OCTOBER 2006

virus BULLETIN

The International Publication
on Computer Virus Prevention,
Recognition and Removal

CONTENTS

IN THIS ISSUE

SECRET CHAMBER

W32/Chamb is the first virus to infect compiled HTML (CHM) files parasitically. Peter Ferrie reveals all its secrets.

page 4

NO WALK IN THE PARK

Christoph Alme looks at the embedding of arbitrary objects into *Word 2003* XML files and shows why finding them and passing them onto the virus scanner is not such a 'walk in the park' as one might expect.

page 8

COMPARATIVE REVIEW

John Hawes serves up another *VB* comparative – this month, he puts 26 AV products through their paces on *Windows 2000 Server* and finds 18 of them worthy of a VB 100%.

page 10

vbSpam supplement

This month: anti-spam news and events; and Mariusz Kozlowski describes the use of his anti-spam filter based on pattern recognition and artificial intelligence techniques.

ISSN 1749-7027

> *'[DDoS attacks] ... are capable of bringing unprotected organizations to a grinding halt.'*
>
> **Danny McPherson**
> **Arbor Networks**

DDoS: THE RISE FROM OBSCURITY

Six years ago, a flurry of high-profile news articles and research papers reported on the emergence of DDoS attacks. Research released by *Arbor Networks* at the end of September revealed that DDoS attacks are the most significant security threat facing ISPs today.

Arbor's *Worldwide Infrastructure Security Report*, a survey conducted in cooperation with the security operations community of the major ISPs, revealed that 46% of surveyed operators now dedicate more resources to addressing DDoS issues than any other security threat.

Respondents also reported a continued growth in the frequency and magnitude of DDoS attacks. ISPs now regularly experience attacks beyond the capacity of core backbone circuits in the 10–20Gbps range. This trend has been driven globally by a proliferation of broadband Internet connectivity and network convergence.

The rise in DDoS attacks reflects a change in the motivation of cyber criminals – Internet-based threats have taken on a more malevolent and sophisticated nature. DDoS attacks are launched with the sole aim of overwhelming a company's website or server by bombarding them with packets of data, usually in the form of web requests, making the site unavailable to regular users until some fee is paid to the attacker. Unlike single source attacks – which can be stopped relatively easily – the attacker compromises a number of host computers as a command and control infrastructure, which in turn, control thousands of other computers which operate as agents for the assault. These infected host computers ('zombies' or 'bots') flood the victim's website with requests for information – creating a vast and continuous stream of data that overwhelms the target site, thus preventing it from providing normal service.

The cost of a DDoS attack can be substantial – they can last hours, weeks and even months, and are capable of bringing unprotected organizations to a grinding halt. The frequency and size of DDoS attacks is increasing at a dramatic rate. Sixty-four per cent of respondents reported having suffered attacks greater than 4Gbps, and nearly 30% suffered attacks greater than 10Gbps. Yet, despite an average of 40 customer-impacting attacks per month, most attacks go unreported to the police, primarily because there is a widespread belief that such bodies do not have the power or means to assist.

All businesses with an online property must implement the necessary preventative measures to mitigate the threat of a DDoS attack. A comprehensive approach to security must be implemented to combat these attacks. Not only should a multi-layered security strategy be instilled at enterprise level, but companies must also work with their ISPs to ensure that they too have taken preventative measures.

It is essential that companies share information about DDoS attacks if they are to be stopped. Such assaults cannot be fought alone and a collaborative effort is vital. Today this cooperation is achieved through direct back-channel communication between security engineers with interpersonal relationships at different providers, and grassroots efforts by network security vendors such as *Arbor Networks*' Fingerprint Sharing Alliance (FSA). A number of major ISPs have joined the FSA which enables them to share detailed attack information in real time and block attacks closer to the source. Once an attack has been identified by one company, the other ISPs in the Alliance are sent the 'fingerprint', enabling them to identify and remove infected hosts quickly from the network.

Alliances such as the FSA are helping to break down communication barriers and mark a significant step forward in the fight against cyber criminals. However, it is imperative that the culture of cooperation between providers continues to prevail, as it is vital that ISPs work together to prevent and mitigate DDoS attacks and other bot-related activities. However, as the market becomes increasingly competitive, there is a danger that the ISPs will become less cooperative – a trend that will play into the hands of increasingly sophisticated attackers.

Editor: Helen Martin

Technical Consultant: John Hawes

Technical Editor: Morton Swimmer

Consulting Editors:
Nick FitzGerald, *Independent consultant, NZ*
Ian Whalley, *IBM Research, USA*
Richard Ford, *Florida Institute of Technology, USA*
Edward Wilding, *Data Genetics, UK*

NEWS

NEWS ROUND-UP

The top news stories of September were undoubtedly those concerning the VML vulnerability in *Microsoft*'s *Internet Explorer*. Not only was an unofficial patch for the vulnerability released by the Zeroday Emergency Response Team (ZERT), but *Microsoft* also saw fit to break its 'Patch Tuesday' cycle in order to release a patch for the vulnerability, such was the level of concern. ZERT – a group of security experts and reverse engineers from across the computer security industry – was formed last December with the aim of releasing emergency patches when zero-day exploits pose a serious risk to the public and/or the Internet. Its patch was released three days after the discovery of the vulnerability, and *Microsoft* followed swiftly with the official patch just a week after the flaw was published.

Another 'extremely critical' vulnerability in *Microsoft* software was reported by *Secunia* at the end of the month – this time in *Microsoft*'s *PowerPoint*. The software giant drew criticism from others in the AV industry for (allegedly) having known about the flaw, but failing to disclose its details – identities for one of the two trojans known to be exploiting the hole were included in an earlier release of identities for the company's *OneCare* product.

News was more heartening for a number of other AV vendors. *ESET* was named as one of the fastest growing private companies in San Diego, ranking number five in the 'San Diego Fast 100'. The company, which originated in Slovakia, has been expanding its North American presence successfully over the last several years from its North American office based in San Diego.

Trend Micro has also had reason to celebrate, after being declared the 'most valuable Taiwan global brand' for the third year running. *Trend* managed to beat *ASUS ASUSTeK Computer*, *Acer Acer* and *Master Kong Tingyi Holdings* to the top spot, having been valued at US$1.127 billion.

BitDefender, meanwhile, celebrated having been nominated for *RetailVision*'s 'Best Software Product Award' (for *BitDefender 10.0*); having formed a technology alliance with *Internet Security Systems*; and having recorded the company's most successful month in terms of customer growth, product announcements and corporate partnerships.

F-Secure opened a brand new research lab and technical support centre in Kuala Lumpur; *Arbor Networks* was named one of New England's fastest growing technology companies in *Deloitte*'s 'Technology Fast 50 Program'; and analyst firm *Forrester* named *McAfee* as the leading brand in the field in its report on 'Client Security Suites'.

Congratulations one and all.

For daily news updates on the anti-malware industry, point your browser to http://www.virusbtn.com/news/.

Prevalence Table – August 2006

Virus	Type	Incidents	Reports
Win32/Mytob	File	4,849,754	31.97%
Win32/Netsky	File	3,836,891	25.30%
Win32/Bagle	File	2,611,984	17.22%
Win32/MyWife	File	1,624,333	10.71%
Win32/Lovgate	File	532,378	3.51%
Win32/Mydoom	File	507,211	3.34%
Win32/Zafi	File	475,643	3.14%
Win32/Bagz	File	345,794	2.28%
Win32/Parite	File	154,328	1.02%
Win32/Funlove	File	38,896	0.26%
Win32/Mabutu	File	29,638	0.20%
Win32/Bugbear	File	23,743	0.16%
Win32/Klez	File	23,261	0.15%
Win32/Valla	File	21,238	0.14%
Win32/Lovelorn	File	9,979	0.07%
VBS/Redlof	Script	9,557	0.06%
Win32/Agobot	File	9,280	0.06%
Win32/Sober	File	9,219	0.06%
W32/Tenga	File	8,573	0.06%
Win32/Elkern	File	8,265	0.05%
Win32/Maslan	File	6,292	0.04%
JS/Kak	Script	5,341	0.04%
Win32/Darby	File	4,158	0.03%
Win95/Spaces	File	3,581	0.02%
Win32/Kipis	File	3,163	0.02%
Win32/Dumaru	File	2,431	0.02%
W97M/Thus	Macro	1,625	0.01%
Win32/Mimail	File	1,361	0.01%
Win95/Tenrobot	File	1,101	0.01%
Win32/Swen	File	1,031	0.01%
Win32/Reatle	File	787	0.01%
Win32/Bobax	File	685	0.00%
Others[1]		6,290	0.04%
Total		15,167,811	100%

[1] The Prevalence Table includes a total of 6,290 reports across 50 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

CHAMBER OF HORRORS

Peter Ferrie
Symantec Security Response, USA

Amongst the glut of viruses that we see every day, sometimes there is one to surprise us. W32/Chamb is one of those: the first virus to infect compiled HTML (CHM) files parasitically.

WHAT A CHAMPION

Compiled HTML files are *Microsoft*'s way of packaging entire web pages – HTML pages, pictures, sounds, etc. – into a single file that can be transported and viewed offline. The environment for displaying the pages is replicated exactly, since they are passed to the browser by the viewing application. The problem is that the files in the package (properly called 'streams' in this context) are not written to disk prior to being rendered, so anti-malware software is out of luck if it does not support the CHM file format. At this point, it should be noted that the file format is both complex and undocumented, but we have reverse-engineered it. Let's have a look inside.

Compiled HTML files begin with the signature 'ITSF'. That signature stands for 'InfoTech Storage File', which is *Microsoft*'s name for the library that is used to read and write CHM files. Interestingly, when the name is shortened to 'IStorage', we get the name of the programming interface that is used to manage such files. More interestingly, the IStorage interface is the same as the one used by OLE2 files, and which dates back to 1992. The only difference between the OLE2 and CHM implementation is the introduction of the InfoTech Storage System (ITSS) DLL that handles the transparent compression and decompression of the data inside CHM files.

IT'SS LIIKE THISS

Apart from the signature, the ITSF header contains nothing of particular interest. Immediately following it are two directories, divided into two quadwords each. The first quadword in each directory contains the file offset of the data in that block; the second quadword in each directory contains the length of the data in that block.

The first directory block contains the file size, and a flag that is set when a CHM file is first created. The purpose of the flag is to indicate that the file is either a 'work in progress' (when set), or has been finalised (when clear) and no other modifications are allowed.

The second directory block begins with the signature 'ITSP'. It contains information about the number and size of the file list blocks, and the location of the indexes used to access the data quickly in the file list blocks.

The file list blocks follow immediately. They begin with the signature 'PMGL'. The PMGL blocks contain the list of stream names for the streams in the CHM file. There are two types of stream in CHM files: system-data streams and user-data streams. The system-data streams are recognisable because their names begin with two colon characters '::'. The user-data streams are recognisable because their names begin with the forward slash character '/'. The reason for the forward slash character is because these are pathnames. These pathnames are relative to the root directory, which in this case is contained within the CHM file. The stream names are stored in alphabetical order to allow for easy indexing. However, index blocks (which begin with the signature 'PMGI') are added only when there are multiple PMGL blocks.

