 'Programmierung' and the message will be in German (an almost exact translation of the English message), otherwise the subject will be 'Programming' and the message will be in English. The virus chooses an 'advanced' message body if it is running on a Unix system, and the string '/gcc' exists in %path% or if it is running on a non-Unix system, and 'Visual Studio' exists in the %ProgramFiles% directory.

The 'advanced' message body is:

```
Hi,

I wrote this program using a new approach. Please
tell me what you think of it.
```

The 'average' message body is:

```
Hi,

I have recently started to try out programming!

This is one of my first programms. What do you think
of it?
```

On Unix systems, both messages continue with:

```
If the programm should not work instantly on your
non-windows-system you probably need to execute it
using mono. (mono-project.com)
```

After constructing the message, the virus sends it to each recipient in turn, using the host SMTP server and credentials, with the virus executable as an attachment. When the mailing is finished, the virus exits. There is no payload.

DUMB AND DUMBER

To create a virus because one did not exist before is just dumb. To incorrectly call it multi-platform is even dumber.

FEATURE 1

EXPLORING THE EVOLUTIONARY PATTERNS OF TIBS-PACKED EXECUTABLES

Rachit Mathur, Aditya Kapoor
McAfee, USA

This year we have seen a very large number of packed executables related to W32/Nuwar, aka the Storm worm, all of which have used a packer commonly known as Tibs.

Broadly speaking, Tibs is a polymorphic closed source packer that is used by its author(s) to obfuscate a variety of malware. All the malware we have seen packed with Tibs to date has been motivated by monetary gains, primarily involving spam.

Tibs packed executables evolve continually, thus allowing the malware to pass undetected through some anti-virus defences. This article presents an analysis of the techniques used by the Tibs packer and describes the reasons for its prolonged effectiveness. (Note: the terms 'packing' and 'encryption' are used interchangeably in this article.)

1. PROLIFERATION TACTICS

The 'Tibs gang' has been very successful in its use of social engineering – luring and tricking large numbers of users into downloading and executing its malware. We have seen downloaders, worms, mass mailers, proxy agents and spam-mailbots all packed with Tibs.

The Tibs gang uses a range of tactics to attempt to penetrate security defences at multiple levels:

- In an attempt to evade spam filters, the text of the emails in which malware is sent is modified frequently.
- To avoid network traffic recognition, variations are introduced in encrypted Overnet traffic.
- To defeat analysis tools used by cautious administrators, the malware installs kernel mode rootkits to hide files, processes etc. In order to minimize their footprint in the registry some variants infect binaries that are loaded at startup. The variant inserts its own loader code into the victim binary thus ensuring the malware will be loaded on system startup.
- To avoid detection by AV scanners the server hosting the malicious binary produces modified executables every so often (approximately every 15 minutes).

Figure 1: Google map – infected nodes.

In order to harvest samples from the servers hosting Tibs-packed files, we monitored thousands of IP addresses for a period of time. The list of IP addresses was updated continually with new links being added while dead links were dropped. Figure 1 is a snapshot of the Geo-Mapping of the IP addresses hosting these Tibs-packed files. (This information is not completely representative of the threat; however, it provides an approximate idea of where those executables were hosted at a point in time.)

Many of the aspects of this threat have already been discussed in *Virus Bulletin* [1, 2] and elsewhere [3–5]. This article adds to the previous articles by discussing the workings of the Tibs packer.

2. TIBS PACKER OVERVIEW

Tibs executables are packed polymorphically, i.e. the decryptor code differs among variants. However, the polymorphic engine is not contained within these executables, which means they do not have the ability to generate polymorphic variants on their own. This is where Tibs differs from the traditional notion of polymorphic malware, and its behaviour falls instead under what is commonly known as 'server-based polymorphism' – the server hosting the malware returns executables with polymorphic variations in the decryptor code when queried at different points in time.

Figure 2 is a diagrammatic representation of a typical Tibs-packed executable containing a 'server-side' polymorphic decryptor and the encrypted malcode. The underlying code may be a pure form of malware or encrypted either by a flavour of TEA [6], UPX etc. or by a combination of these.

Tibs-packed samples implement simple yet effective code transformations in their decryptors to hinder detection.

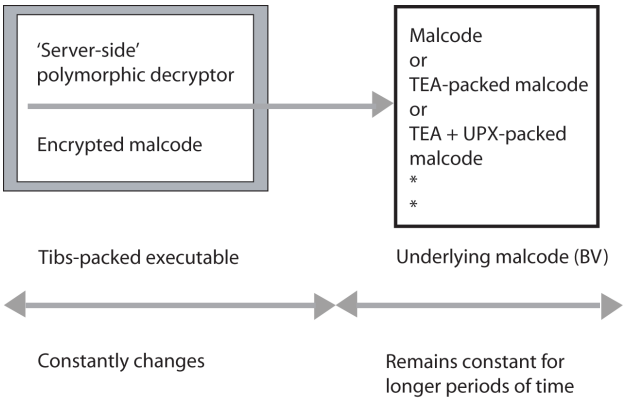


Figure 2: Typical Tibs-packed malcode.

Normally the decryptor code is fairly small and the code bytes of the decryptor are modified frequently, while the decryptor logic and underlying decrypted code (base variant) is changed less frequently – in regular polymorphic behaviour the decrypted code simply remains constant. It is the server-side nature of Tibs that allows the malware authors to manipulate the underlying code as well as the decryptor code.

The decryption steps of Tibs are outlined in Figure 3.

1. Locate the start address of encrypted data and size/end of the data
2. Calculate key(s): key[i]
3. Apply key(s)
4. Transfer control to decrypted code

Figure 3: Decryption steps.

The obvious first step is to locate the beginning and end of the data that needs to be decrypted. Then the key(s) need to be identified – typically there are two. Thereafter the key is applied to the encrypted data, one dword at a time, and finally control is transferred to the decrypted code. Although the decryption steps of many decryptors are the same as those shown here, the evolutionary trends of the decryptor code and the decryption algorithm itself are interesting in the case of Tibs.

3. TIBS EVOLUTION PATTERNS

The evolutionary trends of the Tibs polymorphic decryptor can be identified by analysing the differences between the executables as they change on the server hosting them. The morphing techniques used in the decryptor can be classified according to the frequency with which they are applied (high, medium or low frequency).

3.1 High-frequency morphing techniques

Here, at least one of the keys changes frequently and the executable is recompiled. Since the key is changed, the bytes of the entire encrypted data change, and this makes up the majority of the body of the executable. The decryption algorithm and the decryptor code remain the same except for the key.

This change is introduced a couple of times every hour to produce a new file from the server hosting the malicious executables.

3.2 Medium-frequency morphing techniques

These changes are introduced once every couple of days and involve the application of various code-morphing and anti-emulation techniques. The decryption algorithm remains the same but the code changes.

Some of the transformations that may be introduced are as follows:

a. Use of MMX instructions: code morphing using MMX instructions can be applied as shown in Figure 4.

```
                        movd    mm7,edx
mov [esi], edx →        movq    mm3,mm7
                        movd    [esi],mm3
```

Figure 4: MMX transform.

b. Use of fake *Windows* API (WAPI) calls: fake calls may be introduced to *Windows* functions such as 'CreateMDIWindowA', 'ILGetSize', etc. These API calls are fake because they are not called to perform the actual purpose for which they exist. Instead, null or junk parameters are passed and the returned values are validated during decryption. These return values (which are mostly *Windows* standard error codes) are typically used as one of the keys during decryption. For example, the SHFindFiles function displays the search window user interface if called 'properly', but the malware makes this call with null parameters and without calling CoInitialize, resulting in the error code 0x800401f0. This is then used as one of the decryption keys.

c. Other techniques such as register renaming, CFG obfuscation, dead code insertion, replacing SESE

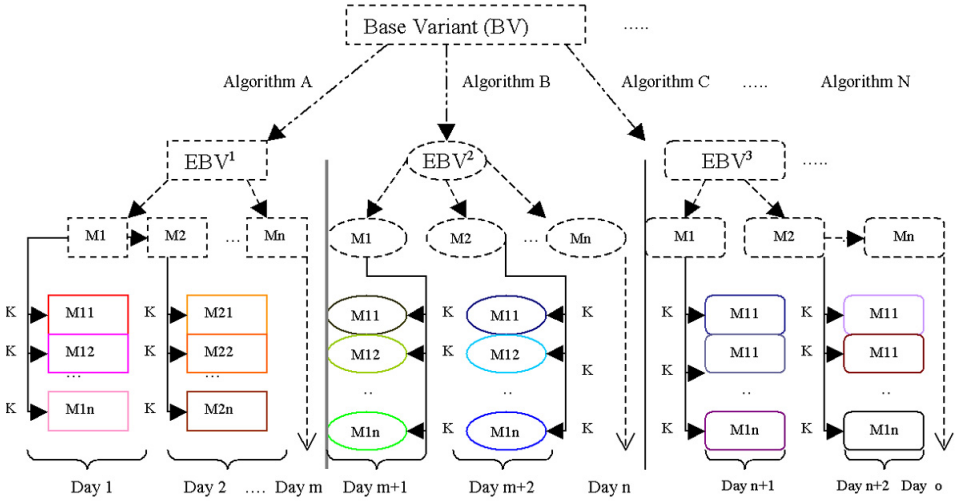


Figure 5: Evolution of a base variant.

(Single Entry Single Exit) blocks with semantically equivalent code, converting simple calculations into time-consuming loops etc. may also be introduced.

3.3 Low-frequency morphing techniques

Here, the length of time between changes can be anything from a week to over a month. In low-frequency transformations it is the apply-key step (step 3 in Figure 3) that changes – i.e. the decryption algorithm changes semantically. The decryption algorithm is generally fairly simple.

Some examples of algorithms applied to encrypted data one dword at a time, are:

- dword + K
- (dword + K1) ^ K2
- rotate ((dword + K1), K2)
- a = RTC(dword, K1) -> 'modify carry flag' -> (RTC(a, K1) + K2) ^ K3 , RTC = rotate through carry
- (dword / K1) ^ K2

Once the obfuscations mentioned in section 3.2 are applied, it switches back to just changing the key in the resulting code for the next couple of days and applies the medium frequency transformation again. This cycle can continue for anything from a week to several months and then the low frequency transformation is applied.

Figure 5 is a pictorial representation of the evolutionary trend of Tibs executables. A base variant is encrypted using an algorithm (Section 3.3) to give an encrypted base variant (EBV). The different shapes of these variants represent the semantic non equivalence of Tibs decryptors. Thereafter, the transformations from Section 3.2 are applied to obtain mutants (M*). K represents a random key that is chosen for the mutant that gets released. The dotted lines represent virtual intermediary steps, while solid lines represent the mutants that are released. Different colours represent different mutants. The time line increases from left to right and the granularity of the high-frequency key change is approximated as one day.

4. TIBS DETECTION TREND

Figure 6 presents detection statistics for a randomly chosen set of 60 Tibs-packed samples obtained during a period of approximately one month. 26 static signature-based scanners were tested against the samples. Note that the scan result for each sample was obtained as soon as the sample was downloaded from the malicious host, with the latest scanner signatures available at that time.

Figure 6 presents the total, as well as accurate (signature-based) detection counts, where the total also includes heuristic detections.

On some of the days the detection rates were better than on others because there had been no significant change in the malware. In fact, there are hundreds of samples with different hashes that appear daily, yet Figure 6 represents a satisfactory test as there is no significant change in the

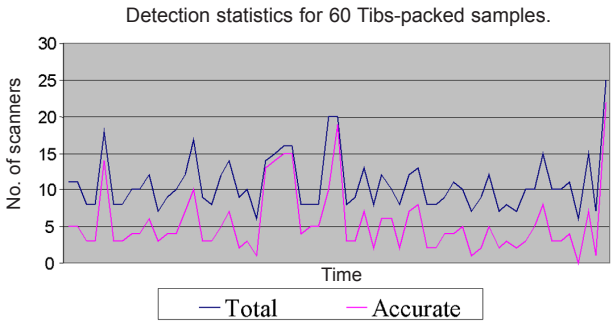


Figure 6: Total and accurate (signature-based) detection counts.

variants that appear within a day (Section 3). The detection rate in a day for most AV vendors is fairly static. It is evident from Figure 6 that the number of accurate detections is low when compared to heuristic detections. Furthermore, no AV vendor showed consistently accurate detection for all samples.

5. DETECTION CHALLENGES

For the hundreds of different samples generated every day as described in Section 3, a byte-based signature could be written that detects on the decryptor loop code itself (which remains the same except for the key value). However, this would not be effective for longer than a day or two because the code-morphing techniques described in Section 3.2 would be introduced to obfuscate the code and muddle the byte patterns.

To handle this mutation one could use emulation, which is a popular way of dealing with polymorphism. The loop could be emulated to decrypt the underlying data, and detection could be achieved based on that decrypted data. This may provide detection for a longer period of time, depending on how robust the emulator is. However, as mentioned in Section 3.2, Tibs introduces anti-emulation techniques along with obfuscation. For example, some emulators may not be able to handle MMX instructions. While emulators handle most common WAPI calls to facilitate sufficient emulation of code, handling all WAPI calls – i.e. figuring out the number of parameters that each takes and the return value(s) depending on the context of the call – becomes increasingly challenging.

In order to achieve accurate detection of Tibs-packed threats for longer periods of time, one could choose to base detection on attributes that change less frequently, such as the underlying code.

Understanding the decryptor logic and using better methods to decrypt could be one way to add generic detection. This could be achieved by using cryptanalysis on the encrypted code. Alternatively, a detection technique may choose not to decrypt and leverage the fact that the encryption is always one dword at a time by performing statistical analysis on the encrypted data. However, one of the major concerns for AV developers with such techniques is efficiency; the desktop scanner's speed should be acceptable to end-users and such cryptanalysis techniques tend to slow performance significantly.

Heuristics based on file geometry can also be used to detect on the overall structural commonalities of these executables. Attributes such as file size, number/names of sections, section flags, linker versions and unusual imports may serve as good aids in writing detections for these samples heuristically. The risk with such approaches, of course, is that false alerts may be produced on clean files.