There are two types of user-data stream: internal and external. The internal user-data streams are recognisable because their names begin with either a hash character '#' or a dollar sign '$'. Anything else is assumed to be an external user-data stream.

Additionally, each PMGL block contains the identity of the previous and next PMGL block, which means that the PMGL blocks can be reordered in peculiar ways, though this would need to be done manually.

CHAMPING AT THE BIT

Each stream name is followed by the dataspace index, the offset of the data relative to the start of the dataspace, and the size of the data. These values are encoded using a seven-bit continuation method: the eighth bit in each byte is used to specify that the value spans multiple bytes. The other seven bits form seven bits of the value, in big-endian format.

The location of the dataspace is found by searching within the stream names for the system-data stream called '::DataSpace/NameList'. After decoding the offset of the NameList, we reach a list of names in zero-terminated Unicode Pascal format (which seems extreme – either zero-terminated or Pascal format alone is sufficient to determine the length of the strings). Only two names should appear in the list: Uncompressed and MSCompressed.

The data in the 'Uncompressed' stream are simply stored. The data in the 'MSCompressed' stream are compressed with *Microsoft*'s LZX compression method, which is also one of the compression methods supported by the CAB file format. However, unlike in CAB format where each file is compressed individually, CHM files compress all of the streams as though they were a single block (a so-called 'solid' archive). While this can increase the compression ratio significantly, it can also increase the time required to extract individual items significantly. *Microsoft* compromised between these two characteristics, by breaking the single large block into smaller blocks of fixed size and compressing those individually. The information about these smaller blocks is stored in a 'reset table' (see below).

In order to decompress the data in the 'MSCompressed' stream, some additional streams must be retrieved first. One of those is the '::DataSpace/Storage/MSCompressed/ControlData' stream, which contains the information about the LZX compression parameters. The other two streams are '::DataSpace/Storage/MSCompressed/Transform/{7FC28940-9D31-11D0-9B27-00A0C91E9C7C}/InstanceData/ResetTable' and '::DataSpace/Storage/MSCompressed/Content'. The 'ResetTable' stream is used to control the periodical resetting of the decompression state. By resetting the decompression state periodically, it no longer becomes necessary to decompress the entire large block to reach an arbitrary file. The reset table allows one to begin the decompression at the nearest reset state prior to the required offset, which can make the decompression faster for some items. Finally, the 'Content' stream contains the compressed data.

As an aside, there is an interesting extension in the '::DataSpace/Storage/MSCompressed/Transform/List' stream. It appears that it was intended to provide support for customised decompression and/or decoding layers, but the stream data in existing CHM files are malformed – the stream contains only a partial GUID in Unicode character form, because the stream is too small to contain a complete GUID. Judging by the stream length, it was probably intended to hold an ASCII string and some small additional data.

CHARM OFFENSIVE

So what does all of this have to do with W32/Chamb? Actually, very little – since the virus makes use of the IStorage interface, all of these details are handled by the ITSS DLL, and all the virus has to do is call a few functions to perform the required actions, much as any other file-infecting virus does for an ordinary file system.

In any case, the virus begins by searching the current directory for CHM files to infect. The infection marker is that the file has the read-only attribute set. Otherwise, the file is considered a candidate for infection.

If the virus finds a file to infect, it creates a new file called 'c' in the current directory, which is used as a temporary working file during the infection process. The temporary file is required because the ITSS DLL does not allow writing to a 'finalised' CHM file.

The virus enumerates all of the storages and streams in the file to infect, and writes each of them to the temporary file. Anything within the original file that is neither a storage nor a stream will be discarded during the infection process. The ITSS DLL decompresses the streams automatically as they are read, and compresses them as they are written.

For any stream whose name ends with '.HTM', the virus will append an object reference to a stream called '.exe'. Upon completion of the enumeration, the virus will add itself as the stream called '.exe', thus ensuring that it will be called whenever a page is viewed in the infected CHM file.

The ITSS DLL sorts the storage and stream names as they are added. The result is that even though the '.exe' stream is the last to be added, thanks to its name, it will be among the first in the PMGL blocks.

After adding the '.exe' stream, the virus will copy the 'c' file over the original file, set the file date and time stamps to those of the original file, and set the read-only attribute to mark the file as infected.

Upon completion of the file enumeration, the virus simply exits. The virus contains no payload; it is simply yet another proof of concept from a virus author who specialises in producing them.

THE CHASM OPENS WIDE

Compiled HTML files have been a favourite of malware authors for several years already, but until now only in static form. For the most part, they have been trojans that downloaded other malware, but at least one family of worms (W32.Blebla) used a CHM file in order to spread. Now that we have a parasitic virus for CHM files, the advice is the same as when the first WinHelp infectors appeared in 1999: don't press F1!

W32/Chamb

Type:	Parasitic direct-action infector.
Infects:	*Windows* CHM files.
Self-recognition:	Read-only attribute is set.
Payload:	None.
Removal:	Delete infected files and restore them from backups.

OPINION

AV TESTING SANS VIRUS CREATION

David Harley
Independent researcher, author and consultant

LETTER TO ALAN PALLER, DIRECTOR OF RESEARCH, THE SANS INSTITUTE

Dear Mr Paller,

Thank you for letting us know that the *Consumer Reports* methodology for testing anti-virus software by creating new variants [1] is fair and rigorous.

We of the anti-virus brotherhood are always grateful for crumbs of enlightenment from the table of the Great and the Good of the security establishment far beyond the walls of our own little ivory towers. Nevertheless, we don't believe you were altogether correct in this instance.

Let's start with some admissions.

What virus scanners do best is find (and hopefully remove) known viruses. They are *not* so good at detecting and removing unknown viruses. The model of adding definitions to detect each virus as you find out about it has a fatal flaw: it means that the anti-virus vendor is always 'playing second' to the malware author. And yes, it is an approach that works better for the vendor's revenue stream than for the consumer's proactive protection. However, it does a worthwhile job of removing malware that has already kicked in, and of keeping many PC systems clear of the common viruses still in circulation.

But that is not what modern scanners do. At least, it's not *all* they do. They use a variety of proactive techniques, which means that they're capable of detecting some unknown viruses as they appear, and before they've been analysed in the vendor's lab. So when you stated in the *SANS Newsbytes* newsletter that anti-virus vendors don't find and block viruses quickly, you're working from a model that is many years out of date. You also seem to imply that anti-virus vendors are still updating their product every few weeks or months (as was the case in the past), whereas most vendors now update their products *at least* daily, and usually make detection available within hours of an in-the-wild virus being reported to them.

Of course, heuristic analysis and generic signatures don't catch 100% of unknown malware, or anything like it. In fact, since malware authors dedicate a serious amount of R&D time to patching their creations until the main anti-virus products *don't* detect them, anyone who thinks that up-to-date scanners can offer perfect protection needs a reality check.

That's one of the reasons why savvy security administrators use AV scanners as just one component of a multilayered defence strategy, as a supplement to other generic/proactive approaches. They use them to clean up outbreaks where proactive defences fail, and to ensure that the many malicious programs still circulating months or years after their discovery don't get a foothold on the sites under their wing. And this is why anti-virus is really a multi-functional product nowadays.

We (anti-virus vendors, independent researchers and testers, canny AV users like the members of AVIEN, and so on) already know all this, so this test isn't really 'important product improvement research', is it? But it does point to a massive failure on our part. We have tried, but failed to educate both the general public and the wider security community about what anti-virus really is, how it really works, and how important it is to use non-reactive defences and *very* rigorous testing practices.

It's not so surprising that we've failed to educate home users, when there is so much misinformation to compete with. But clearly we still can't expect a fair hearing from other sectors of security, either. And when *they* get it wrong, they mislead a whole load of other people.

HEURISTIC TESTS – RIGHT OR WRONG?

So, is it wrong to test a scanner's ability to detect heuristically? Of course not, if it's done competently. Was this a competent test? Well, we don't really know. Only the barest bones of their methodology has been published. Since these people are working outside the AV research community – which is far more collaborative than anyone outside it will ever believe – we really don't know whether they know any more about this specialist area than the average end user.

Back in the days when I was less easily depressed, I tracked some of the 'tests' that were circulating at that time. Testers were using collections of alleged viruses found on 'vx' websites. These were known to contain large numbers of garbage files such as random text files, snippets of source code, intendeds (viruses that couldn't actually replicate, and therefore weren't viruses), corrupted viruses that couldn't work, programs generated by virus generators which may or may not have been viable viruses, the infamous Rosenthal utilities, and (my particular favourite) 'virus-like' programs (I've often wondered what that meant). Even then, testers were trying to test a scanner's heuristic ability by generating 'variants'. Inserting snippets of virus code at random places in a test file. Patching presumed infected files in random places. Changing text strings found in virus bodies on the assumption that that was what scanners were looking for.

Concealing them in objects like *Word* documents where they could never be found naturally, or 13 levels down in an encrypted archive. What they didn't do, almost without exception, was make any attempt to check that what they were testing with was, in every case, a valid, working virus.

Perhaps the *Consumer Reports* test was better than that, though a quote from Evan Beckford suggests that virus generators may have been used – and these are notoriously unreliable when it comes to producing viable viruses. Unless more data is published on the methodology used for these tests, or the test collection is submitted for independent verification, how will we know whether the test is valid?

For all we know, the collection could consist of 5,500 garbage files. (I don't know whether it is significant that most of the files generated were not actually used.) Just think about that scenario for a moment. If this were the case, the scanners that scored the highest number of detections would be hitting high volumes of false positives. If even some of the test files were invalid, you wouldn't just be testing heuristic capability any more: you'd be testing the *sensitivity* of the products' heuristics, and their whole detection philosophy. Perhaps that's a valid test objective, but not the one that seems to have been intended.

All this, of course, presupposes that all the scanners tested were configured appropriately and consistently. In real life, some of the many amateur sites that run new malware against multiple scanners and publish comparative results for that malware have been known to penalize individual products by using out-of-date definitions (or signatures, if you must) or over-conservative settings. Again, we simply don't know how well this was done. I will, for the purpose of this note, assume that at the very least all the necessary precautions were taken to avoid the inadvertent release of these variants beyond the testing labs (as is claimed).

IS IT WRONG TO CREATE TEST VIRUSES?

Is it wrong to create new test viruses and variants? The anti-virus industry is very leery of creating viruses for any purpose: some anti-virus researchers won't do that under *any* circumstances, and probably none will do so when it isn't necessary. It's ironic that half the world is convinced it is members of the AV companies that write the malware, while the industry itself obsesses about keeping its hands clean, not employing virus writers and so on.