The server-side aspect of this polymorphic approach creates the opportunity for blending automated sample generation with periodic human intervention, thus making such threats more insidious than their traditional counterparts.

With Tibs being a proprietary packer, it is tricky to guess how much of its polymorphic process is automated. In theory, a lot of it could be automated, but we do not know how much of it is in reality – this could be an interesting piece of research. The minimum requirement for any detection signature is that it should detect all samples that are generated automatically.

CONCLUSION

This article describes a trend in the evolutionary pattern of Tibs-packed malware and discusses various detection techniques and their pitfalls. The approach described in this article is not the only way in which the server-based malware model can work and this threat may change its tactics in future. There is room for improvement in both the attack and defence techniques and the bar will be raised on each side as this battle progresses.

The authors of Tibs are not the first to use server-based obfuscation techniques, but they are surely amongst the most successful with it. Other threats are likely to follow in its footsteps; we can expect a significant rise in the number of malware samples as the popularity of such techniques will almost certainly increase in the future.

REFERENCES

[1] Florio, E.; Ciubotariu, M. Peerbot: catch me if you can. Virus Bulletin, March 2007, p.6.

[2] Bureau, P.-M.; Lee, A. Malware storms: a global climate change. Virus Bulletin, November 2007, p.12.

[3] McAfee VIL. W32/Nuwar@MM, http://vil.nai.com/vil/content/v_140835.htm.

[4] Wikipedia. Storm Worm, http://en.wikipedia.org/wiki/Storm_Worm.

[5] Boldewin, F. Peacomm.C Cracking the nutshell. http://www.reconstructer.org/.

[6] Wikipedia. Tiny Encryption Algorithm. http://en.wikipedia.org/wiki/Tiny_Encryption_Algorithm.

FEATURE 2

EXEPACKER BLACKLISTING PART 2

Robert Neumann
VirusBuster, Hungary

In the first part of this article (see *VB*, October 2007, p.14) Gabor Szappanos presented a general overview of exepacker blacklisting and considered both the positive and negative aspects of the practice. In this, the second part of the article, we continue with more detailed information about the different types of blacklisting, and take a look at the tools that are available for use during analysis.

THE COLLECTION

As mentioned in part one of this article, we can divide executable packaging tools into four main categories. The categories separate the tools based upon their primary purpose and common behaviour:

- Compressors: the only goal of these tools is to decrease the size of the executable using either common or custom-made compression algorithms. The likelihood of them having any anti-debug-related code is usually close to zero. The most well known tools in this category are UPX, FSG, MEW, PECompact and Upack.
- Cryptors: this category covers packers which utilize simple encryption algorithms to make reverse engineering more difficult. They usually have basic anti-debugging code, but no compression. A few well known cryptors are Yoda Crypter, UPolyx and Morphine.
- Protectors: these are the 'big guns', combining multiple compression and encryption algorithms along with complex anti-debugging code, and sometimes even custom-made virtual machines. The most representative of this category are ASProtect, Armadillo, SVKP and Themida.
- Installers: this category is somewhat different from the other three – these are applications that are capable of creating self-installing packages. We decided to include these applications in a separate category since we are seeing a fair amount of malware using them now. NSIS, Inno Setup and Wise Setup are the most common.

A little over two years ago we realized that there was a need for some kind of united effort among AV researchers to help each other deal with different types of packers, so a mailing list and a collection of known packers was born. The idea was welcomed within the AV community and the mailing list has been growing steadily ever since. Meanwhile, the collection of packers has grown to cover around 95% of the known packers (both public and non-public). The current collection can be broken down into the various categories as follows (also see Figure 1):

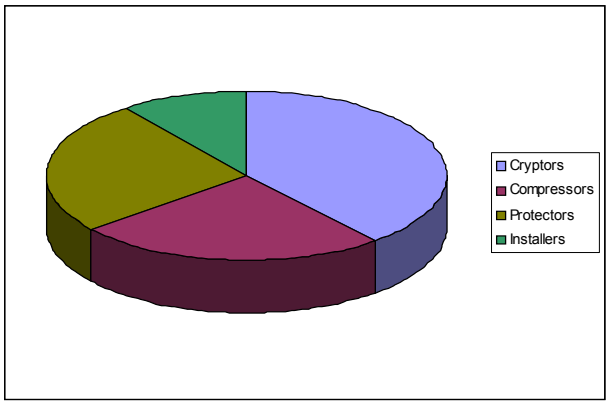


Figure1: The exepacker collection.

- 76 different cryptors, 172 versions in total
- 49 different compressors, 305 versions in total
- 50 different protectors, 291 versions in total
- 20 different installers, 198 versions in total

Overall this means almost 200 different applications and close to a thousand different versions. Only the Win32 packers are counted here, since DOS-based malware no longer forms part of our daily work. Just for the love of the numbers the entire collection consists of 264 different packers totalling 1,195 different versions – and it grows by around 6–8% every month.

COMMON METHODS

Nowadays it is hard to believe that any anti-virus product could survive without some kind of exepacker support. Most product developers aim for combined solutions such as native unpacking together with the use of a powerful emulator. As there are different goals to achieve, there are different approaches and solutions for each.

- Native packer support: this method is the most powerful, but also the most time consuming. An analyst has to fully reverse engineer the given packer, mapping all the compression and encryption algorithms in the usual massive amount of assembly code, then rewriting them in a high-level programming language. Once this has been done it will be very easy to unpack the specific packer (at least until the next version appears) and a working unpacked executable can be obtained

within seconds. There is a downside though: malware authors try to trick the native recognizers of AV engines through trivial or non-trivial modifications (e.g. PE-Patch, various UPX cryptors, fake section names etc.) and even slight modifications can render the unpacking process impossible. Some packers are open source (e.g. UPX, PeX, Morphine), so altering them is an easy task, as has been observed before (e.g. PeX/Bagle).

- Emulator-based unpacking: creating a powerful emulator can save a lot of research time – it is not a quick or easy process, but the benefits will be enjoyed in the long run. Once an emulator has been created we no longer have to worry about each new packer version and very few of the small custom-made ones will pose any problem. However, as with every good thing there is a downside: emulator-based unpacking comes at the cost of performance since emulating through packer code is much slower than unpacking the same with native support.
- Hybrid solutions: to combine the best of two worlds, namely native support and emulator-based unpacking, some AV vendors have come up with hybrid solutions [1]. In this case the common compression and encryption algorithms are supported by native code and a specific emulator is used with the support of a custom script-like language. These scripts control the whole unpacking process by utilizing both native code and the emulator, whenever they are needed. This method requires a lot less CPU time compared to the emulator-only unpacking, yet it is very flexible and easily expandable.
- Simple blacklisting: blacklisting is probably the easiest solution. Whenever we decide that a packer should be blacklisted, one generic detection is enough to make it happen. It is not time consuming by any means and new versions of the same packer can be added very quickly.

TOOLS FOR USE

Whichever form of exepacker support is used, researchers will need the help of a couple of different tools for sample analysis.

At the outset we have no information as to whether a malware sample is packed. A well trained eye using a simple hex editor is usually able to judge if a file is packed, but it takes quite a long time to gain such experience. Otherwise, the most common way to discern whether a sample is packed is by checking the file's entropy. *PEiD* is a handy tool capable of calculating the entropy of a given executable (alongside many other useful details). Of course it has an internal database of known executables, but let's look at an approach for dealing with an unknown and possibly new packer.

If the calculated entropy indicates that our executable is packed, then we need to take a closer look with the help of a disassembler or debugger.

Processing the sample with the *IDA* disassembler will give us enough information to be able to judge whether we need to look for a debugger instead, or whether the deadlist along with *IDA*'s features are sufficient to complete the task. Static disassembly is usually sufficient if we are facing a packer from the cryptor or compressor category, however it quickly becomes more of a pain once the sample has multiple layers of compression and/or encryption.

At this point our second best friend is *VMware* (or any similar virtual machine) – unless we happen to have an additional dedicated computer that is isolated from the network such that a possible outbreak won't affect it. Since there are many different ways to detect the use of virtual environments, we must either try to prevent that happening (e.g. by tweaking *VMware*'s config file) or get ourselves another PC which can be sacrificed at the altar of science [2]. Regardless of this, we will certainly need a debugger to be able to trace through thousands of lines of packer code before finally reaching the original entry point of our executable.

Unless for some reason we need a ring 0 debugger, the slightly outdated *OllyDbg* is the most well suited for the task. It is quite a powerful ring 3 debugger on its own, but when used with the excellent user-made plug-ins such as *OllyScript*, *OllyDump* or *Olly Advanced*, it can be extended to an even greater level. There are other ring 3 debuggers such as the newcomer *Immunity* and the aging *TRW*, but overall we consider *OllyDbg* to be the best choice for this task.

In case *OllyDbg* doesn't suit our needs and we are desperately searching for a ring 0 debugger, the options available are rather frustrating. A few years ago we would have recommended *SoftIce* within the blink of an eye, but unfortunately support was dropped for the kernel debugger last year. However, we can still use *SoftIce* up to *Windows XP SP2*, and unless vast amounts of *Vista*-specific ring 0 malware appear on the horizon in the near future, we shouldn't be too worried. Beside *SoftIce* there are other options available for kernel debugging. We can either use the not-so-pretty, but still quite decent tool *Windbg* (with two computers or connecting it to a *VMware* client through a named pipe) or the *SoftIce* heritage-like *Syser* debugger.

Once we arrive at the original entry point of the executable we can consider our task to be complete, at least for now. A

proper memory dump (and, depending on the feature set of the packer, rebuild of the import table, restoration of stolen code parts and so on), combined with static analysis and generic detection is usually enough to determine what's inside, but that's beyond the scope of this article.

Here is a quick overview of the tools we should have in our arsenal:

- Hex editors: *Hiew* or *PE Explorer*, depending on whether one prefers console or *Windows*-based applications (both have a built-in disassembler).
- Disassemblers: an outstanding product, *IDA* is unquestionably our recommendation.
- Debuggers: *OllyDbg*, *SoftIce*, *Windbg* – the choice is really up to personal preference.
- Other tools: *PEiD* and *RDG Packer Detector* are must-haves for known packer detection.

MAKING OUR LIVES EASIER

The volume of daily incoming malware has reached such a high level that processing and unpacking each and every piece of packed malware manually is no longer possible – and would be a waste of precious human resources. With an average 44% of the total number of incoming samples having some kind of packer on them [3], about 20% of which can easily be unpacked by native support (the likes of UPX, FSG, Upack etc.), the remaining 20–24% still gives us a run for our money.

To be able to utilize further blacklisting, we need to separate the samples according to packer type. Once we have a handy list of the most common packers we can decide what kind of support to plan for them. For the purposes of gathering such information the use of *PEiD* and *RDG* alone might not be enough. First they are GUI-oriented applications which makes automation a bit of a complicated task, and second they are not updated on a regular basis (more like yearly in the case of *PEiD*).

On close examination it turns out that both tools basically work with large collections of packer-specific sequences, along with some advanced detection methods. The key is the *sequences* – it's quite a simple task to collect a few external *PEiD* databases, sort out the duplicated detections, remove the junk and merge our own custom sequences into it. Now we can code our own packer detector which will fully suit our needs, and can be run through large collections of malware to gather accurate statistics. (Note: Metasploit Framework has a built-in packer detector using an external *PEiD* database.)

I'm sure that most of the big AV vendors are taking advantage of automated sample processing systems by now – we certainly are. Since human resources are limited almost everywhere, it was an obvious step to automate some tasks in order to free up highly valuable researcher time. Combining the automated systems with the above-mentioned packer detector gives us the opportunity to do whatever we want with a specific packer.

As stated in the first part of the article, the main problem with blacklisting is that we cannot simply blacklist all packers on arrival. Our current approach is to blacklist all the 'pure' black ones, as they will never be found on anything but malware. The middle or 'grey' category is always going to involve some kind of risk management due to the small – but significant – number of potential false positives. Having a powerful Win32 emulator can be the solution here.

We don't touch the white packers for obvious reasons – most of them are commercial products intended to protect other shareware applications, and it is just unfortunate when malware authors use them. Aiming for native support is the only negotiable – and most of the time pretty rough – way to handle this category.

A LOOK INTO THE CRYSTAL BALL

Exepackers are here to stay for a long while; their roots go back to the shady days of DOS and their future is yet to be seen.

Taking a look at the global picture clearly shows a few things: commercial software developers are stuck right now, with new and ground-breaking ideas sparse on the horizon. The heavyweights of the past years such as ASProtect or Armadillo have entered into a state close to hibernation. An update appears for them once in a while, but these are generally only bug fixes. The only active (and promising) members of this category right now are Themida, the successor of the old Xtreme-Protector, and VMProtect. Both of these feature a built-in custom virtual machine.

One should never paint the devil onto the wall, as the old saying goes, but I for one wouldn't like the idea of dealing with new families of malware where one of the previously mentioned virtual machines is properly implemented into the code. Do we know that this is going to happen? We can be fairly confident since our experience shows that whenever a new professional product gets out of the door and a legal (or more likely illegal) copy finds its way to the various RCE (Reverse Code Engineering) and AV-related boards, we can expect malware to be packed with it within a few days. It's an unfortunate situation where both parties take advantage of the very same sources. We can only be thankful that these products haven't yet been used maliciously to their full potential.

Vista has also put a new twist on things with its new driver policy: we can say goodbye to ring 0-based solutions, as everything is returning to ring 3 once again.

Whether as a result of the above or for other reasons, malware authors nowadays more often develop and maintain their own custom packers. This gives them the opportunity to alter the source whenever they want to, which is a powerful option for them in the fight against AV products.

Since size doesn't really matter any more, as most of the world has entered into the age of broadband Internet, the possibility of new basic compressors suddenly appearing is rather low. Upack was the last real crusader in this area, and it is pretty dead (unless we count the ever-growing number of PECompact betas).