I won't say it is never necessary to write a new variant, or replicative code for testing and development purposes: that is a decision that is best left to the individual researcher. But it is not necessary to write viruses to test anti-virus heuristics. A less contentious approach is the retrospective test, where you 'freeze' a scanner without updates for a period of time then test with a batch of malware that has appeared subsequent to the cut-off point. This needs to be done very carefully, but it avoids the ethical conflicts and many of the technical uncertainties, and it is a better test of a scanner's capabilities than throwing at it objects that may or may not be valid viruses.

IT'S ALL IN THE LITERATURE

Given our previous history of disagreement over virus issues, you may be surprised to know that I still think *SANS* does some excellent work. However, your commentary suggests that, like so many security gurus, you may have succumbed to the inability to say 'I don't know enough about that speciality to make a useful or valid comment.' Perhaps you need to catch up with some of the literature on testing, maintaining a collection (Vesselin Bontchev's paper [2] is still very relevant), and so on. You might also want to look up a very old (but still too relevant for comfort) article by Alan Solomon on how to bias a comparative review [3], as well as Igor Muttik's very recent response to the *CR* test [4].

Robert Slade and I wrote a long chapter on testing issues in our book on viruses [5]. (Although I was lukewarm about retrospective testing then, I've seen it work well in practice since.) You could consider checking out some of the organizations that offer competent independent testing, such as *AV-Test.org*, *av-comparatives.org*, *ICSA Labs* and *Virus Bulletin*. Have you read Peter Ször's book yet, I wonder [6]?

The anti-virus industry is far from perfect. But it includes some amazingly competent people, some of whom have thought long and hard about the testing issue, and work closely with equally competent independent researchers and testers. Some of them may even know more than people who don't work in or on the fringes of the industry. Just a thought.

REFERENCES

[1] See *Virus Bulletin* September 2006, p.2.

[2] Bontchev, V. 1993. http://www.people.frisk-software.com/~bontchev/papers/virlib.html.

[3] http://www.softpanorama.org/Malware/Reprints/virus_reviews.html.

[4] http://www.avertlabs.com/research/blog/?p=71.

[5] Slade, R.; Harley, D.; Gattiker, U. Viruses Revealed. 2001.

[6] Ször, P. The art of computer virus research and defense. Symantec Press. 2005.

FEATURE

SCANNING EMBEDDED OBJECTS IN WORD XML FILES

Christoph Alme
Secure Computing Corporation, Germany

Earlier this year an article by Jan Monsch [1] showcased how rarely-known, 'alternative' *Microsoft Word* file formats can be used to transport malware to end users' PCs. While this doesn't pose an imminent threat to desktop PCs running an on-access anti-virus scanner, their counterparts running at the network perimeter – for example on web and email gateways – will have to go the rocky road to inspect these alternative formats as well. Even with on-access scanners deployed on corporate end-user PCs, this remains a requirement for gateway anti-virus scanners, since users tend to question their scanners whenever they see their desktop's scanner block something that the company gateway has allowed through.

Of course, one can argue that activating embedded malware, for example as an OLE object, still needs a significant amount of user interaction. But we can't rely on that fact to stop it happening, as we simply cannot be sure that end users will not be tricked into activating it through the clever use of social engineering (possibly 'targeted' social engineering).

Scanning VBA macros in *Word 2003* XML files has been covered in [2], so this article will focus on the embedding of arbitrary files into *Word 2003* XML files, giving an overview of how they can be found and passed on to the virus scanner. It also shows why this is not such a 'walk in the park' as one might at first expect. (If it has the magic 'XML' in its name, it ought to be a breeze to parse, oughtn't it?)

THE 'WORDML' XML SCHEMA

All *Office 2003* XML schema files [3] start with an mso-application processing instruction. The actual *Office* application is denoted in its progid attribute, such as 'Word.Document' in our case (and only the version-independent ProgID works here).

The WordprocessingML schema's root element is named wordDocument and since, by default, *Word* defines a 'w' namespace for its schema, it is actually <w:wordDocument>. But XML namespaces can be defined with any name – just use xyz, for example:

```
<xyz:wordDocument xmlns:xyz="...> ...
<xyz:docOleData> ...
```

Word will still happily load and render the document correctly. But using a namespace that is deviant from 'w' causes a decrease in the number of anti-virus scanners (as hosted on *VirusTotal* [4]) that can detect an embedded, ZIP-archived EICAR test virus in *Word 2003* XML files from three (as was the case in [1]) to one (at the time of writing this article).

DECODING EMBEDDED OBJECTS

To determine quickly whether a WordML file has any embedded objects at all, the root element's embeddedObjPresent attribute can be checked for containing 'yes':

```
<w:wordDocument ...
     w:embeddedObjPresent="yes" ...>
```

Otherwise, *Word* does not render the document at all in case it does contain embedded objects, and complains.

So this allows a scanner to decide relatively quickly whether it has to parse the whole document at all. When it has to, it should look for <docOleData> elements. Each such element has one child element named <binData>, but can also have any other child element that may simply be ignored by *Word*. Since XML processors are case-sensitive – and the one used by *Word* is no exception here – it may even have a <binData> child element that *Word* will use, next to a <BiNdAtA> child element or similar that *Word* will ignore. Therefore, a construct like this:

```
<w:docOleData>
<w:BinDaTA w:name="oledata.mso">
Bla bla bla
</w:BinDaTA>
<w:binData w:name="oledata.mso">
0M8R4KGxGuEAAAAAAA
... more base64-encoded data here ...
wMAAAAAAAD/DAAA
</w:binData>
</w:docOleData>
```

could allow scanning of the actual embedded object, contained in the 'real' <binData> element, here to be evaded if a scanner only checks the very first <binData> child element of a <docOleData> element without caring for the case of its tag name.

Next, we cannot rely on the <binData> element's name attribute; it does not need to contain 'oledata.mso' for *Word* to treat it as an embedded object – any other name, such as <w:binData w:name="helloWorld.mp3"> , will do the job just as well. Using a deviant name allows the results of [1] for *Word 2003* XML files to be decreased to two (at the time of writing this article).

The <binData> element's data is encoded in base64 (plus intermittent linebreaks). It may contain entity or character references that get resolved by the XML processor. So *Word* will not have any problem rendering a document with a <binData> element's data encoded, for example, as:

```
<w:binData w:name="oledata.mso">
0M8R4KGxGuEAA&#65;AAA&#65; ...
```

But using such character references here allows the results of [1] for *Word 2003* XML files to be decreased to zero (at the time of writing this article). The same applies when inserting comments into the element data, like

```
<w:binData w:name=”oledata.mso”>
...
EAA<!— comment —>AAgAAAAEAAAD+// ...
```

Now let's have a look at the actual data. After base64 decoding, we'll get an OLE Compound file with one stream underneath its \Root Entry storage, named as specified in the XML file's associated OLEObject element's ObjectID attribute:

```
<o:OLEObject Type=”Embed” ...
      ObjectID=”_1218624971"/>
```

Note that after base64 decoding, the decoded content is not padded up to the big block size alignment specified in its OLE Compound file header (512 bytes as usual).

The OLE2 stream's data starts with an unsigned 32-bit field denoting the uncompressed size of the following data. The data that follows is compressed using the deflate compression algorithm, but of course we first verify the utilized compression method by checking that the lower nibble of the compressed data's first byte is 8 (= Z_DEFLATED).

After uncompressing, we find yet another OLE Compound Structured Storage file. Seeing the 'D0CF11E0' signature appear once more may remind you of the myth of Sysiphus, but hold on – we can already see some light at the end of the tunnel. If the embedded OLE object supports in-place activation, like a PDF document or *Flash* animation for example, we now find its 'contents' stream directly underneath the \Root Entry and, depending on the actual object's persistence strategy, it may even start with the 'raw' data.

When dealing with an instance of the so-called 'Package' object, which is used to embed arbitrary files and does not therefore support in-place activation, we find its raw data in the '\.Ole10Native' stream underneath the \Root Entry. As usual, it is prefixed by a header that consists basically of size fields, a display name and two path names, all in ASCIIZ. At last – we have unveiled the data of interest!

NOTHING IS AS CONSISTENT AS CHANGE

While the *Excel 2003* XML schema does not allow for embedding of OLE objects (or VBA macros), and *PowerPoint 2003* does not have an 'alternative', XML-based file format at all, the upcoming *Microsoft Office 2007* release will change this and more. It will bring a new format called 'Office Open XML', which is supported by *Word*, *Excel*, *PowerPoint* and other *Office* applications. The new file extensions are .DOCX, .XLSX and .PPTX for the default, not 'macro-enabled' document formats, while their 'macro-enabled' counterparts will use the file extensions .DOCM, .XLSM and .PPTM, respectively. Note that Office Open XML is planned to be the default format, so its prevalence can be expected to increase significantly over the coming years. At the time of writing, *Office 2007* is available as Beta 2 and therefore details may still be subject to change.

Office Open XML files consist of a ZIP archive containing various XML files and the embedded OLE objects as separate archive members called 'oleObject1.bin' and so on. By default, they are located in the 'embeddings' archive folder. But embedded objects don't have to be stored here: a new indirection, called 'relationships', defines where an embedded object's data is stored within the archive. To find out which archive members represent embedded objects, or simply to prevent scanning the whole archive looking for embedded objects, you'll have to start (using *Word* as an example) by parsing the '/word/document.xml' file, looking for <OLEObject> elements (the <wordDocument> element's previously mentioned 'embeddedObjPresent' attribute seems to have vanished):

```
<w:object>
...
<o:OLEObject Type=”Embed” ...
      ObjectID=”_1218959120"
      r:id=”rId5" />
  </w:object>
```

The attribute of interest is the relationship identifier, 'r:id'. We can now look up the relationship 'rId5' in the '/word/_rels/document.xml.rels' file to find out where this embedded object's data is stored:

```
<Relationship Id=”rId5" ...
      Target=”embeddings/oleObject1.bin” />
```

So now we have the path and filename of the archive member comprising the embedded object's data. It appears to be an OLE Compound file containing either a '\.Ole10Native' stream or a 'Contents' stream in the usual formats.

To our relief, the outlined embedding approach is consistent among the upcoming *Word*, *Excel* and *PowerPoint* formats – only the names of archive folders, files, XML elements and attributes differ (as of Beta 2, of course). Users of *Office XP/2003* will also be able to use the new formats via the converters already available at [5].

REFERENCES

[1] http://handlers.sans.org/dwesemann/alternativ_word_formats_v2.0.pdf.

[2] http://www.virusbtn.com/pdf/magazine/2003/200302.pdf.

[3] http://www.microsoft.com/office/xml/default.mspx.

[4] http://www.virustotal.com/en/indexf.html.

[5] http://www.microsoft.com/office/preview/beta/converter.mspx.

COMPARATIVE REVIEW

WINDOWS 2000 SERVER

John Hawes

My second time running the *Virus Bulletin* comparative review offered a wildly different experience from the first; whereas August's *Novell NetWare* test drew a mere eight entries, this month saw a bumper 26 products vying for the award. Many of these were entirely new to me, and two were first-timers in the *VB* tests. Both from China, newcomers *Kingsoft AntiVirus* and *Greatsoft Virusclean* were added to the rash of more familiar names with a mixture of excitement and trepidation on my part.