The only real live and growing category of packers remains the pure black ones. We can expect new black packers to continue to appear from time to time, as creating a small cryptor isn't really time consuming and an as-yet-unknown packer will be always capable of hiding malware for a couple of days, or under very extreme circumstances, weeks. However, this behaviour is their main weakness as well: since these packers will never be used on operating system files, not even on shareware applications alone, we won't have to think twice about blacklisting them.

...AND ALL THIS IN ACTION

Talking about different tools and approaches in theory is like explaining to someone how it feels to ride a bicycle without letting him try. If we want to work quickly and efficiently, we always have to be capable of making quick decisions about what tool or application best suits the current situation. Knowing this alone is only a part of the success, mastering their usage to a level where it feels like second nature is another. *Hiew*, *OllyDbg* and *SoftIce* are all very powerful tools on their own, although selecting them for the right task is sometimes more complicated than it would seem.

In the third and closing part of the article we will look at how all this works in reality.

REFERENCES

[1] Svajcer, V.; Mody, S. Unpacking – a hybrid approach. EICAR Conference, May 2006, Hamburg.

[2] Ferrie. P. Attacks on virtual machines. AVAR Conference, December 2006, Auckland.

[3] Lu, B. A deeper look at malware – the whole story. Virus Bulletin Conference, September 2007, Vienna.

FEATURE 3

BLOW UP YOUR VIDEO

Christoph Alme, Dennis Elser
Secure Computing Corporation, Germany

In these days of feature-rich online portals for video and audio files, we are used to seeing interactive multimedia content on websites. Associated file formats are usually perceived by end-users to be trustworthy, with the users expecting them to contain only video and audio content.

Unfortunately it is possible for these file formats to contain more than one might expect. Lately, they have been misused frequently as building blocks in web-centric attack vectors, mostly in combination with cross-site-scripting vulnerabilities as described in [1].

This article presents a round-up of recent multimedia vulnerabilities, looking at today's well known formats for interactive media – *Adobe*'s *Flash* and *Apple*'s *QuickTime* – as well as peeking at *Microsoft*'s upcoming web presentation technology, *Silverlight*.

FLASH AND ACTIONSCRIPT

A recent example of a malicious media file [2] demonstrated the transition of a known *Windows* malware behaviour to the *Flash* 'platform'. In a similar manner to a *Windows* executable that detects the presence of a virtual machine and behaves benignly in that environment, the malicious *Flash* file determines whether it is running on a known back-end system that is intended to analyse the malicious impact of *Flash*-based advertising banners, and does not launch its payload if that is the case.

This is possible because *Flash* comes with its own proprietary scripting functionality, a language called ActionScript, and determining the domain that hosts a *Flash* file is as easy as reading an ActionScript property called 'domain'.

Similarly, after injecting a hidden IFRAME element into the hosting website that bears the malicious code, the attack does not run the exploit blindly, but rather checks whether the MS07-009 patch (addressing a vulnerability in an MDAC ActiveX Control) is installed. Only if the check reveals a vulnerable system is the exploit run. The check is as follows:

```
var c = new ActiveXObject
            ("ADODB.Connection");
if (c.Version == "2.7") {
      // ...
}
```

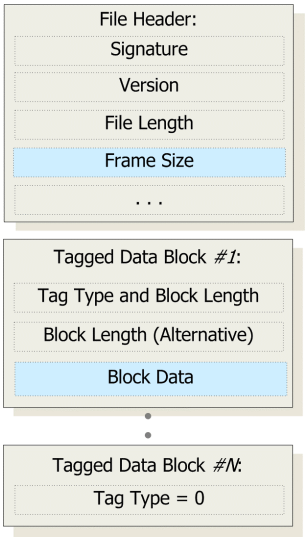
Needless to say, disabling the unspeakable ActiveX functionality (if not using an alternative browser) is advisable.

Rather than going into the details of the plethora of multimedia-related functionality available in ActionScript, we'll focus on one example. As of ActionScript version 2 (from 2004) – which is compliant with the ECMAScript 4 specification – one can invoke any JavaScript function available to the hosting HTML document (e.g. to the *Flash Player* 'container') using an object called 'ExternalInterface'. Wisely, this action is by default (as of player version 8) restricted to media hosted on the same domain as the embedding website. To get the URL of the document embedding a *Flash* file, for example, one can call JavaScript from ActionScript as follows:

```
private var url : String =
      ExternalInterface.call (
            "eval",
            "document.location.href" );
```

This allows powerful interactivity between the JavaScript of a website and an embedded *Flash* object. On the downside, generally speaking it provides another obfuscation layer, allowing malicious JavaScript code to be moved into harder-to-parse *Flash* files.

'SWF' (from the former product name *Shockwave Flash*) is a binary file format. It starts with a file header, whose first field contains the magic bytes of either 'FWS' or 'CWS' – the latter identifying the file as being deflate-compressed. Next comes the file format version (one byte), then an unsigned 32-bit field specifying the total file size in bytes, and next the variable-length 'FrameSize' field.



The file header is followed by an arbitrary number of so-called 'Tagged data blocks'. To find the offset of the first data block, one has to calculate the actual length of the FrameSize field, in bits, as:

```
5 + (((FrameSize[0] &
0xF8) >> 3) << 2)
```

This must then be translated into the respective number of bytes, rounding up by a byte if the number of bits is not divisible by eight, and adding 12 bytes for the other fixed-size fields of the file header.

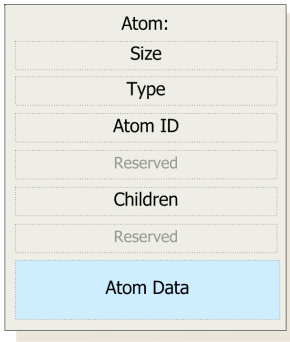
The data blocks each start with a 16-bit field, the upper 10 bits specifying the tag type and the lower six bits the length of the block. If the length is 3Fh, another 32-bit field follows which specifies the block's length. An SWF file ends with a special 'EndTag' block (tag type 0).

Of interest to us is the 'DoAction' block (tag type 0Ch), as it is one of the block types that can contain ActionScript bytecode. Its data is a list of instructions, each called an 'Action' and starting with an eight-bit opcode (called 'ActionCode'). For example, there's an 'ActionGetURL' instruction (opcode 83h) that allows a URL to be opened in a desired target frame – including the current browser window. The length of both the URL and the name of the target frame follows in a 16-bit field behind the instruction opcode, and then comes the URL and the name of the target frame. Both are zero-terminated strings, encoded in UTF-8 (or ANSI in older versions).

QUICKTIME MOVIES AND LINK FILES

QuickTime is another popular format for multimedia, and since it is bundled with *Apple*'s *iTunes* software, it is installed on many end-user PCs. Yet, it's not the scripting-driven, 'interactive' kind of format that *Flash* (for example) is – or is it?

In December last year, a worm based on a *QuickTime* movie file spread by infecting *MySpace* user sites with a link to itself, and using the so-called 'HREF track' to embed JavaScript code into the *QuickTime* movie. URLs (and scripts) in HREF tracks were followed automatically based on elapsed playback time, without user interaction. The JavaScript in turn exploited a cross-site-scripting vulnerability to infect the visiting user's *MySpace* site.



The file format does not start with a file header at a fixed offset, rather the whole file consists of so-called 'atoms'. The presence of an atom called 'movie atom' is mandatory; its type value is 6D6F6F76h ('moov'). Each atom starts with a header, followed by atom-specific data. The size and type fields are each 32-bit, stored in big-endian order (which is the default in this format). The size refers to the size of the whole atom, including its header, in bytes. If it is set to one, an optional unsigned 64-bit field follows behind the type field that contains the actual size.

An atom can contain other atoms as children, allowing hierarchical storage in movie files. Apart from that, atoms can be stored in (almost) any order. *Apple* recommends certain ordering of atoms, and files that adhere to it can be played while they are being downloaded from the Internet.

About three months after the incident, *Apple* fixed the vulnerability by removing support for embedded JavaScript

– although this was an intended feature, it was probably not a required one.

Just half a year later, something similar seems to apply to scripting 'capabilities' discovered in the *QuickTime* player link file. In September [3] this feature proved to allow privilege escalation when combined with *Firefox*'s 'chrome' URL protocol.

A *QuickTime* player link file is basically an XML document. It can contain exactly one <embed> element (those other than the first are ignored), which should point to a multimedia file's URI. This element's 'qtnext' attribute could be tampered with in order to execute JavaScript code with maximum privileges – allowing, for example, the execution of arbitrary executables.

The extension name of these files, which should be '.QTL', seems to be meaningless. Renaming it to '.MP3' and other *QuickTime*-supported file formats is not only possible, but it even removes the last line of defence: *Firefox*'s 'Open With' dialog.

Probably in an attempt to make thorough, yet generic detection especially challenging (apologies for the sarcasm), the 'qtnext' XML attribute can have many names – up to 256 in fact:

```
<?xml version="1.0">
<?quicktime type="application/x-quicktime-media-
link"?>
<embed src="a.mov" autoplay="true"
qtnext3="" qtnext4="" qtnext5=""
qtnext29="... malicious code here ..." />
```

The *Mozilla* team was (once again) quick in providing a *Firefox* update that closed this vulnerability. And with the associated *QuickTime* update released by *Apple* three weeks later, not only had the actual vulnerability been fixed, but the whole scripting 'feature' seems to have been removed. Flexibility and susceptibility go together hand in hand.

INTRODUCING SILVERLIGHT

Microsoft's web presentation platform *Silverlight* [4], which is assumed to be the company's answer to rival *Adobe*'s *Flash* format, has just been released. While its first version 'only' allows the use of JavaScript and VBScript for interactivity within *Silverlight*, the upcoming version (which is already available in Alpha format) will add support for the .NET platform. The available .NET functionality is accommodated to the web browser context though as, for example, classes like 'System.Web.HttpCookie' are not accessible from within *Silverlight*.

To make use of *Silverlight*, an embedder has to invoke a script function such as 'createSilverlight()' first, passing an 'Extensible Application Markup Language' (XAML) document to be rendered by the plugin. With *Silverlight 1.1*, the document's 'Canvas' element can further include a 'Class' attribute in order to reference a .NET managed code assembly that implements event handlers:

```
<Canvas ...
      xmlns:x="http://.../winfx/2006/xaml"
      x:Class="MyNamespace.MyClass;
                  assembly=MyAssembly.dll"
            ...>
```

The assembly would implement event handlers, such as for the 'Loaded' event (which fires just before the loaded content is rendered):

```
namespace MyNamespace {
    public partial class MyPage : Canvas {
      public void MyPage_Loaded (object o,
EventArgs e) {
        if (!HtmlPage.DocumentUri.ToString().Contains
("ad-verification-domain-here.com")) {
          // ...
        }
      }
    }
}
```

You might think that *Silverlight* is limited to the '*Windows*-with-*Internet Explorer*' platform, but hold on: the *Silverlight* browser plugin is already available for *Firefox* as well, and that includes the above-mentioned .NET support. *Mac OS X* with either *Firefox* or *Safari* is supported as well, and support for *Linux* will be realized together with *Novell*, based on the 'Mono' project (a cross-platform .NET implementation, that is said to be binary compatible to *Microsoft*'s IL bytecode).

CONCLUSION

The file formats that we have covered briefly here have been shown to be susceptible to cross-site-scripting on a case-by-case basis throughout the last couple of months, and more generally, they allow malicious code to become harder to find. While they cannot be said to be less secure than, say, HTML, the opposite (being more secure) is certainly not the case.

REFERENCES

[1] Picture theft through hole in Google's Picasa. http://www.heise-security.co.uk/news/96554.

[2] Yahoo feeds Trojan-laced ads to MySpace and PhotoBucket users. http://www.theregister.co.uk/2007/09/11/yahoo_serves_12million_malware_ads.

[3] Apple QuickTime Player Zero-Day Vulnerability. http://www.securecomputing.com/SWAT/BlogEntries/SecureBlog_QuicktimePlayer.html.

[4] Microsoft Silverlight – Light Up the Web. http://www.microsoft.com/silverlight/.

COMPARATIVE REVIEW

WINDOWS 2000 PROFESSIONAL

John Hawes

Windows 2000 is getting a little long in the tooth, having been superseded within two years of its release by *Windows XP* – whose slightly shinier surfaces seemed so revolutionary back in 2001 – and this year by the even shinier *Vista*. Despite its age and rather drab looks, Win2k soldiers gamely on, serving its purpose perfectly adequately for plenty of users and still being the operating system of choice in many homes and businesses.

For the developers of security products this represents something of a challenge. New platform versions will inevitably present plenty of new hurdles, with tweaks needed to various parts of the products, not least the interfaces to keep pace with the ever-improving look and feel of computer desktops. But while all this newness is being added there is also a duty for developers to keep in touch with the old.

While many (but by no means all) security vendors, including *Microsoft* itself, have retired support for the Win9x family, *Windows 2000* (currently held in an 'extended support' period by *Microsoft*) remains too big a market to drop, and its close proximity to current market leader *Windows XP* has meant that, in most cases, little extra work is needed to ensure mutual compatibility. Of course, with most development and QA eyes firmly on the more common, more recent platforms, bugs and troubles on older versions are more likely.

However, with yet another bumper crop of products to slog through in a somewhat short month, I hoped that the products would prove as stable, reliable and trouble free as the platform itself.

PLATFORM AND TEST SETS

Windows 2000 has been sitting on Service Pack 4 for several years, and as usual with VB100 tests the platform was used in a fairly bare state with no further updates added unless required by a specific product.

The installation and setup of *Windows 2000* was thus a fairly straightforward task, familiar from countless previous ventures down the same path, and complicated only by a lack of support for some components in the fairly new machines preferred for VB100 testing. Rather than face several weeks testing at low resolution, extra drivers were added to fully enable the modern graphics cards, as well as network interfaces, but otherwise the systems were left untouched. I expected some products to require updates, such as upgrading *Internet Explorer* or *Windows Installer* to more recent versions, but these changes were not made by default in order to ensure that products with such requirements could easily be identified.

The test sets were based on the most recent WildList available on 26 October, with the product submission deadline a few days later. This month, a spurt of hard work from the *WildList Organization* meant that the September WildList was available in plenty of time to be included, and it was upon this list that the main test set was based.

With a large number of new additions by recent standards, replicating and validating samples for the set was a bigger job than usual, but helped by the preponderance of familiar old names: large numbers of W32/Rbot and W32/Sdbot, with plenty of W32/Agobot and W32/Rontokbro and other similar items. There were a few less common additions, including plenty of file infectors, mainly from the W32/Looked and W32/Fujacks families, but including a W32/Virut variant which promised to present significant challenges in detection.

Also of note was the fact that, for the first time in a while, there was not a single new W32/Mytob variant to be added – a sign, perhaps, that this family is finally showing its age.

With a lot of lab time taken up with additions to the core set, expansion of the other test sets was limited. A sprinkling of items were added to the collection of worms and bots (mostly yet more variants of the major families) and the existing polymorphic test sets were expanded.

The clean set was enlarged with the usual selection of items, mostly from popular and recently released software packages on common download sites.

To assist in the presentation of speed results a small new set of files was added. With products offering some wildly different sets of default settings, the archive test has long presented problems when showing speed measurements, with products that do not scan inside archives unfairly showing better speeds than their more thorough rivals. To guide readers in interpreting these results, a set of common archive types has been created at various depths of nesting, with the Eicar test file at the bottom of each. A plain, uncompressed copy of the test file was added to check that it was indeed included in the detection, and as an extra, another copy renamed to a random extension was added to test scanning of non-standard filenames.

I created a rather arbitrary cut-off point, deciding that products should detect at least five levels deep in at least four of the eight archive types included in the set in order to be included in the 'all files' speed graphs, and below this level a product's scan times would only be included on the 'default settings' display.

AEC Trustport Antivirus 2.8.0.1607

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	99.94%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	15

Czech Republic-based *AEC* has been doing pretty well with its *Trustport* product, achieving some impressive scores in numerous tests thanks to its multi-engine approach. The product submitted here, the anti-virus component, is not available as a standalone product but is part of the *Trustport Workstation* suite, along with a swathe of other security solutions, and is rolled into a range of server and gateway products.

Installation of the product hit an immediate, if not unexpected stumbling block in the form of the requirement for *Internet Explorer 6* or newer. While this is not an extravagant demand, it does raise a small concern – it's more than possible that a user, having restored a system to an old safe state (perhaps using a rescue CD provided by the system retailer), would be in the position of running a bare *Windows 2000* installation, and would thus have to spend quite some time online in an entirely unprotected state to acquire the required updates. Given the scare stories that estimate the average infection time for an unprotected system connected to the web to be as little as ten minutes, this window of exposure could be unacceptable.

Once installed, *Trustport* presents a solid and reliable appearance with its graphics depicting well shielded footsoldiers – an image backed up by the multi-engine scanner at its heart. The product's makeup has changed somewhat since its last appearance, with the *BitDefender* engine included in earlier versions replaced by those of *Dr.Web* and *VirusBlokAda* – an interesting selection, not least as it includes an engine which has yet to appear on the *VB* test bench. A lot of heuristic technology hinted at a high risk of false positives, but could be expected to ensure pretty thorough coverage of infected items.

Tests were carried out easily, with the speed tests particularly straightforward as the default action is to scan all files, including the contents of archives, both on demand and on access. The new set of archive types was detected in depth, although neither of the engines implemented on access seemed capable of penetrating .LZH files – the on-access mode uses only two of the available scanning engines, though more can be added by the more paranoid user as long as they have the available processing power. Of course, multiple engines are unlikely to achieve the best speeds or lowest overheads, and speed figures here showed a pretty hefty drain on resources.

The many engines spotted a fairly large number of potentially unwanted items in the clean sets, a large number of which were system tools from *Sysinternals*, and all of which were labelled in the log with the rather stark and worrying 'Infected!'. However, their full definitions described them more accurately as tools or programs. As feared a few full false positives were also flagged, spoiling the product's chances of winning another VB100 award. More surprisingly, a few samples of the new W32/Virut variant were missed on access, indicating that these were likely to prove a problem for at least a few more products as testing continued.

Agnitum Outpost Security Suite Pro 6.0.2165.8226

ItW	100.00%	**Worms & bots**	99.74%
ItW (o/a)	100.00%	**DOS**	99.58%
File infector	98.86%	**Macro**	100.00%
Polymorphic	84.18%	**False positives**	0

Agnitum's product is fairly recent and almost certainly developed since the arrival of *Windows XP*. It showed no signs of requiring any extra software – at least until halfway through the installation, when an error message revealed the absence of a required DLL. This did not seem to be a fatal problem, and the installation continued to the requested reboot, on return from which the system froze in an unresponsive state.

Reimaging and trying the installation again with the extra DLL in place led to a much more complex installation process, with a series of configuration pages to be worked through before reaching the reboot phase. Again, the system failed to return – even safe mode seemingly inaccessible – and the developers were called for assistance. Investigation indicated that the problem related to the rather modern systems being used for the test, and when the test image was ported to more humble hardware there were no such difficulties.

With no clear way of circumventing the problems on the main systems, tests proceeded minus the speed test, which would have been all but meaningless on the considerably slower hardware.

The product looked good and proved pleasant to work with, offering a wide range of modules which sadly went unexplored. With good detection across the test sets and no false positives generated in the clean sets, *Agnitum* earns a VB100 award.

On-demand tests	ItW		Worms & bots		DOS		File infector		Macro		Polymorphic		Clean set	
	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	False positives	Susp.
AEC Trustport Antivirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	15	19
Agnitum Outpost	0	100.00%	3	99.74%	28	99.58%	8	98.86%	0	100.00%	220	84.18%		
Alwil avast!	0	100.00%	7	99.69%	757	97.74%	0	100.00%	18	99.56%	657	85.69%	1	
Avira AntiVir	0	100.00%	0	100.00%	32	99.79%	0	100.00%	0	100.00%	3	99.85%	2	
BitDefender AntiVirus	0	100.00%	1	99.84%	8	99.79%	2	98.48%	1	99.98%	0	100.00%		
Bullguard Bullguard	0	100.00%	1	99.84%	8	99.79%	2	98.48%	1	99.98%	0	100.00%		1
CA Antivirus	20	99.18%	0	100.00%	235	99.70%	1	99.77%	0	100.00%	9	99.60%		
CA eTrust	0	100.00%	0	100.00%	235	99.70%	3	99.02%	12	99.82%	9	99.60%		
Doctor Web Dr. Web	11	98.50%	1	99.84%	0	100.00%	2	99.24%	0	100.00%	0	100.00%	2	2
ESET NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.95%		
Fortinet Forticlient	2	99.98%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	2	99.90%		
Frisk F-PROT	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.95%	1	
F-Secure Anti-Virus 2008	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.91%		2
GDATA Anti-virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.91%		4
Grisoft AVG	0	100.00%	5	99.86%	200	98.96%	7	97.73%	0	100.00%	695	76.07%		
Ikarus Virus Utilities	9	99.88%	6	99.81%	2461	91.37%	23	93.37%	171	96.07%	353	80.58%	13	31
Iolo Antivirus	32	99.71%	1	99.84%	0	100.00%	0	100.00%	0	100.00%	4	99.83%		
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.91%		
Kingsoft AntiVirus	60	95.63%	600	18.23%	14022	13.56%	96	74.05%	463	90.97%	1634	31.32%		
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Microsoft Forefront	0	100.00%	1	99.84%	0	100.00%	1	99.86%	0	100.00%	80	96.05%		
MWTI eScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.91%		3
Norman Virus Control	7	99.94%	0	100.00%	269	99.29%	9	98.48%	0	100.00%	710	82.17%	3	
PCTools Anti-Virus	0	100.00%	2	99.89%	22	99.58%	8	98.86%	0	100.00%	221	84.99%		
PCTools Spyware Doctor	0	100.00%	2	99.89%	42	99.78%	8	98.86%	3	99.93%	220	85.05%	1	
Quick Heal Quick Heal	0	100.00%	0	100.00%	1035	95.18%	17	96.59%	73	98.23%	1130	73.04%		
Redstone Redprotect	1	99.86%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.91%		1
Rising Antivirus	1	99.97%	6	99.44%	10993	41.26%	51	90.30%	1273	69.32%	1327	46.17%	2	
Sophos Anti-Virus	4	99.96%	0	100.00%	0	100.00%	0	100.00%	8	99.80%	8	99.61%		3
Symantec Endpoint Protection	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Trend Micro OfficeScan	2	99.98%	3	99.89%	749	98.16%	9	98.67%	0	100.00%	738	84.88%		
VirusBuster VirusBuster	0	100.00%	2	99.89%	20	99.79%	8	98.86%	0	100.00%	220	85.05%		

Alwil avast! Professional 4.7.1075

ItW	100.00%	**Worms & bots**	99.69%
ItW (o/a)	100.00%	**DOS**	97.74%
File infector	100.00%	**Macro**	99.56%
Polymorphic	85.69%	**False positives**	1

Alwil's product is one of the more dependable regulars in *VB*'s tests, and while the interface is far from my favourite,

its intricacies no longer cause too many difficulties. Some admirably solid results were achieved on scanning the new archive set, with neither the archived nor the renamed copies of the Eicar test file spotted in the default modes, but everything detected with the archive and 'all files' settings switched on.

Speeds on demand were good, although on-access times were harder to come by – the product does not check files on simple opening, and on-access results for the infected sets were taken by copying the collection to the system across the network.

Results were pretty much as expected for *avast!*, with some older items missed but little from the more up-to-the-minute sets. Full coverage of the WildList was achieved, but hopes of a VB100 award were dashed by a single false positive in the clean set.

Avira AntiVir Windows Workstation 7.06.509

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	99.79%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.85%	**False positives**	2

AntiVir is another solid performer in *VB*'s comparative testing, with an excellent history both in detection and speed, and it did well again here.

The product is pleasingly laid out and simple to use, with the installer especially rapid and problem-free, and the tests zipped along at a similarly impressive rate. The archive sets were covered fully by default on demand, and almost so on access, with the rather odd exception of a few files in the .ACE format – while most were spotted, including the deepest nested to 10 levels, levels 3, 5 and 8 were missed.

Infected items were covered pretty well, with only a small number of polymorphic samples of rather rare and obscure variants missed. With the WildList test set covered in full, including those pesky Virut samples, only false positives could stop *Avira* claiming another award, and unluckily, two files were indeed erroneously flagged as infected, denying *Avira* the chance to add to its collection of VB100 awards this month.

BitDefender AntiVirus 2008

ItW	100.00%	**Worms & bots**	99.84%
ItW (o/a)	100.00%	**DOS**	99.79%
File infector	98.48%	**Macro**	99.98%
Polymorphic	100.00%	**False positives**	0

The *BitDefender* product stated that a better version of the *Windows Installer* was needed to install it – but as a pleasant surprise it also informed me that it had a copy handy and would install it for me. The pleasurable moment soon passed though, when after a reboot and a second attempt at installing, it was found that *IE6* would also be needed and on that count I would have to fend for myself.

I had also been informed that *Update Rollup 1* was required for the product to function – but a quick check without this generated no warnings from the product, and left the on-access functionality crippled, despite a comforting green tick insisting that all protection was active.

After several reboots therefore, I was finally able to get to work, and initial scans proceeded quite happily, with no false positives spotted on demand and most of the archive types detected easily, although .TGZ and self-extracting zips were only delved into to a depth of eight levels.

Scanning of the infected sets proved simple and highly successful, with a tiny number of misses and no false positives, thus earning *BitDefender* another VB100 award.

Bullguard Bullguard 8.0-32bit

ItW	100.00%	**Worms & bots**	99.84%
ItW (o/a)	100.00%	**DOS**	99.79%
File infector	98.48%	**Macro**	99.98%
Polymorphic	100.00%	**False positives**	0

Installing *Bullguard* confirmed a suspicion I had had all along – that the requirement for upgrades to *Internet Explorer* (already made by a few products and likely to crop up at least a few more times before I was done) is purely for cosmetic reasons. *Bullguard* has no such dependency, and installed smoothly on the bare system with no need for any extra work on my part.

The user experience was not adversely affected by the lack of modern display technology, and the tests proceeded nicely, recording similar times and detection rates to *BitDefender*, whose engine the product is based on.

The archive results were likewise the same, with .TGZ and self-extractors limited to eight levels but everything else covered. With admirable detection rates – missing barely a handful of samples per set, none of which were in the WildList set – and no false positives, *Bullguard* earns its second VB100.