TEST SETS AND PLATFORM

The platform for the test was *Windows 2000 Server*, just barely on the edge of supported status and almost certainly seeing its last outing in the *VB* lab. The aging operating system was succeeded several years ago by *Windows 2003 Server* – which will, apparently, soon be made obsolete itself by the forthcoming and much hyped *Windows Vista*. Patched with the most recent service pack (the three-year-old SP4), setting up the test machines with *Windows 2000* was a familiar and trouble-free experience.

The In the Wild (ItW) test set was aligned with the June 2006 WildList, which saw the addition of a sprinkling of familiar Mytob and Bagle variants, along with a few new names. W32/Areses, W32/Rontokbro and W32/Banwarum are fairly standard email worms with a few nasty AV-disabling and general anti-tampering devices thrown into some variants.

On top of the additions to the WildList, the clean set was expanded somewhat, but the most significant change this month was a handful of new viruses in the polymorphic test set, all of which have been around for some time, and rarely trouble users these days. However, although most are limited to older operating systems, as infectious viruses they all have the chance of making a nuisance of themselves should they ever make their way onto a vulnerable machine. Of the batch, the venerable W95/Zmorph is perhaps the most notable, with its highly metamorphic nature aimed at baffling the detection engines of its day. Let's see how the modern-day versions fared.

AhnLab V3Net for Windows Server 6.0

ItW	100.00%	**Macro**	98.97%
ItW (o/a)	100.00%	**Macro (o/a)**	98.97%
Standard	97.13%	**Polymorphic**	90.48%

Oct 2006 VIRUS BULLETIN 100% www.virusbtn.com

AhnLab's product installed in a straightforward fashion, but I found the GUI a little uncomfortable at first, as I made copies of the default jobs available in order to tweak the configuration to suit my needs.

The progress screen for the on-demand scanner amused me, with its row of folder icons progressing past a magnifying glass, which sucked green bugs out of them as they went by. I was less amused by the logging, which seemed not to record the paths of infected files, and by the on-access scanner, which appeared not to block any files from being opened. However, when configured to delete infected items it did the job – after slowly building a list of all infections spotted, and then going through deleting them once the delete option had been selected.

After all this, although much was missed in the zoo collections, all the WildList viruses were spotted, and no false positives were alerted on in the clean set, thus earning *V3Net* a VB 100% award. The product also did rather well in the speed tests.

Alwil avast! v.4.7

ItW	100.00%	**Macro**	99.56%
ItW (o/a)	100.00%	**Macro (o/a)**	99.56%
Standard	98.74%	**Polymorphic**	89.90%

Oct 2006 VIRUS BULLETIN 100% www.virusbtn.com

The piratical note in *avast!*'s title warned me to expect no mercy, and the greyed-out 'Back' button preventing me from retracing my steps after accepting the EULA felt a little like stepping out onto the plank. The multi-pane GUI was reasonably usable, and the on-demand and speed tests were carried out with ease and reasonable success, although several of the new polymorphic viruses were missed. On-access testing proved more difficult, as files were not blocked on opening, but copying them onto the machine and having them deleted brought results. On several tries the product got snarled up with the large numbers of warnings it was issuing and its GUI froze, requiring forcible shutting down. In the real world, however, such a problem is unlikely to occur, and with only a single file in the clean set labelled a 'Joke' to report, *avast!* qualifies comfortably for the VB 100% award.

Avira AntiVir Windows Server 2003/2000/NT v.6.35

ItW	100.00%	**Macro**	99.93%
ItW (o/a)	100.00%	**Macro (o/a)**	99.93%
Standard	100.00%	**Polymorphic**	96.37%

Avira’s product was one of the plethora I was trying for the first time, and it rather pleased me.

The installation process offered no difficulties, although an image of what seemed to be a man holding a red umbrella indoors gave me reason to wonder how lucky *Avira* would be. The GUI reassured me with its pared-down, vaguely techie feel, simple icon-style graphics and text-heavy displays and menus. The progress display, updating itself every 50–100 files scanned, gave an impression of thoroughness, and results in the first few tests were admirable.

A few of the new polymorphic viruses went unrecognised, but this was not too surprising. It was in the clean set that *Avira*’s luck ran out, however, and with two false positives recorded, *AntiVir* misses out on its VB 100%.

BitDefender Antivirus v.10

ItW	100.00%	**Macro**	96.69%
ItW (o/a)	100.00%	**Macro (o/a)**	96.69%
Standard	99.27%	**Polymorphic**	97.02%

BitDefender was another product I sampled for the first time this month, and I was pleased to see mention of the VB 100% award proudly presented on the second screen of the installation process, as well as in the readme. I also found the slick, simple, oddly flat-looking GUI easy on the eye and untaxing on the brain, although the little black block indicating that the on-access component is functioning was a little spooky.

The product did well in both the WildList and zoo collections, missing nothing in the ItW test set and not a great deal in the other sets, but sadly it was let down by yet another false positive in the clean test set, which spoiled *BitDefender*’s chances of adding another VB 100% award to its collection.

CA eTrust 8.0.403.0 (InoculateIT engine)

ItW Overall	100.00%	**Macro**	99.90%
ItW Overall (o/a)	100.00%	**Macro (o/a)**	99.51%
Standard	99.82%	**Polymorphic**	97.23%

eTrust’s professional-looking installation, with its requirement to scroll through several lengthy EULA segments and a lengthy survey of personal information, was familiar to me from the *NetWare* tests last time around, as was the browser-based GUI. This didn’t work as well as I remembered, indeed refusing to initiate an on-demand scan, which rather scuppered me until I learned that the browser installed with *Windows 2000 – Internet Explorer 5.0* – was not supported by the product, and *IE* version 6 SP1 was required.

With the required version of *IE* installed, the only remaining issue was with the logs – which, being large and filled with notices of infections, were rather slow to open up in the display window. They were also not exportable to plain text for parsing, but that annoyance was soon worked around to find good scores all round. Of course, since *InoculateIT* is not the default for the product, it does not qualify for the VB 100% award.

CA eTrust 8.0.403.0 (Vet engine)

ItW	100.00%	**Macro**	99.82%
ItW (o/a)	100.00%	**Macro (o/a)**	99.84%
Standard	99.96%	**Polymorphic**	94.26%

When run with the *Vet* engine, *eTrust* missed slightly more of the new polymorphic viruses than when run using the *InoculateIT* engine, and was also a fraction slower in some of the throughput tests, but still put in a strong performance, amply qualifying for another VB 100% award.

Oct 2006
100%
VIRUS BULLETIN
www.virusbtn.com

CAT Quick Heal 2006 v.8.0

ItW	100.00%	**Macro**	98.23%
ItW (o/a)	100.00%	**Macro (o/a)**	97.96%
Standard	96.51%	**Polymorphic**	87.07%

Quick Heal surprised me during installation by carrying out an automatic scan of memory and system files, before requesting a reboot to complete the installation.

Oct 2006
100%
VIRUS BULLETIN
www.virusbtn.com

Once installed, the GUI presented to me was simple and slick, although it seemed to offer no method of disabling the on-access protection; this, I soon found, was achieved by right-clicking the icon in the system tray.

On checking the scan results, I was a little confused that the timings seemed to have had an hour added to each, resulting in many scans claiming to have finished 55 minutes in the future. However, I was soon able to correct for this, and found the scanning speeds reasonable enough to justify the product’s title. Despite missing a fair chunk of the zoo viruses, *Quick Heal* detected everything in the ItW test set, while generating no false positives in the scan of the clean set, thereby earning its VB 100% award comfortably.

Command Authentium AntiVirus for Windows 4.93.8

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	99.98%	**Polymorphic**	99.93%

Authentium's product installed zippily, and presented me with a small and simple GUI. Things seemed to be progressing nicely with on-demand scanning until I attempted to save the log produced; while a log was indeed saved, it seemed to include only the last 1,000 lines of the full scan report – all of which were still viewable within the product's GUI. Resorting once more to the deletion method, *Authentium* did excellently on the infected files, but was let down when a file in the clean set was flagged as suffering an infection, which it suggested was possibly a new variant of a known threat. This was enough to deny the product the VB 100% award this time around.

Doctor Web Dr.Web Scanner for Windows v.4.33.2

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	98.08%

Dr.Web installed in a sleek and stylish fashion, and after a reboot and several automatic scans of memory and system files, I found the GUI equally slick. I found my way around it quickly – although the 'SpIDerGuard' on-access component of the product seemed not to have started itself – and it charged through the tests with little difficulty.

Oct 2006 100% VIRUS BULLETIN www.virusbtn.com

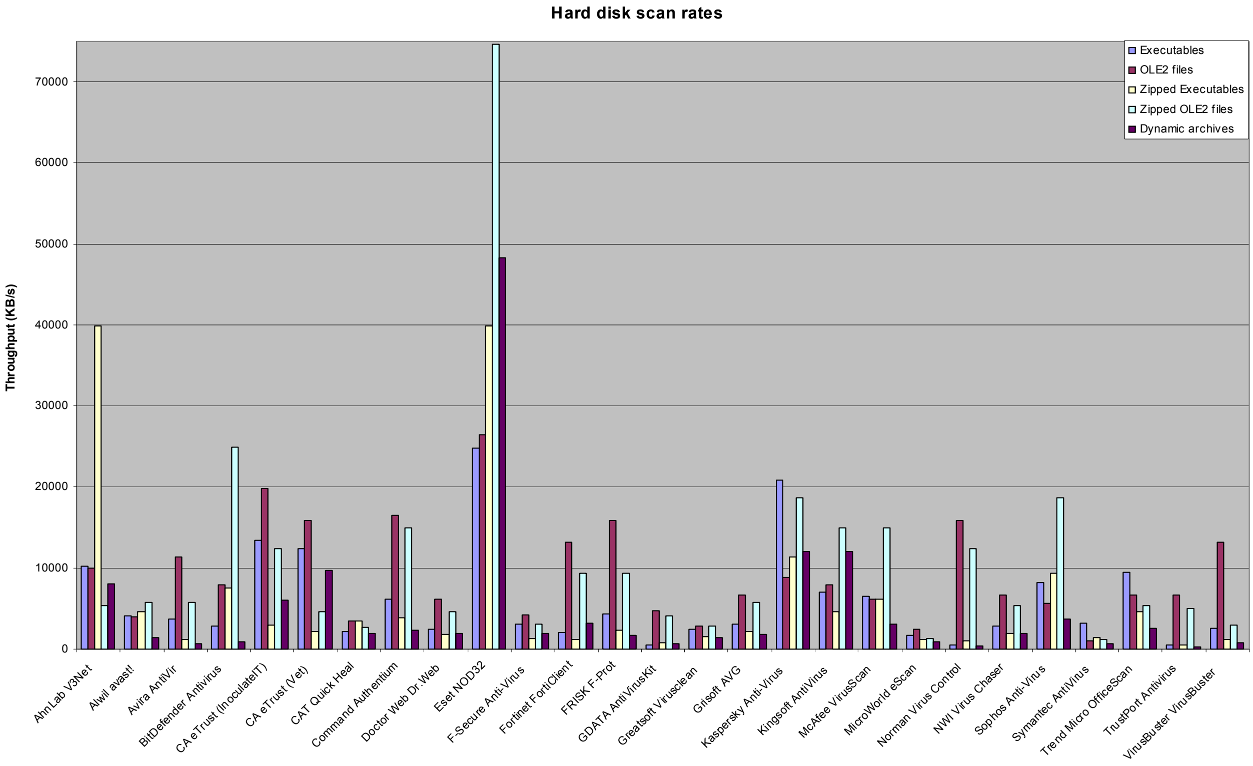
With only a single set of polymorphic samples missed, and a few zips in the standard set ignored on access, *Dr.Web* put in an impressive performance – no false positives were produced in the clean set, allowing *Dr.Web* to gain its VB 100% award with ease.