On-access tests	ItW		Worms & bots		DOS		File infector		Macro		Polymorphic		Clean set	
	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	No. missed	%	False positives	Susp.
AEC Trustport Antivirus	7	99.94%	0	100.00%	92	99.59%	2	99.24%	0	100.00%	553	89.53%		
Agnitum Outpost	0	100.00%	3	99.74%	28	99.58%	10	98.11%	0	100.00%	220	84.18%		
Alwil avast!	0	100.00%	7	99.69%	757	97.74%	0	100.00%	18	99.56%	657	85.69%	1	
Avira AntiVir	0	100.00%	0	100.00%	32	99.79%	0	100.00%	0	100.00%	3	99.85%	2	
BitDefender AntiVirus	0	100.00%	1	99.84%	8	99.79%	2	98.48%	2	99.96%	0	100.00%		
Bullguard Bullguard	0	100.00%	1	99.84%	8	99.79%	2	98.48%	1	99.98%	0	100.00%		
CA Antivirus	20	99.18%	0	100.00%	235	99.70%	3	99.02%	0	100.00%	9	99.60%		
CA eTrust	0	100.00%	0	100.00%	235	99.70%	3	99.02%	12	99.82%	9	99.60%		
Doctor Web Dr.Web	11	98.50%	1	99.84%	0	100.00%	2	99.24%	0	100.00%	0	100.00%	2	
ESET NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.95%		
Fortinet Forticlient	2	99.98%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	2	99.90%		
Frisk F-PROT	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.95%		
F-Secure Anti-Virus 2008	0	100.00%	1	99.84%	0	100.00%	2	99.24%	0	100.00%	1	99.91%		1
GDATA Anti-virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	99.91%		2
Grisoft AVG	0	100.00%	6	99.84%	200	98.96%	9	96.97%	3	99.93%	695	76.07%		
Ikarus Virus Utilities	9	99.88%	8	99.68%	2461	91.37%	21	94.13%	171	96.07%	353	80.58%	13	31
Iolo Antivirus	34	99.69%	1	99.84%	0	100.00%	2	99.24%	0	100.00%	4	99.83%		
Kaspersky Anti-Virus	1	99.86%	0	100.00%	0	100.00%	2	99.24%	0	100.00%	1	99.91%		
Kingsoft AntiVirus	60	95.63%	600	18.23%	14022	13.56%	96	74.05%	463	90.97%	1634	31.32%		
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Microsoft Forefront	0	100.00%	1	99.84%	0	100.00%	3	99.10%	0	100.00%	80	96.05%		
MWTI eScan	0	100.00%	1	99.84%	0	100.00%	0	100.00%	0	100.00%	1	99.91%		2
Norman Virus Control	7	99.94%	0	100.00%	269	99.29%	11	97.73%	0	100.00%	867	76.84%	3	
PCTools Anti-Virus	0	100.00%	4	99.72%	22	99.58%	10	98.11%	0	100.00%	221	84.99%		
PCTools Spyware Doctor	0	100.00%	2	99.89%	42	99.78%	8	98.86%	3	99.93%	220	85.05%	1	
Quick Heal Quick Heal	0	100.00%	0	100.00%	1088	91.07%	18	96.02%	73	98.23%	1130	73.04%		
Redstone Redprotect	1	99.86%	0	100.00%	0	100.00%	2	99.24%	0	100.00%	1	99.91%		1
Rising Antivirus	2	99.96%	9	98.97%	10993	41.26%	53	89.55%	1273	69.32%	1327	46.17%	1	
Sophos Anti-Virus	4	99.96%	0	100.00%	0	100.00%	0	100.00%	8	99.80%	8	99.61%		
Symantec Endpoint Protection	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%		
Trend Micro OfficeScan	2	99.98%	3	99.89%	749	98.16%	9	98.67%	0	100.00%	738	84.88%		
VirusBuster VirusBuster	0	100.00%	4	99.72%	20	99.79%	10	98.11%	0	100.00%	220	85.05%		

CA Antivirus 9.0.0.143

ItW	99.18%	**Worms & bots**	100.00%
ItW (o/a)	99.18%	**DOS**	99.70%
File infector	99.77%	**Macro**	100.00%
Polymorphic	99.60%	**False positives**	0

A few hiccups occurred during the installation of *CA*'s home-user product, starting with the seemingly inevitable need to upgrade the browser (a minimum of version 5.5 this time). I also noted that some other items come along with the product, including the *Yahoo! Toolbar*, and that the browser homepage was set to *Yahoo!*, which I found rather surprising. I was positively upset by the fact that the boxes

to accept these changes were checked by default – since the VB100 testing protocol requires default settings, this meant agreeing to yet more EULAs, which in traditional *CA* style must be scrolled all the way through before they can be accepted.

The design of the product itself is pretty slick, with clear and easy controls, and despite my misgivings about the optional extras I found myself quite liking it. Configuration was fairly minimal, but the defaults made sense, with archive scanning switched on for on-demand scanning (.ACE files not scanned) and off for on-access scanning except for a single level of the ubiquitous .ZIP (and its twin sister .JAR, essentially zip renamed).

Scanning speeds were very good indeed, and detection generally good, but in the WildList set several items were missed including some W32/Rbot variants and the entire set of the W32/Viruts. *CA* thus misses out on a VB100 award here.

CA eTrust 8.1.637.0

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	99.70%
File infector	99.02%	**Macro**	99.82%
Polymorphic	99.60%	**False positives**	0

CA's more grown-up product, the corporate-targeted *eTrust*, did not complain about the browser in use during the installation, but I found myself needing to upgrade regardless when I later found that some of the popup screens in the options areas of the interface lacked their vital control buttons.

This interface has never been a favourite of mine, but its usual slowness under *Windows XP* was somewhat less intrusive under *2000*. Accessing logs was as tricky as ever, with large ones occasionally overwhelming the display system and leaving me with blank browser windows and no option to export to a text file. As usual I simply removed the raw files to a *Linux* machine and stripped out the required data.

The logs indicated much better coverage of the WildList by *eTrust* than by its sister product, hinting that the home-user product submitted may have been using some slightly older definition data. Archive scanning was a little odd, with a maximum of nine levels checked on demand and none on access, despite the GUI inferring that they should be. Speeds were very good, and without any false positives *eTrust* succeeds where *CA AV* failed, and wins another VB100 award.

Doctor Web Dr.Web 4.44.0

ItW	98.50%	**Worms & bots**	99.84%
ItW (o/a)	98.50%	**DOS**	100.00%
File infector	99.24%	**Macro**	100.00%
Polymorphic	100.00%	**False positives**	2

Dr.Web proved much less problematic, with a simple installer requiring no extra fiddling and another very pleasing interface, laid out with impressive clarity and logic as well as being appealing to the eye. Running through the tests was quite enjoyable as a result, which was a good thing as they did take some time – *Dr.Web* is a very thorough product, delving deeply into files before passing them as clean. On demand, archives were not scanned by default. However, .CHM help files, of which a few are included in the clean set, are checked in all their many sub-parts, which explains the relatively low throughput, rendered even lower when full archive scanning is activated. Full archive scanning covered everything but .ACE to a depth of 10 levels.

Detection rates were excellent across the test sets until the WildList tripped the product up with several misses, including those pesky W32/Virut samples. A couple of false positives added to *Dr.Web*'s problems, and the product unfortunately misses out on a VB100 once more.

ESET NOD32 2.70.39

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

Nod32 has undergone a bit of a revolution recently, with a spanking new interface introduced to coincide with the launch of version 3, and that of its big sister *Smart Security* (see *VB*, November 2007, p.19). However, *ESET* opted to give the ever-reliable version 2.7 one last hurrah this month.

Installing and using the product has never been too difficult, and as usual testing sped through in remarkable time, with the usual excellent results. Speeds were as fast as ever, although archives could not be scanned on access, and detection was at the expected near flawless level, with only a single rather obscure and highly polymorphic sample missed. With the WildList fully covered and no false positives, *ESET* adds yet another VB100 award to its groaning trophy cabinet.

On-demand throughput	Archive files - default		Archive files - all files		Binaries and system files - default		Binaries and system files - all files		Media & documents - default		Media & documents - all files		Other file types - default		Other file types - default	
	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Through-put (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)	Time (s)	Throughput (MB/s)
AEC Trustport Antivirus	4929	0.6	4929	0.6	3157	0.9	3157	0.9	255	5.7	255	5.7	387	1.8	387	1.8
Alwil avast!	20	151.4	812	3.7	194	14.0	223	12.2	22	66.0	57	25.5	18	38.0	43	15.9
Avira AntiVir	616	4.9	650	4.7	100	27.1	99	27.4	25	58.1	30	48.4	19	36.0	36	19.0
BitDefender AntiVirus	1051	2.9	1051	2.9	342	7.9	342	7.9	51	28.5	51	28.5	54	12.7	54	12.7
Bullguard Bullguard	1054	2.9	1054	2.9	318	8.5	318	8.5	64	22.7	64	22.7	67	10.2	67	10.2
CA Antivirus	761	4.0	761	4.0	93	29.2	93	29.2	32	45.4	32	45.4	23	29.8	23	29.8
CA eTrust	466	6.5	466	6.5	70	38.8	70	38.8	28	51.9	28	51.9	20	34.2	20	34.2
Doctor Web Dr. Web	637	4.8	3230	0.9	487	5.6	604	4.5	93	15.6	92	15.8	88	7.8	100	6.8
ESET NOD32	9	336.5	877	3.5	55	49.4	434	6.3	36	40.4	39	37.2	26	26.3	32	21.4
Fortinet Forticlient	471	6.4	471	6.4	494	5.5	494	5.5	27	53.8	27	53.8	45	15.2	45	15.2
Frisk F-PROT	192	15.8	192	15.8	245	11.1	245	11.1	32	45.4	32	45.4	21	32.6	21	32.6
F-Secure Anti-Virus	2218	1.4	2388	1.3	236	11.5	235	11.6	80	18.2	80	18.2	24	28.5	83	8.2
GDATA Anti-virus	2609	1.2	2609	1.2	418	6.5	418	6.5	86	16.9	86	16.9	83	8.2	83	8.2
Grisoft AVG	1130	2.7	2826	1.1	381	7.1	392	6.9	119	12.2	155	9.4	81	8.4	279	2.5
Ikarus Virus Utilities	200	15.1	N/A	N/A	244	11.1	N/A	N/A	50	29.1	N/A	N/A	65	10.5	N/A	N/A
Iolo Antivirus	237	12.8	246	12.3	249	10.9	251	10.8	19	76.5	27	53.8	21	32.6	40	17.1
Kaspersky Anti-Virus	2061	1.5	2061	1.5	403	6.7	403	6.7	90	16.1	90	16.1	81	8.5	81	8.5
Kingsoft AntiVirus	828	3.7	828	3.7	248	10.9	248	10.9	63	23.1	63	23.1	74	9.3	74	9.3
McAfee VirusScan	50	60.6	821	3.7	288	9.4	301	9.0	49	29.6	49	29.6	59	11.6	82	8.3
Microsoft Forefront	939	3.2	939	3.2	276	9.8	276	9.8	62	23.4	62	23.4	40	17.1	40	17.1
MWTI eScan	1832	1.7	1832	1.7	432	6.3	432	6.3	297	4.9	297	4.9	298	2.3	298	2.3
Norman Virus Control	905	3.3	905	3.3	1368	2.0	1368	2.0	46	31.6	46	31.6	150	4.6	150	4.6
PCTools Anti-Virus	397	7.6	704	4.3	1333	2.0	1339	2.0	1283	1.1	1285	1.1	1592	0.4	1595	0.4
PCTools Spyware Doctor	963	3.1	963	3.1	340	8.0	340	8.0	73	19.9	73	19.9	60	11.4	60	11.4
Quick Heal Quick Heal	730	4.1	767	3.9	91	29.8	95	28.6	61	23.8	67	21.7	18	38.0	36	19.0
Redstone Redprotect	1717	1.8	1827	1.7	304	8.9	304	8.9	166	8.8	166	8.8	162	4.2	162	4.2
Rising Antivirus	1564	1.9	1564	1.9	357	7.6	357	7.6	63	23.1	63	23.1	56	12.2	56	12.2
Sophos Anti-Virus	53	57.1	1020	3.0	197	13.8	244	11.1	27	53.8	44	33.0	16	42.8	54	12.7
Symantec Endpoint Protection	684	4.4	684	4.4	218	12.5	218	12.5	64	22.7	64	22.7	63	10.9	63	10.9
Trend Micro OfficeScan	124	24.4	125	24.2	180	15.1	181	15.0	27	53.8	27	53.8	32	21.4	40	17.1
VirusBuster VirusBuster	533	5.7	695	4.4	211	12.9	212	12.8	29	50.1	47	30.9	19	36.0	40	17.1

Fortinet Forticlient 3.0.470

ItW	99.98%	**Worms & bots**	100.00%
ItW (o/a)	99.98%	**DOS**	100.00%

File infector	100.00%	**Macro**	100.00%
Polymorphic	99.90%	**False positives**	0

Fortinet's desktop product remains little changed since I first encountered it, presenting a serious-looking interface

with a wealth of security functions accessed via a string of tabs. During installation the product complained about a missing DLL file, but presumably this related to some other part of the product, as the anti-virus seemed as solid and robust as ever.

Usability was similarly problem-free, and scanning times were decent for the level of thoroughness offered by the default settings, detecting the majority of the nested archives without the need for adjustment.

Detection was splendid almost across the board until those troublesome Virut samples reared their ugly heads, with two missed detections being enough to prevent *Fortinet* from winning another VB100 award.

Frisk F-PROT Anti-virus

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

F-PROT is perhaps the simplest product on test this month, with a fairly basic interface providing access to straightforward anti-virus scanning and cleaning and no additional bells and whistles. This made testing pretty straightforward, and everything zoomed through in good time, with the more in-depth speed tests skipped on access thanks to a dearth of configuration.

Detection was as top-class as ever, with just about everything taken in the engine's stride, but a single false positive showed up in the clean set, a file apparently highly similar to a known malicious item, meaning that *Frisk* joins the growing list of vendors narrowly failing to reach the VB100 standard this month.

F-Secure Anti-Virus 2008

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

F-Secure's current product is another highly familiar one which took little time to get set up and going.

The in-depth scanning with multiple technologies meant speed times were not the best, even though archives could not apparently be scanned deeper than five levels. While running sizeable scans, the interface choked up a few times, lingering unresponsive at the very last stage of the scanning process, with only a reboot able to bring it back in touch with the user. Logging was also a little pesky, with sizeable chunks of information apparently missing from logs exported from the viewer interface.

However, detection was excellent, and there were no false detections, and *F-Secure* thus comfortably earns another VB100 award.

GDATA Anti-virus 18.0.7295.201

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

GDATA's product is another multi-engine beast, which for this submission at least seems to have dropped the familiar '*AVK*' name. The interface seemed unchanged however – a clear and well-laid-out thing which is always a pleasure to operate.