ESET NOD32 2.5

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

NOD32 also impressed me, with a very simple and rapid installation process and a simple, clear GUI – although I imagine anyone who isn't familiar with the product may be a little baffled by the numerous modules labelled only as 'AMON', 'IMON' etc.

Oct 2006 100% VIRUS BULLETIN www.virusbtn.com



Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables			Zipped OLE Files			Dynamic files		
	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]
AhnLab V3Net for Windows Servers 6.0	63.0	10216.6		8.0	9916.7		4.0	39854.1		14.0	5329.1		6.0	8040.4	
Alwil avast! v.4.7	158.0	4073.7	[1]	20.0	3966.7		35.0	4554.8		13.0	5739.0		34.0	1418.9	
Avira AntiVir Windows Server 2003/2000/NT v. 6.35	176.0	3657.1	1	7.0	11333.4		132.0	1207.7	1	13.0	5739.0		81.0	595.6	
BitDefender Antivirus v.10	224.0	2873.4	1	10.0	7933.4		21.0	7591.3		3.0	24869.2		54.0	893.4	
CA eTrust 8.0.403.0 (InoculateIT)	48.0	13409.3		4.0	19833.4		55.0	2898.5		6.0	12434.6		8.0	6030.3	
CA eTrust 8.0.403.0 (Vet)	52.0	12377.9		5.0	15866.8		75.0	2125.6		16.0	4663.0		5.0	9648.5	
CAT Quick Heal 2006 v.8.0	288.0	2234.9		23.0	3449.3		46.0	3465.6		28.0	2664.6		25.0	1929.7	
Command Authentium AntiVirus for Windows 4.93.8	105.5	6102.1	1	4.8	16527.9		41.7	3822.9		5.0	14921.5		20.5	2353.3	
Doctor Web Dr.Web v.4.33.2	264.0	2438.1		13.0	6102.6		90.0	1771.3		16.0	4663.0		25.0	1929.7	
Eset NOD32 2.5	26.0	24755.7		3.0	26444.6		4.0	39854.1		1.0	74607.5		1.0	48242.6	
F-Secure Anti-Virus for Windows Servers v.5.52	213.0	3021.8		19.0	4175.5		129.0	1235.8		24.0	3108.6		25.0	1929.7	
Fortinet FortiClient 3.0.001	318.0	2024.1		6.0	13222.3		139.0	1146.9		8.0	9325.9		15.0	3216.2	
FRISK F-Prot v.3.16f	147.0	4378.6	[1]	5.0	15866.8		68.0	2344.4		8.0	9325.9		28.0	1723.0	
GDATA AntiVirusKit 16.0.7	1208.0	532.8	1	17.0	4666.7		192.0	830.3		18.0	4144.9		73.0	660.9	
Greatsoft Virusclean v.2.0.3286.3	260.0	2475.6	5	28.0	2833.3	1	101.0	1578.4	2	27.0	2763.2	1	35.0	1378.4	2
Grisoft AVG Anti-Virus 7.1	207.0	3109.4		12.0	6611.1		72.0	2214.1		13.0	5739.0		27.0	1786.8	
Kaspersky Anti-Virus 5.0 for Windows File Servers v. 5.0.77.0	31.0	20762.8		9.0	8814.9		14.0	11386.9		4.0	18651.9		4.0	12060.7	
Kingsoft AntiVirus 2006 v.7.1	91.0	7073.1	3	10.0	7933.4		35.0	4554.8	2	5.0	14921.5		4.0	12060.7	
McAfee VirusScan Enterprise 8.0	99.0	6501.5		13.0	6102.6		26.0	6131.4		5.0	14921.5		16.0	3015.2	
MicroWorld eScan Internet Security for Windows 8.0.673.1	392.0	1642.0		32.0	2479.2		143.0	1114.8		60.0	1243.5		55.0	877.1	
Norman Virus Control v.5.82	1211.0	531.5		5.0	15866.8		154.0	1035.2		6.0	12434.6		123.0	392.2	
NWI Virus Chaser v.5.0a	227.0	2835.5		12.0	6611.1		82.0	1944.1		14.0	5329.1		25.0	1929.7	
Sophos Anti-Virus v.6.03	79.0	8147.4		14.0	5666.7		17.0	9377.4		4.0	18651.9		13.0	3711.0	
Symantec AntiVirus 10.0.0.359	205.0	3139.7	[1]	79.0	1004.2		115.0	1386.2		67.0	1113.5		77.0	626.5	
Trend Micro OfficeScan Corporate Edition v.7.3	68.0	9465.4		12.0	6611.1		35.0	4554.8		14.0	5329.1		19.0	2539.1	
TrustPort Antivirus 2.01.855	1129.0	570.1	1	12.0	6611.1		302.0	527.9		15.0	4973.8		180.0	268.0	
VirusBuster VirusBuster 2006 for Windows Servers v.5.2	256.0	2514.3	[1]	6.0	13222.3		143.0	1114.8		25.0	2984.3		59.0	817.7	

I also spent some moments figuring out how to export logs, as the 'log' section of the GUI seemed to have no function. This brief dithering on my part took up most of the testing time, as the product powered through the scans in stunning time, and effortlessly detected everything offered to it without false positives, earning yet another VB 100% award for its work.

F-Secure Anti-Virus for Windows Servers v.5.52

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	99.85%	**Polymorphic**	100.00%

On-access tests	ItW File		Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%
AhnLab V3Net for Windows Servers 6.0	0	100.00%	47	98.97%	626	90.48%	57	97.13%
Alwil avast! v.4.7	0	100.00%	18	99.56%	385	89.90%	34	98.14%
Avira AntiVir Windows Server 2003/2000/NT v. 6.35	0	100.00%	3	99.93%	0	96.37	150	100.00%
BitDefender Avtivirus v.10	0	100.00%	13	96.69%	37	97.02%	10	99.27%
CA eTrust 8.0.403.0 (InoculateIT)	0	100.00%	4	99.51%	42	97.23%	1	99.82%
CA eTrust 8.0.403.0 (Vet)	0	100.00%	3	99.84%	103	94.26%	1	99.96%
CAT Quick Heal 2006 v.8.0	0	100.00%	86	97.96%	602	87.07%	153	93.00%
Command Authentium AntiVirus for Windows 4.93.8	0	100.00%	0	100.00%	2	99.93%	4	99.67%
Doctor Web Dr.Web v.4.33.2	0	100.00%	0	100.00%	9	98.08%	3	99.69%
Eset NOD32 2.5	0	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus for Windows Servers v.5.52	0	100.00%	0	100.00%	0	100.00%	3	99.85%
Fortinet FortiClient 3.0.001	0	100.00%	0	100.00%	277	84.47%	0	100.00%
FRISK F-Prot v.3.16f	1	99.85%	0	100.00%	8	99.91%	4	99.49%
GDATA AntiVirusKit 16.0.7	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Greatsoft Virusclean v.2.0.3286.3	1	99.85%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG Anti-Virus 7.1	0	100.00%	3	99.93%	414	82.59%	30	98.41%
Kaspersky Anti-Virus 5.0 for Windows File Servers v. 5.0.77.0	0	100.00%	0	100.00%	0	100.00%	2	99.69%
Kingsoft AntiVirus 2006 v.7.1	2	99.78%	358	78.31%	14043	14.70%	871	54.70%
McAfee VirusScan Enterprise 8.0	0	100.00%	0	100.00%	46	99.01%	0	100.00%
MicroWorld eScan Internet Security for Windows 8.0.673.1	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Norman Virus Control v.5.82	0	100.00%	0	100.00%	309	92.76%	12	99.45%
NWI Virus Chaser v.5.0a	0	100.00%	4	99.90%	14	98.06%	12	99.14%
Sophos Anti-Virus v.6.03	0	100.00%	8	99.80%	1	99.86%	14	99.33%
Symantec AntiVirus 10.0.0.359	0	100.00%	0	100.00%	4	99.91%	0	100.00%
Trend Micro OfficeScan Corporate Edition v.7.3	0	100.00%	13	99.68%	851	94.42%	30	98.76%
TrustPort Antivirus 2.01.855	0	100.00%	0	100.00%	25	98.88%	0	100.00%
VirusBuster VirusBuster 2006 for Windows Servers v.5.2	0	100.00%	0	100.00%	128	93.92%	25	99.12%

Having heard much about the Finnish company, I was eager to try out its product, and was not disappointed by the experience.

The installation splash screen contrasted a funky blaze of colour in one corner with an expanse of chilly white, after which the product set itself up rapidly without need for a reboot (although I was warned after applying the update that it might need a few minutes to settle in).

It strode comfortably through the on-demand tests, presenting me with a usable HTML log, but indulged in some odd blocking behaviour on access, forcing me to resort once more to deletion. This went just as well as the on-demand scan, and with the only samples missed being in file types not scanned by default, *F-Secure*'s excellent performance amply justifies a VB 100% award.

Fortinet FortiClient 3.0.001

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	84.47%

FortiClient added yet another new product to my rapidly broadening experience – one which left more good impressions.

Stylish good looks, ease of use and a comprehensive range of functions, all controlled from a central interface, were added to decent speeds and solid detection rates, although many of the new polymorphic samples were missed. *FortiClient* also earns a VB 100% award.

Oct 2006 100% VIRUS BULLETIN www.virusbtn.com

FRISK F-Prot v.3.16f

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	99.85%	**Macro (o/a)**	100.00%
Standard	99.69%	**Polymorphic**	99.93%

F-Prot provided another of the more techie-looking GUI experiences, oozing reliability and solidity. As *FRISK* provided the engine for the false-positiving *Authentium*, I feared this product may suffer the same problem, but fortunately the alert system described the problem file merely as a 'suspicious file' – which is permissible under the rules of the VB 100% award – before recording the same infection message displayed by *Authentium*.

However, in a bizarre twist, a sample of W32/Aimbot was consistently ignored on-access, despite equally consistent detection on demand, so *F-Prot* misses out on the award this time round.