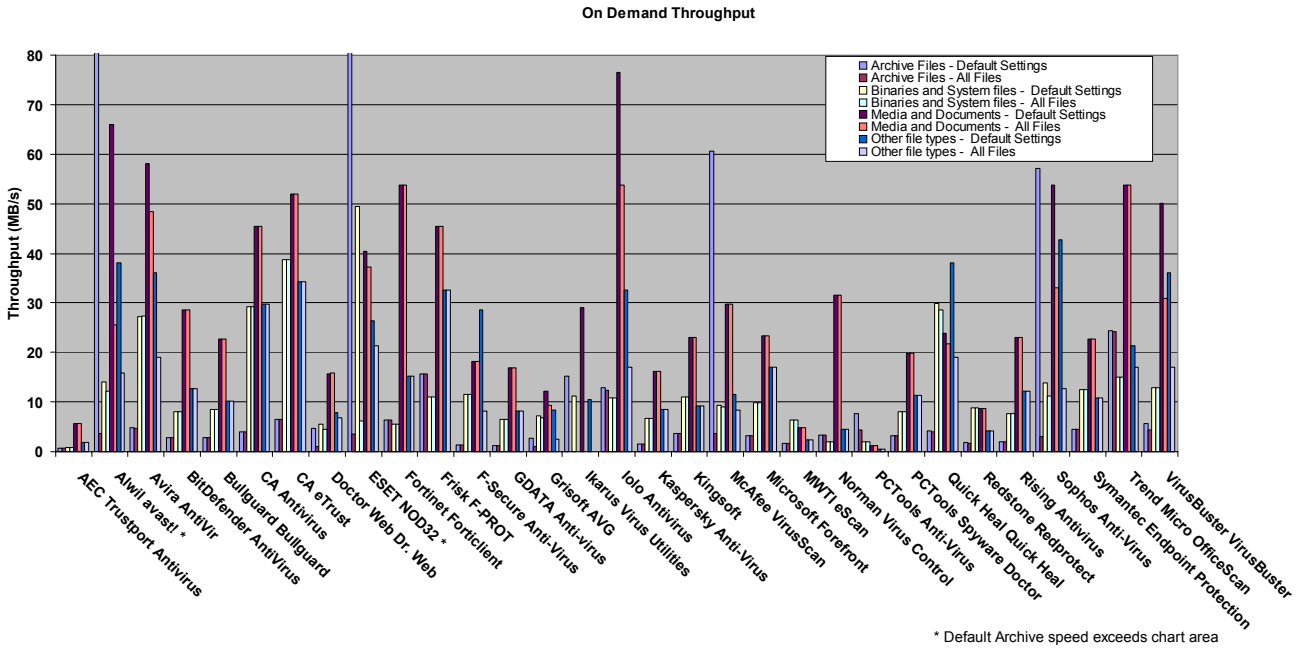
Of course, the multiple engines meant that scanning speeds were slow, even on access, but depth of scanning and accuracy are clearly the product's strengths, and with barely any misses and no false positives *GDATA* also wins a VB100 award for its collection.

Grisoft AVG 7.5.503

ItW	100.00%	**Worms & bots**	99.86%
ItW (o/a)	100.00%	**DOS**	98.96%
File infector	97.73%	**Macro**	100.00%
Polymorphic	76.07%	**False positives**	0

Wildly popular *AVG*, the free home-user version of which seems to be in almost every home these days, has always been a little fiddly for my liking, but whether it has been tweaked a little or I've just grown used to it, in this test I found the interface perfectly reasonable and even quite pleasant to work with.

Configuration was a little short for the on-access scanner, but elsewhere everything worked fine, with very good if not great detection in the infected sets, including flawless coverage of the WildList despite those difficult polymorphic samples. With no false positives either, *Grisoft* also wins another VB100 award.



* Default Archive speed exceeds chart area

Ikarus Virus Utilities 1.0.60

ItW	99.88%	**Worms & bots**	99.81%
ItW (o/a)	99.88%	**DOS**	91.37%
File infector	93.37%	**Macro**	96.07%
Polymorphic	80.58%	**False positives**	13

Ikarus has had some problems in its recent entries in VB100 comparative reviews, but earlier issues with its interface seem to have been resolved – on this occasion everything ran fine and stably with no difficulty. Even the updates to the *Windows Installer* and the .NET framework required by the product were provided thoughtfully as part of the submission and installed automatically as part of the setup process.

Configuration of scanning is somewhat limited by the interface, but the default setting of scanning up to three levels into archive files seems sensible, and speeds were fairly good across the sets.

Detection was a little improved on previous efforts, but a handful of samples of each of two Virut variants in the set proved undetectable, and a rash of false positives added to *Ikarus*'s woes. There were also a fair number of items labelled 'not-a-virus: Monitor.Win32.Keylogger', which for now I have generously recorded as 'suspicious' rather than full false positive detections, but which certainly seem a little suspect themselves.

Despite these problems the product seems to be improving fast and looks a likely candidate to qualify for a VB100 award sometime soon.

Iolo Antivirus 1.1.15

ItW	99.71%	**Worms & bots**	99.84%
ItW (o/a)	99.69%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.83%	**False positives**	0

Iolo returns to the test bench for another stab after being denied a VB100 by a whisker a few months ago. The product is well designed and pleasant to use, and although it requires *IE6* to operate, it politely offers to go online and fetch a copy.

As with many of the products aimed more squarely at the home user, configuration was somewhat limited, with on-access scanning barely adjustable and actions on discovering malware restricted to delete, disinfect or quarantine. With logging also absent, I allowed the product to delete the virus collections from the system, which left only a few samples in most sets but also many of the two Virut strains along with another file infector, W32/Expiro. *Iolo* will therefore have to try again for the VB100 award, which should be well within its grasp with just a little more work.

Kaspersky Anti-Virus 7.0.0.125

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	99.86%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.91%	**False positives**	0

Kaspersky is a much more seasoned product, version 7 of the product having dropped a lot of the cuddly cartoonishness of the previous offering and presenting a sterner but glossier face to the world. Usability has not been diminished however, and few problems were encountered other than some slowness exporting particularly large logs to file.

Detection rates were excellent as ever, with the new nested set detected very neatly. With no false positives spotted, all looked good until a single item was missed on access. This, an instance of W32/Autorun added recently to the list, could be detected by the product on demand, but was not scanned on access unless the 'scan installation packages' option was activated. *Kaspersky* thus narrowly misses out on a VB100 award this time.

Kingsoft AntiVirus

ItW	95.63%	**Worms & bots**	18.23%
ItW (o/a)	95.63%	**DOS**	13.56%
File infector	74.05%	**Macro**	90.97%
Polymorphic	31.32%	**False positives**	0

Kingsoft achieved a VB100 award in its previous appearance in *VB* (see *VB*, August 2007, p.13). The product this time seemed little changed, with the interface nicely laid out and appearing pretty stable, but experiencing some difficulties in the log viewer when faced with unfamiliar locales – only US English is supported, and others cause a nasty crash.

Scanning speeds were rather average, and configuration absent on access, but false positives and even suspicious flags were encouragingly absent throughout the clean sets.

The infected sets were less well covered, in particular the older items, and in the WildList set several nasties were missed, including most of the files infected with Virut and Expiro, as well as several W32/SDbot variants. *Kingsoft* thus falls short of the required standard this time, and will have to try again to achieve its second VB100 award.

McAfee VirusScan Enterprise 8.50i

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

McAfee's desktop product is another that seems to have remained relatively unchanged for some time, and its performance was similarly predictable.

Scanning times were decent, with archives ignored by default in both modes but thoroughly handled if requested; detection was impeccable, with nothing missed anywhere and no false positives. *VirusScan* wins a VB100 award effortlessly.

Microsoft Forefront Client Security 1.5.1941

ItW	100.00%	**Worms & bots**	99.84%
ItW (o/a)	100.00%	**DOS**	100.00%
File infector	99.86%	**Macro**	100.00%
Polymorphic	96.05%	**False positives**	0

Perhaps unsurprisingly, *Forefront* makes use of all available *Microsoft* technology and requires numerous updates to be in place before it will install. The rollup package, an improved version of the installer, and an update to the Agent API are all required. It also uses the event log to record its activities rather than providing its own system, which I found a little awkward, but the server-side management system doubtless provides a more usable form of information management.

Configuration was rather minimal, which again may be explained by the absence of the management side of things, but defaults were sensible and testing ran without difficulties. With nothing of significance missed and no false positives, *Forefront* qualifies for a VB100 award.

MWTI eScan 9.0.747.1

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

Microworld Technologies presents a fairly comprehensive product, including the *Kaspersky* engine alongside a range of its own protection technologies. The product's default settings lean towards the paranoid, with on-access defaults including all archive types. With a well designed interface providing for all my needs, testing thus took little of my own time, but quite a bit for the system, as clean sets were probed deeply.

Detection of the infected sets was excellent, *eScan* managing to avoid the problem which upset *Kaspersky*'s own product, and comfortably earning a VB100 award.

File access lag time	Archive files - default		Archive files - all files		Binaries and system files - default		Binaries and system files - all files		Media & documents - default		Media & documents - all files		Other file types - default		Other file types - all files	
	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)	Time (s)	Lag (s/MB)
AEC Trustport Antivirus	1007	0.3	1007	0.3	328	0.1	328	0.1	98	0.1	98	0.1	129	0.2	129	0.2
Alwil avast!	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Avira AntiVir	28	0.0	112	0.0	95	0.0	127	0.0	24	0.0	35	0.0	15	0.0	43	0.1
BitDefender AntiVirus	114	0.0	N/A	N/A	259	0.1	259	0.1	53	0.0	53	0.0	58	0.1	58	0.1
Bullguard Bullguard	113	0.0	900	0.3	283	0.1	315	0.1	51	0.0	58	0.0	63	0.1	67	0.1
CA Antivirus	22	0.0	N/A	N/A	83	0.0	83	0.0	33	0.0	33	0.0	27	0.0	27	0.0
CA eTrust	19	0.0	N/A	N/A	73	0.0	73	0.0	33	0.0	33	0.0	26	0.0	26	0.0
Doctor Web Dr. Web	540	0.2	2050	0.7	480	0.2	908	0.3	84	0.1	84	0.1	81	0.1	87	0.1
ESET NOD32	12	0.0	N/A	N/A	63	0.0	63	0.0	42	0.0	42	0.0	33	0.0	33	0.0
Fortinet Forticlient	308	0.1	308	0.1	268	0.1	268	0.1	28	0.0	28	0.0	43	0.1	43	0.1
Frisk F-PROT	64	0.0	N/A	N/A	263	0.1	263	0.1	41	0.0	41	0.0	27	0.0	27	0.0
F-Secure Anti-Virus	36	0.0	1432	0.5	202	0.1	222	0.1	36	0.0	133	0.1	26	0.0	105	0.1
GDATA Anti-virus	222	0.1	1380	0.5	371	0.1	396	0.1	163	0.1	172	0.1	116	0.2	132	0.2
Grisoft AVG	18	0.0	N/A	N/A	130	0.0	130	0.0	22	0.0	28	0.0	10	0.0	29	0.0
Ikarus Virus Utilities	209	0.1	N/A	N/A	254	0.1	254	0.1	53	0.0	53	0.0	70	0.1	70	0.1
Iolo Antivirus	52	0.0	N/A	N/A	241	0.1	261	0.1	26	0.0	37	0.0	25	0.0	27	0.0
Kaspersky Anti-Virus	37	0.0	214	0.1	199	0.1	222	0.1	75	0.0	84	0.1	48	0.1	72	0.1
Kingsoft AntiVirus	59	0.0	N/A	N/A	229	0.1	229	0.1	71	0.0	71	0.0	80	0.1	80	0.1
McAfee VirusScan	48	0.0	479	0.2	284	0.1	295	0.1	47	0.0	47	0.0	58	0.1	58	0.1
Microsoft Forefront	90	0.0	N/A	N/A	273	0.1	273	0.1	77	0.0	77	0.0	40	0.0	40	0.0
MWTI eScan	999	0.3	999	0.3	218	0.1	218	0.1	80	0.1	80	0.1	73	0.1	73	0.1
Norman Virus Control	16	0.0	N/A	N/A	110	0.0	110	0.0	53	0.0	53	0.0	74	0.1	74	0.1
PCTools Anti-Virus	345	0.1	N/A	N/A	890	0.3	N/A	N/A	123	0.1	N/A	N/A	97	0.1	N/A	N/A
PCTools Spyware Doctor	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Quick Heal Quick Heal	14	0.0	N/A	N/A	81	0.0	N/A	N/A	37	0.0	N/A	N/A	15	0.0	N/A	N/A
Redstone Redprotect	43	0.0	1448	0.5	227	0.1	259	0.1	112	0.1	115	0.1	91	0.1	96	0.1
Rising Antivirus	55	0.0	N/A	N/A	327	0.1	327	0.1	64	0.0	64	0.0	62	0.1	62	0.1
Sophos Anti-Virus	31	0.0	1011	0.3	204	0.1	228	0.1	35	0.0	36	0.0	21	0.0	49	0.1
Symantec Endpoint Protection	24	0.0	N/A	N/A	216	0.1	N/A	N/A	35	0.0	N/A	N/A	33	0.0	N/A	N/A
Trend Micro OfficeScan	1052	0.3	1052	0.3	930	0.3	930	0.3	40	0.0	40	0.0	43	0.1	43	0.1
VirusBuster VirusBuster	31	0.0	N/A	N/A	214	0.1	215	0.1	27	0.0	45	0.0	15	0.0	40	0.0

Norman Virus Control v.5.9

ItW	99.94%	**Worms & bots**	100.00%
ItW (o/a)	99.94%	**DOS**	99.29%
File infector	98.48%	**Macro**	100.00%
Polymorphic	82.17%	**False positives**	3

Norman's is another interface which has grown on me after struggling to understand its complexities in earlier tests. The

only lingering annoyance is the lack of information on scan progress, with there being no progress bar or count of files scanned so far.

Speeds were reasonable, and detection levels decent, with most misses on old and obscure items. However, two files in the clean sets were flagged as nondescript malware by the heuristics, thanks to the use of a rather unusual packer, and again some of those tricky Virut samples were missed, leaving *Norman* just short of the mark for the VB100 award this month.

PCTools Anti-Virus 3.6.1.7

ItW	100.00%	**Worms & bots**	99.89%
ItW (o/a)	100.00%	**DOS**	99.58%
File infector	98.86%	**Macro**	100.00%
Polymorphic	84.99%	**False positives**	0

PCTools is a relative newcomer to VB100 comparative testing, taking its first award just a few months ago (see *VB*, June 2007, p.10).