GDATA AntiVirusKit 16.0.7

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

GDATA's installation featured a rather scary swirly cog on its splash screen, and set itself up with two separate desktop shortcuts, both featuring its red-and-white logo. After a reboot, the product – which combines *BitDefender* and *Kaspersky* detection technology with its own user experience – presented a handy desktop gizmo featuring a clock, a news ticker, virus alerts, a virus info lookup system, and a set of handy links, with *Virus Bulletin* placed second behind *GDATA* itself.

The scanner GUI itself was reasonably user-friendly, although the 'protocol only' option in the actions list confused me somewhat, and the logging was a little over complicated and slow to display. Despite excellent detection throughout the infected test sets, results were marred by what eagle-eyed readers will be expecting – a false alarm in the clean set from the *BitDefender* engine, which was enough to deny the product the VB 100% award.

Greatsoft Virusclean v.2.0.3286.3

ItW	99.85%	**Macro**	100.00%
ItW (o/a)	99.85%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

Receiving offers of new products for the comparative review was an exciting experience – I responded to preliminary enquiries from developers with a mix of hope and worry. *Greatsoft*'s web presence revels in the URL viruschina.com, which was reassuringly clear and slick. The installation process, although in need of a little proof reading, was equally smooth, and the GUI offered several useful tools, including a system for backing up and restoring boot records.

Using the product was a less happy experience, however. My first worry came when I found the 'Select Folders' window of the scanner only had options for the floppy and network drives; this was mitigated by a handy toolbar where folders could be typed in manually for scanning.

With speed tests and on-demand scans completed in this manner, I came to the on-access tests, only to find little information about the on-access scanner. Fearing my discussions with the developers had been less than clear, I thought at first this must be an on-demand only scanner. Eventually, however, I discovered that the on-access component, the 'monitor', was enabled for some routes of ingress to the machine but not locally – options for 'file' and 'big file' monitoring needed to be enabled to make this happen. The system did not seem to be in place by default, and indeed was only active when the scanner GUI was, but also seemed to require a reboot to activate configuration changes.

After several false starts and confusing results however, an accurate set of statistics was obtained, with impressive detection in the zoo sets, but a sample of W32/Eyeveg missed in the ItW test set and a rash of false positives spoiled *Greatsoft*'s chance of a VB 100% award first time out of the blocks.

On-demand tests	ItW File		Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%
AhnLab V3Net for Windows Servers 6.0	0	100.00%	47	98.97%	626	90.48%	57	97.13%
Alwil avast! v.4.7	0	100.00%	18	99.56%	385	89.90%	30	98.74%
Avira AntiVir Windows Server 2003/2000/NT v. 6.35	0	100.00%	3	99.93%	0	96.37	150	100.00%
BitDefender Antivirus v.10	0	100.00%	13	96.69%	37	97.02%	10	99.27%
CA eTrust 8.0.403.0 (InoculateIT)	0	100.00%	4	99.90%	42	97.23%	1	99.82%
CA eTrust 8.0.403.0 (Vet)	0	100.00%	12	99.82%	103	94.26%	1	99.96%
CAT Quick Heal 2006 v.8.0	0	100.00%	73	98.23%	602	87.07%	98	96.51%
Command Authentium AntiVirus for Windows 4.93.8	0	100.00%	0	100.00%	2	99.93%	1	99.98%
Doctor Web Dr.Web v.4.33.2	0	100.00%	0	100.00%	9	98.08%	0	100.00%
Eset NOD32 2.5	0	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus for Windows Servers v.5.52	0	100.00%	0	100.00%	0	100.00%	3	99.85%
Fortinet FortiClient 3.0.001	0	100.00%	0	100.00%	277	84.47%	0	100.00%
FRISK F-Prot v.3.16f	0	100.00%	0	100.00%	2	99.93%	3	99.69%
GDATA AntiVirusKit 16.0.7	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Greatsoft Virusclean v.2.0.3286.3	1	99.85%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG Anti-Virus 7.1	0	100.00%	3	99.93%	414	82.59%	27	98.56%
Kaspersky Anti-Virus 5.0 for Windows File Servers v. 5.0.77.0	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Kingsoft AntiVirus 2006 v.7.1	2	99.78%	358	78.31%	14043	14.70%	871	54.70%
McAfee VirusScan Enterprise 8.0	0	100.00%	0	100.00%	46	99.01%	0	100.00%
MicroWorld eScan Internet Security for Windows 8.0.673.1	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Norman Virus Control v.5.82	0	100.00%	0	100.00%	309	92.76%	4	99.71%
NWI Virus Chaser v.5.0a	0	100.00%	4	99.90%	14	98.06%	13	98.96%
Sophos Anti-Virus v.6.03	0	100.00%	8	99.80%	1	99.86%	15	99.30%
Symantec AntiVirus 10.0.0.359	0	100.00%	0	100.00%	4	99.91%	0	100.00%
Trend Micro OfficeScan Corporate Edition v.7.3	0	100.00%	13	99.68%	851	94.42%	30	98.76%
TrustPort Antivirus 2.01.855	0	100.00%	0	100.00%	25	98.88%	0	100.00%
VirusBuster VirusBuster 2006 for Windows Servers v.5.2	0	100.00%	0	100.00%	628	92.00%	27	99.27%

Grisoft AVG Anti-Virus 7.1

ItW	100.00%	**Macro**	99.93%
ItW (o/a)	100.00%	**Macro (o/a)**	99.93%
Standard	98.56%	**Polymorphic**	82.59%

Installation of *AVG* was slowed down not only by the marathon licence code (totalling 31 characters, plus seven hyphens), but also by the absence of a necessary DLL in the default *Windows 2000* setup – MSVCP60.DLL, also required by many variants of W32/Mytob. With these hurdles overcome, and a restart suggested but not initiated by the product, I was offered a tall, skinny GUI, with the option to switch to a more friendly ‘Basic Interface’. Both of these were fairly straightforward to operate, and on-demand scanning surprised me only by the numbers of ‘could be’ lines in the log.

With good speeds and solid detection, only let down seriously by several misses in the polymorphic set, along with a miraculous lack of false positives, *Grisoft* earns itself a VB 100%.

Kaspersky Anti-Virus 5.0 for Windows File Servers v.5.0.77.0

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

Kaspersky’s product came as a basic command-line operated system, with a GUI available for those who require it. With time pressing and many more products to come, I opted to skip this extra step, and ran through the tests using the simple and well documented command-line controls. After an initial test during which the product seemed consistently to ignore a single Mytob sample in the Wild set, a reinstall on a fresh machine soon smoothed out this odd quirk, and I was not surprised (given *GDATA*’s performance), to find another product capable of taking the entire test set in its stride. Only two files were missed across all collections, both zips in a zoo set not scanned by default on-access, and with no false positives *Kaspersky* racks up another VB 100% award.

Kingsoft AntiVirus 2006 v.7.1

ItW	99.78%	**Macro**	78.31%
ItW (o/a)	99.78%	**Macro (o/a)**	78.31%
Standard	54.70%	**Polymorphic**	14.70%

The second of the VB 100% first-timers arriving this month from China, although the first to hit the test bench, was provided by *Kingsoft* – a company whose primary output is computer games and office software. The product offered a fairly standard experience however, with a straightforward installation process remarkable only for a few odd uses of language.

The GUI, once up, was simple to operate, and on-demand scans were admirably rapid. Once completed, the set of infections detected was presented, along with the option to ‘clean’ them. Once this was rejected, and after some processing, the same list returned, this time with a ‘quarantine’ option, and then a third time with the offer to delete. With all these rejected, a log was provided which when parsed revealed very large numbers of misses across the zoo test sets.

The WildList, however, was handled much more impressively, with only two samples missed: a W32/Mytob and a Kakworm in .HTA format. These misses, along with no fewer than five false positives in the clean set, denied *Kingsoft* the VB 100% this time, but leaves the product looking a good contender for qualification in the near future.

McAfee VirusScan Enterprise v.8.0.0

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	99.01%

McAfee’s product installed cleanly, and once done informed me that some components would require a reboot to be fully operational. These did not, it seems, include the on-access virus scanner, which appeared operational from the off.

The main GUI was simple and pared-down, but opened numerous other windows during the process of configuring and running a scan.

Speeds were impressive, although the on-access scanner was noticeably slow, and only one of the new polymorphic set prevented *McAfee* from taking a clean sweep of the infected sets. With no false positives either, *McAfee* joins the other high achievers on this month’s VB 100% platform.

MicroWorld eScan Internet Security for Windows 8.0.673.1

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

Another product using the *Kaspersky* engine, *MicroWorld eScan* provided its own interface and also added in a little slowness over the scans of infected areas, although it achieved decent throughput over the clean sets.

On first attempt, a single file was missed on access, but I could not get this bad behaviour to repeat itself, and another VB 100% award is the result.

Norman Virus Control v.5.82

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	99.71%	**Polymorphic**	92.76%

Norman's installation was fast and simple, with no reboot required, but the GUI seemed over complex, with numerous windows used in the process of configuring and running a scan 'task'.

Throughput in the speed tests was somewhat slow in some areas and remarkably fast in others, while detection in the infected sets was mostly very good, missing a handful of standard viruses and a few sets of polymorphic samples. The WildList and clean sets were dealt with without a flaw, earning *Norman* a VB 100% award.

NWI VirusChaser 5.0a

ItW	100.00%	**Macro**	99.90%
ItW (o/a)	100.00%	**Macro (o/a)**	99.90%
Standard	98.96%	**Polymorphic**	98.06%

VirusChaser offers a rebadged invocation of the *Dr.Web* scanning engine, and much attention has been paid to the rebadging. After a fast and easy installation, with language options leaning towards the Asian market, there were options to tweak the GUI into any of a variety of pastelly shades for my visual pleasure.

Graphics were also configurable, and a choice of system tray icons for the on-access scanner was prominent, with *VirusChaser*'s own available as an alternative to the SpIDer. A disk usage monitor was one of a few innovative ideas added to the interface.

Scanning was decent, once the logs were discovered, although on-access seemed to offer little configuration and some unpredictable behaviour, and the product fared slightly less well than the engine it is built upon has proved itself capable of. Despite this, few infections were missed, with the entire ItW set detected, without false positives, and *VirusChaser* earns itself a VB 100% award.

Sophos Anti-Virus v.6.03

ItW	100.00%	**Macro**	99.80%
ItW (o/a)	100.00%	**Macro (o/a)**	99.88%
Standard	99.30%	**Polymorphic**	99.86%

The AV component of *Sophos*'s recently-released enterprise suite is not visibly very different from the previous version, apart from offering to install a firewall during the browser-style installation process.

The GUI, which feels a little lopsided and lacking in symmetry, was easy to use and scans were initiated without difficulty. The progress bar provided was a little misleading, hinting that a scan was 80% complete when the figures showed that less than half the files had been processed, and a change in the logging method meant that many files were labelled as part of an infection rather than merely an infection in themselves.