The plain anti-virus product, based on the *VirusBuster* engine, offers a reasonable level of configuration and a pleasant user experience for the most part. The logging presented rather a strange problem though – opening logs from the interface brought up a 'file in use' error, and they could thus only be accessed by copying the files and opening the copies.

Some good detection rates were shown, but also some remarkably slow times in the speed tests, even with the default on-demand settings scanning archives to a depth of one level only. However, with nothing missed in the WildList and no false positives, *PCTools AV* wins itself a second VB100 award.

PCTools Spyware Doctor 5.1.0.272

ItW	100.00%	**Worms & bots**	99.89%
ItW (o/a)	100.00%	**DOS**	99.78%
File infector	98.86%	**Macro**	99.93%
Polymorphic	85.05%	**False positives**	1

Spyware Doctor is *PCTools*' rather more venerable anti-spyware product, now available with anti-virus functionality rolled in, and while the interface closely resembles the plain AV product there were a number of differences.

Logging seemed to be limited to a small file size, meaning that larger scans needed to be split up into chunks to acquire the necessary data, while on-access scanning seemed not to be sparked by simple file opening, which meant the product had to be excluded from the on-access speed test.
On-demand times were considerably better than those of its sister product, despite defaults including all archive types (apart from the rather obscure .LZH) to a depth of at least 10 levels.

Detection rates differed slightly too, and in the clean set the anti-spyware side of things detected a single false positive, thus denying *Spyware Doctor* a VB100 despite full coverage of the WildList.

Quick Heal Quick Heal AntiVirus Lite 9.50

ItW	100.00%	**Worms & bots**	100.00%
ItW (o/a)	100.00%	**DOS**	95.18%
File infector	96.59%	**Macro**	98.23%
Polymorphic	73.04%	**False positives**	0

Quick Heal (which is now the name of both the product and its vendor, having recently changed from *CAT*) is another well designed product.

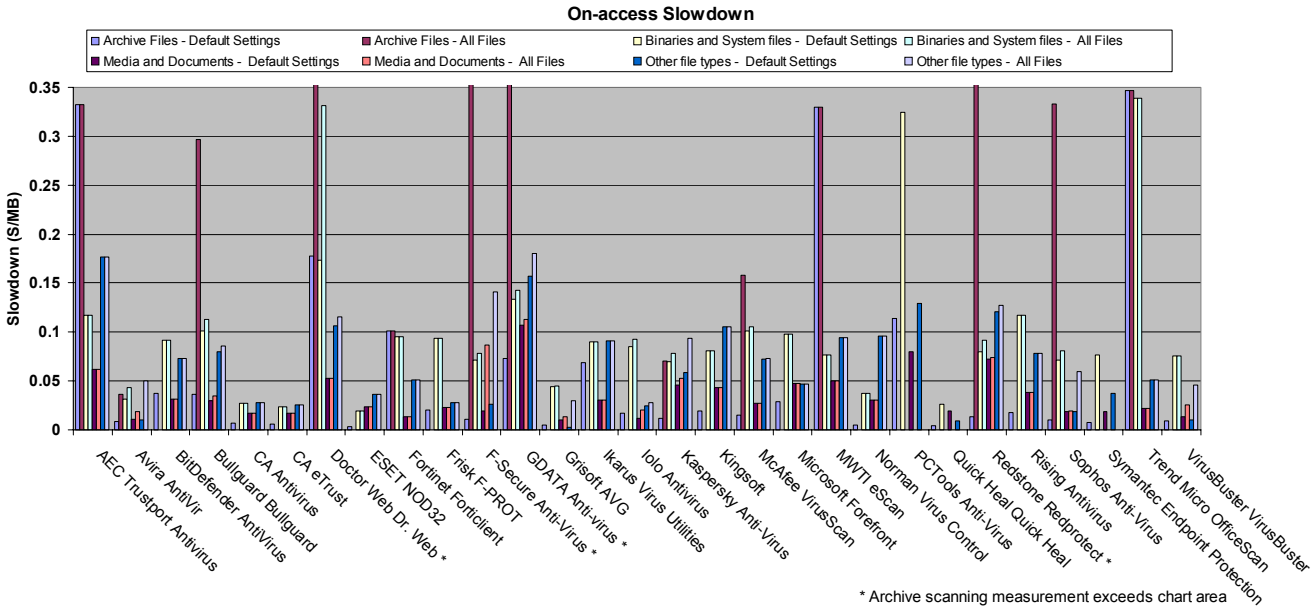
It zipped through speed tests in good time and could only be cajoled into scanning to a depth of five levels, into a limited selection of archive types. A few nasty crashes occurred during the scanning of infected sets, but they were handled better on a second attempt, and while detection was a little short on the older sets nothing more important was missed, and false positives were also absent. *Quick Heal* thus earns itself a VB100 award.

Redstone Redprotect 0.4.1.27681

ItW	99.86%	**Worms & bots**	100.00%
ItW (o/a)	99.86%	**DOS**	100.00%
File infector	100.00%	**Macro**	100.00%
Polymorphic	99.91%	**False positives**	0

UK-based *Redstone* produces a managed-service protection product, of which this is a simple client version using the .NET framework for its interface. Running the product is a straightforward business, with a simple menu accessed via the system tray. Configuration is a little more fiddly, requiring the tweaking of registry settings, but the submission came with a prepared set of useful entries, enabling testing to proceed without too many problems.

The product is based on the *Kaspersky* engine, and detection rates were thus at the top end of the scale, while speed times were more average. A few difficulties were encountered,



including the absence of logging and some odd behaviour on demand, when the 'always delete' option seemed to be ignored for a few items, resulting in a string of popups requesting confirmation before deleting.

False positives were absent, but the W32/Autorun sample which tripped up *Kaspersky* was also missed here, in both modes, and *Redstone* will thus have to try again before gaining a VB100 award.

Rising Antivirus 2008 20.15.32

ItW	99.97%	**Worms & bots**	99.44%
ItW (o/a)	99.96%	**DOS**	41.26%
File infector	90.30%	**Macro**	69.32%
Polymorphic	46.17%	**False positives**	2

Another newcomer to the VB100 test bench, China-based *Rising* has developed a considerable profile outside its home country in recent years, and it was with some excitement that I took my first look at its product. First impressions were excellent, with the product looking very clean and stylish, clearly laid out and easy to use.

Speed test results were fairly good, and stability seemed solid too, but during on-access scanning of the infected sets the product seemed to stop blocking after 10,000 samples or so. The test was retried at a slower pace. The problem did not recur, and results were thus obtained, showing the expected high numbers of misses in older sets but little in the newer areas. Two misses in the WildList, both single samples from sets of file-infectors, and a pair of false positives in the clean sets, were enough to spoil *Rising*'s chances of qualifying for the VB100 at first attempt, but it is another likely candidate to make the grade pretty soon.

Sophos Anti-Virus 7.03

ItW	99.96%	**Worms & bots**	100.00%
ItW (o/a)	99.96%	**DOS**	100.00%
File infector	100.00%	**Macro**	99.80%
Polymorphic	99.61%	**False positives**	0

Sophos is among the most regular of VB100 entrants, with its product little changed in the half-dozen *Windows* tests I have performed in my time here, and as usual setting it up and running the tests were simple tasks. Speeds were very good, even with archive scanning turned up to the maximum available five levels, and after a false positive upset things last time (see http://www.virusbtn.com/vba/2007/10) the clean sets were cleared with only a few hacker tools alerted on as possible security risks.

Detection was at its usual high levels, with almost everything covered, but again in the WildList set those Virut samples proved too difficult, and *Sophos* is denied the VB100 for the second time in a row.

Symantec Endpoint Protection 11.0.780.1109

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

This month *Symantec* presented a totally different looking product from those seen in previous tests, considerably more colourful and less severe. The cosmetic enhancements required *IE6*, and after the installer had aborted requesting this update it left something lingering behind, which meant the *IE6* installer insisted on a reboot before it could run itself. However, after several reboots to set up, tests continued apace.

Speeds were reasonable, although configuration was somewhat less in-depth than in previous submissions and archives could only be scanned to a depth of three levels, with .ACE and .TGZ ignored. However, detection was excellent, with nothing missed, and without false positives either *Symantec* earns another VB100 award.

Trend Micro OfficeScan Client 8.0

ItW	99.98%	**Worms & bots**	99.89%
ItW (o/a)	99.98%	**DOS**	98.16%
File infector	98.67%	**Macro**	100.00%
Polymorphic	84.88%	**False positives**	0

OfficeScan also required *IE6* in order to operate the web console which provides much of the product's configuration, although options were available to delegate some control to the simpler local interface.

Testing slipped rapidly along, flipping between the two control systems as required, and times were good and detection rates decent, although the renamed Eicar test file was not spotted with the default settings. Some older sample sets were a little short, but more seriously two Virut samples were missed, one each of the two variants causing most trouble here, and *Trend* is thus denied an award this time.

VirusBuster VirusBuster Professional 5.3 Build 39

ItW	100.00%	**Worms & bots**	99.89%
ItW (o/a)	100.00%	**DOS**	99.79%
File infector	98.86%	**Macro**	100.00%
Polymorphic	85.05%	**False positives**	0

Bringing up the alphabetical rear, *VirusBuster* presented its usual colourful and reasonably usable product, which provided adequate configuration options and its usual slightly fiddly system of setting up scanning jobs. These jobs showed good scanning speeds, and pretty thorough detection across the sets; with those troublesome Virut variants taken in its stride, and without any sign of a false positive *VirusBuster* takes home another VB100 award.

CONCLUSIONS

Having expected numerous problems to have arisen from the aging platform, these proved to be limited to the chore of installing extras before products could install or operate properly.

In fact, far more difficulties were thrown up by another rather old issue, the polymorphic file-infecting virus. With modern malware trends having tended for some time towards the non-self-replicating, or at least towards static worms which simply drop identical copies of themselves around the place, old-style file infectors have been making something of a comeback lately. W32/Detnat, W32/Looked (aka Viking), W32/Fujacks, and of course the more tricky polymorphic type, W32/Polip and W32/Virut, all lurk on the WildList and some of them have made a considerable impression on global prevalence charts in recent months. This month's Virut addition revealed deficiencies in detection for several products, the vendors of which have all been informed of the problem, which should have been resolved by most in advance of the publication of this review.

A swathe of products have also fallen to another problem which has shown a rising trend lately: false positives. A relatively small addition to the clean test sets threw up several individual examples (few of the files that were false alarmed on affected more than one product, or more specifically one engine), and in some cases several files were misidentified by a single product.

The result has been one of the poorest scores for some time in a *VB* comparative, with fewer than half the entrants making the grade, and another trend – the inclusion of third-party engines in products – magnifying the scale of the problem. Hopefully the shock of so much devastation caused by a few polymorphic viruses will ensure virus labs remain on their guard and encourage more thorough checking of detection for file-infecting items in future.

Technical details

Test environment: Tests were run on identical machines with *AMD Athlon64 3800+* dual core processors, 1GB RAM, 40GB and 200 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive, all running *Microsoft Windows 2000 Professional SP4*.

Agnitum Outpost was tested on a 1.6 GHz *Intel Pentium* machine with 512 MB RAM and is thus excluded from speed measurements.

END NOTES & NEWS

The 23rd ACSAC (Applied Computer Security Associates' Annual Computer Security Conference) will be held 10–14 December 2007 in Miami Beach, FL, USA. 42 refereed papers, six case studies, three panel sessions and a 'work in progress session' will cover a range of research topics, from security for P2P and mobile computing to malware and forensics. For details see http://www.acsac.org/.

Black Hat DC 2008 Briefings and Training will be held 11–14 February 2008 in Washington, DC, USA. The conference will focus on wireless security and offensive attacks in addition to the core set of training sessions. A call for papers for the Briefings closes 4 January 2008. For full details and registration see http://www.blackhat.com/.

The SecureLondon Conference on emerging threats will be held 4 March 2008 in London, UK. Attendees will be given an overview of the interaction between web, spam and malware, with a focus on specific campaigns. Sessions will engage in the devastating effects and developments of DDoS attacks and how to avoid them, email encryption and the social engineering threat communities pose to a company. For further information see https://www.isc2.org/cgi-bin/events/information.cgi?event=48.

Black Hat Europe 2008 takes place 25–28 March 2008 in Amsterdam, the Netherlands. Registration is now open, and a call for papers closes 1 February. For details see http://www.blackhat.com/.

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

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

EICAR 2008 will be held 6–8 May 2008 in Laval, France. A call for papers has been issued, the deadlines for which are 20 January 2008 for peer-reviewed papers and 20 December 2007 for non-reviewed papers. See http://www.eicar.org/conference/ for the full details.

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

The 9th National Information Security Conference (NISC) will be held 21–23 May 2008 in St Andrews, Scotland. An early bird discount applies until 31 January. For full details and registration information see http://www.nisc.org.uk/.

The 20th annual FIRST conference will be held 22–27 June 2008 in Vancouver, Canada. The conference provides a forum for sharing goals, ideas, and information on how to improve global computer security. The five-day event comprises two days of tutorials and three days of technical sessions where a range of topics of interest to teams in the global response community will be discussed. For more details see http://www.first.org/conference/.

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

Black Hat USA 2008 takes place 2–7 August 2008 in Las Vegas, NV, USA. Online registration and a call for papers open 1 January 2008. For details see http://www.blackhat.com/.

VB2008 will take place 1–3 October 2008 in Ottawa, Canada. *VB* is seeking submissions from those wishing to present papers at VB2008. Full details of the call for papers are available at http://www.virusbtn.com/conference/vb2008. Other enquiries should be directed to vb2008@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)1235 531889
Email: editorial@virusbtn.com Web: http://www.virusbtn.com/

vb Spam supplement

CONTENTS

NEWS & EVENTS

OPENING OF A (SPAM) CAN OF WORMS

Hormel Foods Corp., inventor and manufacturer of the world-famous canned meat product *SPAM*, has lost a lawsuit against Seattle-based company *Spam Arrest* in which it called for the company to drop the word 'Spam' from its name, arguing that it damages the trademark associated with the luncheon meat.

Hormel is famously protective of the word 'spam', having trademarked the word in upper case letters and having launched several previous attempts to prevent other companies from using the word as part of their names or trademarks. This time it was the US Trademark Trial and Appeal Board that ruled against *Hormel*, saying that consumers of canned *SPAM* were unlikely to confuse it with the *Spam Arrest* anti-spam software.

Spam Arrest's attorney is reported to have said that the decision opens the door for other anti-spam software companies to incorporate the word 'spam' into their trademarked product names. *Hormel* was said to be disappointed with the outcome and reviewing its options, including an appeal.

EVENTS

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

The 2008 Spam Conference will take place 27–28 March 2008 in Cambridge, MA, USA. Potential speakers are invited to submit proposals for papers, tutorials or workshops at any point until 1 March 2008. For the full details see http://spamconference.org/.

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

FEATURE

SPAM AND THE DIGITAL DIVIDE

Reza Rajabiun
COMDOM Software and York University, Canada

Over recent years, large volumes of spam seem to have become a permanent feature of the Internet. While the exact volume of spam changes on a daily basis, typically around 80% of all messages are classified as unwanted. Even if the theoretical ideal of a perfect Bayesian content filter were to exist, a high noise-to-signal ratio would still incur significant costs in terms of the physical and human resources required to provide end-users with access to new information technologies [1].

In addition to frustrating end-users and weakening their trust in information technology, spam increases the total volume of traffic that must be processed by ISPs and other network providers – thus also pushing up their costs. The increased costs can have significant implications for the provision of Internet and messaging services, especially in developing countries.

This article focuses on the implications of spam for developing countries and the persistence of the digital divide. We assert that the adoption of high-capacity, self-learning content filters at the server level must be an integral part of efforts to address the gap in access to information technologies across the global population.

THE PROBLEM

Given the way in which spam has evolved over the past decade, it would seem safe to assume that the very low cost of sending messages via email and other new information technologies has acted as a strong driving factor for individuals to send spam in open networks. Closing a network to certain classes of external traffic, or imposing some form of 'tax' on senders would, of course, mitigate this. However, such drastic measures would be inconsistent with the role of the Internet as a platform for end-users to communicate cheaply and effectively, at both local and global levels.

Although we are not aware of any empirical studies that have measured the impact of spam on the digital divide, the extent of the gap between rich and poor countries in terms

of access to new information technologies is clear. Recent data published by the International Telecommunications Union (ITU) highlights the challenge in bridging the digital divide. Figure 1 illustrates the evolution of this divide in terms of the percentage of Internet users in the population from 1995 to 2006. The graph shows that while a large gap remains, some narrowing has taken place due to an increase in Internet availability in developing countries over the past few years [2]. As global access to the telecommunications infrastructure increases, we can expect to see a further increase in the volume of spam.

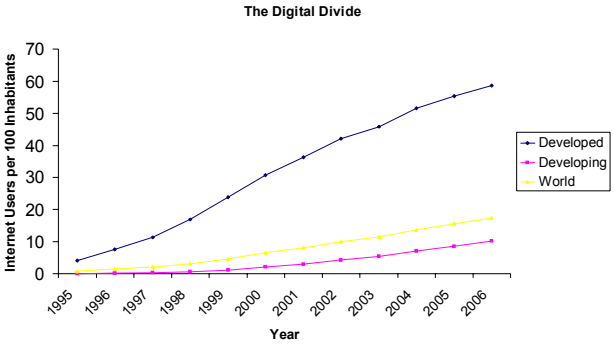


Figure 1: The digital divide.

A high proportion of noise relative to signal consumes large amounts of bandwidth and processing powers that are already scarce. This results in a high total cost of ownership (TCO) of providing Internet access to the billions of people who cannot otherwise benefit from educational and commercial services, as well as lower rates of return on investment – thus discouraging the public and private sectors from building network capacity.

Despite some narrowing of the digital divide, the ITU data shows that significant asymmetries persist geographically in the available Internet bandwidth. For instance, Denmark, a relatively small developed economy, has twice the capacity of Latin America and the Caribbean put together. From the end-users' perspective, bandwidth constraints slow the downloading of data, which, outside of major urban areas, tends to take place through expensive dial-up connections and (increasingly) mobile platforms.

Additionally, if the emerging networks in developing countries get clogged up with spam too quickly, end-users' trust in new information technologies will be undermined. This could present a significant obstacle to the development of new economic applications of telecommunications technology such as mobile text messaging payment systems in countries with underdeveloped banking infrastructures.

For the ISPs that provide the underlying services, bandwidth constraints exacerbate the traditional problems that face network operators in developed economies. These providers must employ a larger number of servers to process noisy incoming traffic. Bandwidth constraints mean that more of the processing must take place at the server level, in order to allow more people to access their messaging applications. The adoption of efficient anti-spam systems at the server level generally lowers the impact of spam on bandwidth, hardware and software.

Managing volatile and complex forms of spam impacts further on the administrative resources of the service providers. A 2005 report by the Organization for Economic Cooperation and Development (OECD) highlighted how the impact on human resources is magnified for infrastructure providers in developing countries [3]. The report identified the limited experience of workers in developing countries with spam, which challenges the stability of the operating infrastructure during sudden surges of global or local spam. Moreover, the assignment of scarce administrative skills to fighting spam has costs in terms of lost opportunities in implementing more productive applications, from education and training to those aimed at improving the efficiency of markets in developing countries [4]. Increased automation of the spam-filtering process will help mitigate the human resource costs of spam.

The 2005 OECD report estimated the cost of spam for a (very) large ISP, operating under administrative and bandwidth constraints common to developing countries, to be around 10% of its operating budget. Given the presence of scale economies in information technology management, the overall costs are likely to be higher for smaller ISPs with less capacity to hire well trained administrators, implement state of the art hardware or receive volume discounts on bandwidth from backbone operators.

The significance of the network costs of spam has resulted in proposals for a wide range of regulatory, economic and technological mechanisms for tackling the spam problem, some of which appear more practical than others.

COST MITIGATION

Regulatory solutions to the spam problem have been adopted in an increasingly large number of jurisdictions since the early 2000s, including those in several developing countries. These laws typically involve imposing restrictions on spammers through criminal and/or civil sanctions. However, as detailed by Ramachandran and Feamster [5], there are many ways in which spammers can hide their identity within the infrastructure of large backbone providers through BGP spectrum agility techniques. These techniques render sender-oriented regulatory strategies ineffectual.

The economic approach to mitigating the spam problem suggests that the noise we observe today is an inevitable by-product of the adoption of technologies that radically lower the costs of sending information. A large number of proposals have been put forward for adopting mechanisms that are aimed at reallocating the costs of sending massive volumes of messages back to the spammers. However, much like regulatory solutions, the implementation of economic mechanisms requires the presence of credible sender authentication procedures. Widely used spamming technologies that are available on a commercial basis can bypass authentication protocols easily, thus both regulatory and economic solutions have found limited practical success.

One unfortunate result of the resilience of spammers has been the increased use of blacklisting, which arguably threatens to divide the global email system. Blacklisting and other ad hoc methods of identifying spam can be inefficient and discriminatory. For instance, large parts of the Chinese system are now blocked by the rest of the world, raising significant concerns for China's Internet users. Local networks may construct national or regional 'walled gardens' by limiting incoming and outgoing traffic through ad hoc administrative decisions such as blocking messages containing non-standard text such as Chinese, Cyrillic or Arabic.

With the emergence of image spam, which places even higher demands on processing and bandwidth, some administrators have reacted similarly by excluding from their networks all messages that contain pictures. Such efforts may be justified to maintain the stability of a network in the shorter term, but clearly they limit the usefulness of the Internet as a global platform for personal and business communications.

Given the inadequacy of regulatory and economic solutions, the optimization and automation of anti-spam systems appears to be the most practical solution for reducing the network costs of spam, and hence their impact on the digital divide.

Over the past years, the rising network costs of spam have motivated anti-spam software developers not only to enhance the accuracy of their systems by taking account of end-user preferences, but also to increase automation and throughput.

At least since the proposal by Sahami *et al.* [6], computer scientists have argued that the development of Bayesian content filters will offer the most efficient solution in terms of accuracy. One reason for this is that content classifiers that can learn from end-user preferences about what constitutes ham and spam can take account of the subjective nature of such a classification process in a heterogeneous network. The open source *SpamAssassin* project, which now serves as the core of numerous commercial front-end software and appliances, has followed these early insights [7].

However, some large ISPs switched to a second type of spam filtering in the early 2000s which relied on the characterization of spam as a large number of similar messages, rather than by scanning and filtering the content of the messages themselves. Although less accurate than Bayesian filters, the so-called fingerprinting/checksum systems offered much higher throughput rates [8].

Our tests indicate that a *Linux* server using *SpamAssassin* running on a 1.7 GHz CPU can process around 20 messages per second. The throughput rate of the leading fingerprinting/checksum systems available today (as reported by their providers) converges to a rate of around 100 messages per second on comparable hardware and OS configurations. This difference explains to some degree why some large ISPs switched to commercial fingerprinting systems, despite their limited accuracy relative to Bayesian filters. Theoretically, fingerprinting systems lowered the total number of servers required to handle a specific volume of traffic by a factor of 5.

Unfortunately, spammers quickly learned to automate the production of large volumes of messages that each appear unique to a fingerprinting system. To some degree, the battle between spammers and these systems has contributed to the growth and sophistication of the spam we observe today [9].

More recently, developers of Bayesian systems have worked on increasing their throughput rates radically by implementing advanced pattern scanning and content classification techniques. For instance, the Tachyon Core scanning engine in the *COMDOM Antispam for Servers* software produces throughput rates of around 600 messages per second on similar configurations to those noted above. Anti-spam system developers have responded to the economic problem by increasing the processing capacity of their software by more than 30 times during a period of less than five years. This improvement means that one mail server operating on a second-generation Bayesian filter can handle the same volume of traffic as six servers relying on the fastest of fingerprinting/checksum systems.

In addition to their higher processing efficiency and accuracy, Bayesian content filters allow for the decentralization and automation of spam identification. Management of fingerprinting systems necessitates a centralized architecture through which anti-spam software developers adjust their checksum-generating algorithms to respond to changes in randomization techniques. This design feature requires communication between local

servers and a centralized database of checksums, which further drains bandwidth and processing power. Advanced Bayesian content filters learn automatically from end-user behaviour, place this knowledge in a local database, and then identify spam/ham based on the historical preferences that have been used to train them. Localization reduces bandwidth constraints, as advances in automation reduce the need for continuous administrative intervention, and ad hoc exercises in rule setting.

IMPLEMENTATION

Unfortunately, the development of more efficient technological solutions does not necessarily translate into increased availability of electronic communications. One reason for this is the fixed switching costs arising from decisions made earlier about operating systems and security applications. However, switching costs are likely to be more relevant to the choice of anti-spam technologies in developed countries, where more people are already 'tied in' to older and/or less efficient software.

The urban/rural divide within developing countries poses specific challenges in terms of extending points of contact between end-users and the local hubs required to process and deliver their messages. Adoption of more efficient anti-spam technologies will lower the network costs facing all ISPs. However, this does not mean that existing providers will necessarily use these resources to extend access to more remote areas.

On a more positive note, it is imperative to remember that advances in mobile technologies are making it increasingly less costly to extend the traditional reach of the digital economy beyond urban areas. Solving the 'last mile' problems lowers the costs of extending access into areas with a low population density. In conjunction with low-cost and multi-tasking mobile devices, such as the 'one laptop per child' program, such advances have the potential to narrow the divide we observe today at the global level [10]. Unfortunately, if the experience of developed countries is any guide, the reduction in costs will be accompanied by a rise in undesirable content for the new end-users.

Some of the noise will be from the large global flows of spam that we see today, sent mostly from the networks of large operators in Western Europe and North America. Another portion will be produced by local sellers, who will use the new technologies to find buyers for their products and services.

Regardless of their origins, large volumes of spam necessitate a capacity to scan and filter electronic content efficiently, that is: accurately, quickly, and with the minimum level of administrative intervention. The adoption of fast, self-learning filters should be encouraged with the implementation of targeted programs that condition technology licensing on increased level of access [11]. The lower the resources required to run messaging servers, the more resources (both human and physical) will be available to narrow the digital divide.

REFERENCES & NOTES

[1] Loder, T. C.; Van Alstyne, M. W.; Wash, R. 'Information asymmetry and thwarting spam' for an intuitive economic model of spam production, the perfect Bayesian filter, and other classes of solutions. 2004. http://ssrn.com/abstract=488444.

[2] http://www.itu.int/ITU-D/ict/statistics/ict/index.html.

[3] Spam issues in developing countries. http://www.oecd-antispam.org/.

[4] Chowdhury, S. K. 'Search cost and rural producers' trading choice between middlemen and consumers in Bangladesh', Journal of Institutional and Theoretical Economics (JITE), Mohr Siebeck, Tübingen, vol. 127(3), (2004). (An insightful analysis of the impact of communication technologies to the dynamics of local exchange.)

[5] Ramachandran, A.; Feamster, N. Understanding the network-level behavior of spammers. 2006. SIGCOMM 06, Pisa, Italy.

[6] Sahami, M.; Dumais, S.; Heckerman, D.; Horvitz, E. A Bayesian approach to filtering junk email. 1998. AAAI Workshop on Learning for Text Categorization.

[7] http://wiki.apache.org/spamassassin/ThirdPartySoftware.

[8] Kosik, P.; Rajabiun, R. 'Antispam technology impact assessment: fingerprinting versus Bayesian filtering' (September, 2007) contains an updated review of the accuracy and throughput rates of fingerprinting/checksum systems and Bayesian content filters used at the ISP and large corporate server level. http://www.comdomsoft.com/en/antispam/white-papers/.

[9] http://www.jgc.org/tsc.html contains updated data and analysis of history and emerging forms of spam.

[10] http://laptop.media.mit.edu/.

[11] For example, the COMDOM Software Educational Program at: http://www.comdomsoft.com/en/education.