Despite these minor issues, with speeds good and only a single sample from a large set of new polymorphic types added to its usual low rate of misses, *Sophos* easily earns another VB 100%.

Symantec AntiVirus 10.0.0.359

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	99.91%

Symantec required me once again to update the browser on my test machine, the minimum it supports being *IE 5.5 SP2*. With *IE* upgraded, the installation was speedy and efficient, with no rebooting and an automated scan of important areas.

The browser seemed necessary only for viewing reports, which showed a file in the clean set flagged as a 'security risk' during the speed tests, which were a little on the slow side. During scanning of the infected sets, this slowness increased dramatically; presumably encountering an infection triggers some super-in-depth analysis of the file in question, as the scan dragged on for a spectacular 4,700 minutes. This may have had something to do with on-access reactivating itself without my noticing.

Once logs for the four days were gathered, rejoined and parsed, a tiny handful of polymorphic viruses were the only misses, and a VB 100% was earned without difficulty.

Trend Micro OfficeScan Corporate Edition 7.3

ItW	100.00%	**Macro**	99.68%
ItW (o/a)	100.00%	**Macro (o/a)**	99.68%
Standard	98.76%	**Polymorphic**	94.42%

Trend's installation process was by far the most complex of all the products, with numerous dialogs offering and requesting information on a huge array of components and functions. This product also required a browser upgrade, this time *IE 5.5 SP1* being the minimum.

The client side was adequate for many tests, its big fat buttons and chunky checkmarks making setting things up fairly foolproof, but the 'options' button was greyed out and the server console was needed for more advanced configuration.

Having zipped through the speed tests, the machine got a little bogged down towards the end of a hefty scan of infected collections, but soon recovered. Several alerts were issued for items found in the quarantine folder, rather confusingly, and detection in the polymorphic set was a little disappointing, but in the end the WildList viruses were all found and the clean set produced no surprises, resulting in a VB 100% award for *Trend Micro*.

Trustport AntiVirus 2.01.855

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	98.88%

Trustport is another product combining two engines from separate providers, along with some useful functionality of its own, and controls them from a useable GUI, marred only by the occasional bit of odd English and some strange logging behaviour – including reporting times for scans seemingly unrelated to the system time.

The combination of *BitDefender* and *Norman* engines worked well for *Trustport*, giving better detection rates across the zoo sets than either provider on its own, but of course it also suffered the same false positive as *BitDefender*, rendering its flawless detection of ItW viruses inadequate to earn it the VB 100%.

VirusBuster VirusBuster 2006 for Windows Servers v.5.2

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	99.27%	**Polymorphic**	92.00%

After a straightforward installation process, *VirusBuster* offers a selection of GUIs, including a Microsoft Management Console (MMC) based configuration system, opened from the desktop shortcut provided, and a more user-friendly scanner control, somewhat confusingly entitled the 'console' and opened from the system tray menu.

After a slightly complicated setup process, scanning speeds were decent, although a file in the clean set snagged the product rather nastily and another was reported 'suspicious'. These issues aside, detection rates were very good, and another VB 100% award is due to *VirusBuster*.

CONCLUSIONS

With such a huge raft of entries to test, time to analyse individual products in detail was a little short, but a few broad patterns seemed to emerge. There appeared to be a fairly distinct divide between the products that thought they knew best, and provided little chance to conform their behaviour to suit an individual's requirements, and those that seemed aimed more firmly at the expert or corporate user, and thus provided a wealth of detailed levels of configurability. On either side of this divide detection rates were generally strong, although the small handful of new samples introduced managed to sneak something past most of the entries.

Most noticeable was the large number of false positives, an effect not helped by many other products running one or other of the engines affected by them. All of these were in the older part of the clean set, and so should have been inspected many times before by most of these products. The exceptions to this, the two new entries, unsurprisingly suffered most heavily from false positives, but also missed out where it matters most, in the WildList. Hopefully all these issues will soon be resolved by the respective vendors. A select few can, of course, walk away with their heads held high.

Technical details

Test environment: Identical 1.6 GHz *Intel Pentium* machines with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive, running *Windows 2000 Server*, service pack 4.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/Win2K/2006/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

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

END NOTES & NEWS

The 16th Virus Bulletin International Conference, VB2006, will take place 11–13 October 2006 in Montréal, Canada. Email vb2006@virusbtn.com for details of sponsorship opportunities. Register online at http://www.virusbtn.com/.

RSA Conference Europe 2006 takes place 23–25 October 2006 in Nice, France. Online registration and full details of the conference agenda are available now at http://2006.rsaconference.com/europe/.

Infosecurity USA will be held 24–25 October 2006 in New York, NY, USA. See http://www.infosecurityevent.com/.

InfoSec World 2006 - Lapland takes place 21–24 November 2006 in Rovaniemi, Lapland Finland. The conference will build on MIS Training's successful series of flagship information security conferences such as: CISO Executive Summit, WebSec, FinSec and InfoSec World USA. For details see http://www.mistieurope.com/.

SecureGOV 2006 takes place 3–5 December 2006 in Farmington, PA, USA. The fourth annual SecureGOV strategic intelligence meeting offers senior government IT, security and privacy officers insight into the latest developments critical to maximizing the protection of information resources, wireless communications, networks and critical infrastructure. See http://www.convurge.com/index.php?section=67.

AVAR 2006 will be held 4–5 December 2006 in Auckland, New Zealand. For full details and online registration see http://www.aavar.org/.

The 2nd AVIEN Virutal Conference will take place on Wednesday 10 January 2007, from 15:00 to 17:00 GMT (starting at 8am PST, 11am EST). This year's conference topic is 'The new face of malware: stories from the battlefield'. Sign-up details will be announced in due course.

RSA Conference 2007 takes place 5–9 February 2007 in San Francisco, CA, USA. The theme for this year's conference – the influence of 15th century Renaissance man Leon Battista Alberti, the creator of the polyalphabetic cipher – will be covered in 19 conference tracks including: 'applied security case studies', 'authentication', 'consumer protection', 'cryptographers', 'deployment strategies', 'developing with security', 'enterprise defence', 'identity & access management', 'law & liability', 'policy & government', 'security solutions', 'standards', and 'wireless'. For full details see http://www.rsaconference.com/2007/US/.

Black Hat Federal Briefings & Training 2007 take place 26 February 26 to 1 March 1 2007 in Arlington, VA, USA. Registration for the event will close on 18 February 2007. For details see http://www.blackhat.com/.

Websec 2007 will take place 26–30 March 2007 in London, UK. More information will be available in due course at http://www.mistieurope.com/.

The 16th annual EICAR conference will be held 5–8 May 2007 in Budapest, Hungary. A call for papers for the conference has been issued with a deadline of 12 January 2007 for peer reviewed papers and 1 December 2006 for non-reviewed papers. Full details can be found at http://conference.eicar.org/2007/index.htm.

The 22nd IFIP TC-11 International Information Security Conference takes place 14–16 May 2007 in Sandton, South Africa. Papers offering research contributions focusing on security, privacy and trust are solicited. For more details see http://www.sbs.co.za/ifipsec2007/.

The International Conference on Human Aspects of Information Security & Assurance will be held 10–12 July 2007 in Plymouth, UK. The conference will focus on information security issues that relate to people – the methods that inform and guide users' understanding of security and the technologies that can benefit and support them in achieving protection. For more details, including a call for papers, see http://www.haisa.org/.

ADVISORY BOARD

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

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

Corporate rates include a licence for intranet publication.

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

Editorial enquiries, subscription enquiries, orders and payments:
Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England
Tel: +44 (0)1235 555139 Fax: +44 (0)1235 531889
Email: editorial@virusbtn.com Web: http://www.virusbtn.com/

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

vb Spam supplement

CONTENTS

NEWS & EVENTS

PHISHERS INDICTED

Six US men have been indicted on charges of masterminding a phishing operation. According to the indictment, the six men used email-harvesting software to obtain *AOL* members' details from the Internet. Then, between 2004 and earlier this year, the group bombarded the *AOL* members with messages that used a variety of phishing techniques.

Those customers unfortunate enough to have fallen into the trap of the phishing scams had their financial details stolen and used to purchase merchandise on the Internet including gaming consoles, laptops and gift cards. Later, the phishing gang changed their tactics and began using the stolen information to produce counterfeit debit cards.

If convicted, the defendants face up to 15 years in prison for fraud in connection with access devices and aiding and abetting fraud in connection with access devices; up to 7.5 years in prison for charges of conspiracy to commit fraud in connection with access devices; and up to five years in prison for fraud in connection with electronic mail.

Meanwhile, *Microsoft* is celebrating a successful civil case brought against British spammer Paul Fox. *Microsoft*'s decision to pursue a civil action against the spammer paid dividends, since Fox has been ordered to pay the company £45,000, whereas the UK's anti-spam laws allow for a maximum fine of just £5,000.

EVENTS

The Text Retrieval Conference (TREC) 2006 will be held 14–17 November 2006 at NIST in Gaithersburg, MD, USA. For more details see http://plg.uwaterloo.ca/~gvcormac/spam/.

Inbox 2007 will be held 31 May to 1 June 2007 in San Jose, CA, USA. For more details see http://www.inboxevent.com/.

FEATURE

AISK – A DIFFERENT APPROACH

Mariusz Kozlowski
Independent researcher, Poland

One might wonder if there is anything more we can do to fight spam. And even if the answer is yes, one might then ask: 'Is there any point in starting a new anti-spam filter project?'.

During the years I spent as an administrator I had to deal with reams of spam and different filters in different configurations depending on the owner of a given server. I followed the spamassassin, bogofilter and other mailing lists and newsletters to keep up to date with things. What struck me straight away was how many people spend so much of their time (and other resources) on maintaining the filters and trying to keep up with new spam trends.

The body of the message is the place where the 'war' goes on and it is common to have header checks, regular expressions, different parsers, HTML engines, or tools such as FuzzyOCR employed. There are plenty of other weapons one can use to defeat spam – whitelists, blacklists, ACLs, DNSBLs, SURBLs, greylisting, transaction delays, etc.

But what if in couple of years we start to see spam containing short movies? I really don't know, but we may need take a look at the spam problem from a different perspective.

HUMAN ACCURACY

I'm quite sure that pretty much anyone who knows their own mail box traffic well enough is able to filter their mail box content with very high accuracy. But what one may not realize is that after a sufficient period of training of your brain, you don't need to look at the body of the message to tell if a message is spam or not.

So, why not replace the human brain connected to the body that clicks mouse or presses the delete button with an artificial brain implemented on the computer? That would be an artificial neural network (ANN) – for which the human brain is the biological inspiration. Besides, computers don't get tired and can complete some tasks much faster than humans do.

This is how a small piece of code called AISK (Artificial Intelligence Spam Killer) was conceived in the late summer of 2005.

AISK

I left the body of the message alone and focused instead on the headers. Some way of presenting the email header to the ANN was needed. The most intuitive way of doing this is to split the header into words, where each word is a so-called 'token'. The ANN needs a preclassified training set from which it can learn. But, in order to learn from the entire set of messages, it is necessary to have uniform representation for every message in the set. This is called a vector of message attributes.

To get such a vector we need to parse the training set and split it into tokens. Then the best tokens – that is the tokens that best separate spam from ham – are used to form a vector of attributes. There is a great article by Gary Robinson [1] on tokens and statistics. The 'degree of belief' Gary introduces in his article (called f(t)) is used to determine which tokens might be useful. The 'degree of belief' for each token may vary from 0 (ham tokens) to 1 (spam tokens).

The best tokens are those whose f(t) is close to 0 or 1. To define what 'close' actually means we need to define two thresholds, T1 and T2. The tokens that we are interested in are those with f(t) in the range of 0 to T1, and T2 to 1.

Research has shown that, to produce the best results, the distribution of f(t) for a typical training set would require T1 and T2 to be 0.05 ~ 0.15 and 0.85 ~ 0.95, respectively. The values of these thresholds have great impact on the filter's accuracy. The length of the vector of attributes depends directly on these values. Having such a vector of attributes matched against a message from the training set results in the so-called 'image' of the message.

Research has shown that the most efficient ANN structure for this kind of classifier is a multilayered, fully connected perceptron. Spam classification is believed to be a non-linear problem, so a hidden layer is needed. However, adding more layers slowed the system down, leaving the filter accuracy at the same level.

The spam classification problem is also often presented as binary, which means there are only two classes: 'ham' and 'spam'. Therefore we need only one network output such that a value of 0 would represent ham and a value of 1 would represent spam. Here, supervised learning requires both an input vector (the image) and a desired output value. These two together create a single training instance for the ANN.

As one might expect, the size of the input layer of the ANN depends on the length of the attributes vector found for the given training set. When the attributes vector, together with the desired output, are matched against all messages in the training set, the result is a set of training instances which is actually a matrix of data used directly to train the ANN. The attributes are binary, so either the message has a certain attribute or it does not.

Now let's think again. The first thing is to realize that the mail header has a structure. Spammers respect this structure at least to a minimal level so that mail clients can display an email correctly. A word found in the subject area of a message might mean something different from the same word found in some other part of header. For now, AISK can recognize four types of token found in mail headers: tokens found in the 'From:', 'To:' and 'Subject:' fields, and elsewhere in the header.

AI, HEADERS, TOKENS – DOES IT WORK AT ALL?

To answer this question let's look at some numbers. A test was done with the TREC 2005 public corpus. The training set consisted of 6,269 randomly chosen ham messages and 8,431 spam messages. As stated previously, only four types of token were used. Data from other sources are simply not available in the TREC corpus. Threshold T1 was set to 0.05 and T2 to 0.95. The length of the vector of attributes was 5401. The following are the results obtained against a test set consisting of randomly chosen messages (not present in the training set) from the TREC corpus:

Total of 4,490 messages (2,584 spam and 1,906 ham)

False positives:	21 (0.81%)
False negatives:	77 (4.04%)
Spam recall:	99.19%
Spam precision:	97.08%
Acc:	0.978174
Err:	0.021826

This is quite good, but we can do better. This leads us to the next topic – data sources.

DATA SOURCES

The second thing to realize is that, for most filters, words found in the message are tokens. Tokens are words. There is a good quote that says 'A filter is only as good as its training set'. So why not apply token logic to something more than just words? There are some data sources that are more reliable and much harder to forge than the message content which, as you realize, is generated by client (spammer) software.

In fact, a token can be any piece of data carrying information connected in some way to the message that will help us classify it, and for which we can compute the ‘degree of belief’. Being an administrator and having a close look directly at the SMTP traffic flow I can think of at least a couple of important data sources. For example, the day of the week and the hour of the day the message arrived. Many companies keep office hours from 8am to 4pm and the traffic at night and over weekends is mostly spam.

The SMTP session chat prior to the DATA command is another data source. This may vary a lot depending on the peer software that talks to us and sometimes it is possible to distinguish real MTA from worms simply by looking at the SMTP session. The presence of RSET right after HELO/EHLO and before MAIL FROM is also a very good spam indicator.

IP, PORT and DNS PTR records provide a lot of useful information. IP might help us to catch a permanent spammer, ports below 1024 (privileged range) might indicate a worm, and PTR might tell us the country, the ISP and the line type (ADSL/dialup) of the SMTP peer. But probably the most interesting part is the ‘operating system fingerprint’. Using a technique called passive OS fingerprinting, tools such as p0f [2] can tell us by analysing the TCP/IP flow what the remote operating system is. Moreover, it can provide us with OS genre, details such as the kernel version used or service pack installed, the distance between us and the remote peer, the link type, real OS detection, uptime of the remote machine, some TCP/IP flags or even masquerade detection.

All this is very useful. Research has shown [3] that most spam comes from hacked *Windows* boxes connected to miscellaneous kinds of botnets. They are often connected to some ADSL or dialup lines from well known ISPs and have short uptime. Meanwhile, clients detected as *BSD or *Linux* boxes were found to be reliable ham sources.

PLAYING WITH AISK

The following are some tokens extracted from the sample training set:

```
Head  0.982759     charset=”Windows-1252"
RSET  0.979167     RSET
GENR  0.977856     Windows
DETA  0.977856     2000
DETA  0.977338     XP
DETA  0.972222     SP3
PTR   0.968750     adsl
PTR   0.967426     tpnet
Head  0.961457     7bit
IP    0.958333     83.19.255.242
PTR   0.958333     dvr242
Head  0.954056     V6.00.2800.1106
LINK  0.944444     DSL
```

The first column represents the origin of the token (Head -> mail headers, RSET -> SMTP session reset prior to DATA command, GENR -> OS genre, DETA -> OS details, PTR -> DNS PTR record, LINK -> link type). The second column is the ‘degree of belief’. The third column is the textual representation of the data described by the token. The ‘degree of belief’ is close to 1, which means that these tokens are very good spam indicators. As can you see, the common denominators are the operating system, RSET and some words such as ‘V6.00.2800.1106’ found in email headers.

This is just an example of what one might find when playing with AISK. After a year of work I am nearing the conclusion that something like a ‘spam fingerprint’ might actually exist. This is much like an OS fingerprint, but a little more fuzzy. It can fingerprint some common spam sources, software used, exploitable boxes, whole botnets, etc. Spammers’ software is often rather ‘stupid’ and leaks a lot of useful information either in the headers or SMTP session or somewhere where (almost) nobody expects to find anything. An experienced administrator provided with such information after some analysis can say with reasonable accuracy whether the SMTP session carries a spam message or not.

DOING JUST FINE

Human resources are limited though, and we are simply unable to parse all the data sources mentioned earlier, find interesting factors and combine them to produce an accurate conclusion in ‘real time’ – not to mention that the patterns we are able to see are just the tip of the iceberg. But hopefully that’s where the ANN is doing just fine. Quoting the wikipedia:

> ‘In more practical terms, neural networks are non-linear statistical data modelling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. There is no precise agreed definition among researchers as to what a neural network is, but most would agree that it involves a network of simple processing elements (neurons) which can exhibit complex global behaviour, determined by the connections between the processing elements and element parameters.’

This might sound somewhat crazy, but let’s look at the numbers. The test was done with the data from my personal server using an earlier version of AISK that used all the data sources described above but an OS fingerprint wasn't supported yet. A total of 2,820 messages were split into 2/3 of the training set and 1/3 of the testing set. 16.8% was spam. The attributes vector length was 3009 and the T1 and T2 thresholds were set to 0.10 and 0.90 respectively. The training set was of a very good quality – which means that I knew exactly what was spam and what wasn't.

The following are the results on the testing set:

Total of 904 (136 spam and 804 ham)

False positives:	1 (0.12%)
False negatives:	1 (0.74%)
Spam recall:	99.26%
Spam precision:	99.26%
Acc:	0.997872
Err:	0.002128

That's good enough for everyday use I believe. Since then, much more research has been done with small and large training sets from different servers with different traffic characteristics. Accuracy on a good training set can quite easily reach close to or more than 99%. But, as one might expect, a perfect anti-spam filter does not exist. It is simply impossible to reach the magic 100% accuracy level due to a number of different factors. The passive OS fingerprinting helps a lot, though, in places where other data sources are unreliable or of no use for some reason.

Resource consumption is at an acceptable level for server use. Finding interesting tokens and training the ANN takes some time (typically a couple of minutes) depending on the machine we have and the size of the training set, but the length of time AISK needs to handle a single incoming message (which is 20 ~ 30 ms) is a reward here. For now, AISK is a fairly simple filter which was implemented with an idea of acting like an experienced administrator.

PROBLEMS TO SOLVE

There are still many problems to solve though. Some of the best tokens selected using the 'degree of belief' are probably highly correlated with each other, which leads to an unnecessary increase in ANN size and thus slows the system down. This is seen after ANN training when some inputs are left with a weight equal to zero, which basically means that they carry no new information.

Using LDA or PCA would probably help here. In fact, many pattern recognition techniques may be applied here. FANN library implementing ANN can be optimised for AISK use, which will result in system speedup and reduced memory consumption.

Threshold T1 and T2 levels have a great impact on system accuracy. For now they are set manually to some (probably not often optimal) levels, but choosing the right levels in some automated way would result in system speedup, better accuracy and limited consumption of resources. The more users per ANN the worse results are seen, as when the users prepare the training sets themselves the definition of spam and ham differs from user to user, resulting in conflicting instances in the training set. Using one ANN per user or domain would solve the problem and result in increased accuracy.

One might expect AISK to fail to work correctly when it receives relayed mail traffic, as some of its data sources are of no use. Research has proved that even with mixed traffic where one box receives both relayed and regular mails, AISK behaves very well. AISK simply adapts to this situation and some of its data sources will be less important than the others. (It behaves best without relayed traffic though.)

The last thing is that the spam classification problem is believed to be binary – a message is either spam or ham. I believe this to be wrong. Some uncertainty class needs to be introduced, which would be perfectly natural here. The world is not black and white and when the filter says that the message is 'grey' the filter might be right. It might just indicate that its knowledge is insufficient. What we should do is to provide the filter with the required knowledge and try to make this class as small as possible.

CONCLUSION

I believe the answer to both questions at the beginning of this article is 'yes'. Working on AISK was, and is, fun and has given me the freedom of choice of what my anti-spam filter should look like.

If you would like to use or experiment with AISK you are free to do so. It is available under the GPL licence. You will find more information on the project homepage http://aisk.sourceforge.net. If you have any comments or questions please email me at m.kozlowski@tuxland.pl.

REFERENCES

[1] Robinson, G. A statistical approach to the spam problem. http://www.linuxjournal.com/article/6467.

[2] p0f: http://lcamtuf.coredump.cx/p0f.shtml.

[3] http://aisk.tuxland.pl/os-fp-vs-spam-src.html.