JUNE 2006

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

IN THIS ISSUE

YET ANOTHER FIRST …

In April 2006 a virus appeared for a new virusable platform – the general-purpose, mathematics-oriented *MatLab*. Vesselin Bontchev provides us with the full details of the unremarkable and slightly buggy proof-of-concept virus MLab/Balogy.A.

page 4

… AND A LAST

In Matt Ham's final comparative review for *Virus Bulletin* he puts 26 products for *Windows XP* through their paces. Two products enter the test line-up for the first time this month: *TrustPort Antivirus* and the rather more well-known *Microsoft OneCare*. In his own inimitable style, Matt provides rude comments and/or praise for all products, as well as the all-important VB 100% results.

page 11

June 2006 100% VIRUS BULLETIN www.virusbtn.com

vbSpam supplement

This month: anti-spam news and events; and John Graham-Cumming describes the aims of and ideas behind the *SpamOrHam* project.

ISSN 1749-7027

‘Phishing levels declined during the first part of 2006, but they are expected to rise again due to the adoption of spear-phishing techniques.’

Mark Sunner
MessageLabs, UK

FROM THE BEDROOM TO THE BANK – IT THREATS EVOLVE

In the past, the perception of the ‘typical’ virus writer or hacker was that of a male teenager, beavering away at a PC in his bedroom, intent on gaining notoriety for his exploits.

Now, a British teenager who fits that description very closely is facing extradition to the US to stand trial for what has been described as ‘the biggest military hack of all time’. If found guilty, Gary McKinnon could face decades in jail as well as massive fines.

We often hear about cases where individuals from the online world have received severe offline punishments for their crimes. Yet, despite the fact that the stakes are high, hackers and virus writers are increasingly lured into criminal acts by the prospect of monetary gain, and many are honing their skills accordingly.

Spyware provides a lucrative revenue stream for the growing number of criminals who have control over robot networks (or botnets). In a survey conducted by the US National Cyber Security Alliance (NCSA) and *AOL* in December 2005, it was found that 61% of the computers in the survey had some type of spyware or adware installed on them, less than 10% of which was with the owner’s knowledge or permission (see http://www.staysafeonline.info/pdf/safety_study_2005.pdf).

A criminal can make several thousand dollars by installing adware remotely on the compromised PCs under their control, without the owner’s knowledge. While each individual installation may generate only a few pennies of revenue, the overall gain can be significant for someone who has control of a large botnet.

Of more concern are the malicious worms that are used to create the botnets. These gather very sensitive information from users’ machines, including cracked usernames, passwords, credit card numbers and other personal data stored inside web browser auto-fill databases. With this level of intelligence, fraudsters can target their attacks very effectively.

Indeed, the bad guys are becoming increasingly tactical and their attacks more targeted. For example, the days of the scatter-gun approach to phishing seem to be numbered, having been replaced by ‘spear-phishing’.

By improving the structure and content of the phishing emails, reducing the size of each attack and targeting selective groups of addresses – such as the employees of a particular bank or organization – phishers can improve their chances of success significantly.

According to the most recent *MessageLabs* Intelligence Report (see http://www.messagelabs.com/Threat_Watch/Intelligence_Reports/), phishing levels declined during the first part of 2006 (1 phish in every 356.2 messages in Q1 06, compared to 1 in every 279.8 messages in Q4 05), but they are expected to rise again due to the adoption of spear-phishing techniques.

We have discussed just some of the threats associated with email, which has become as ubiquitous as the telephone. Although email is currently the favourite vehicle for the bad guys, other tools such as Instant Messaging, VoIP telephony and mobile devices will increase in popularity and will increasingly be targeted by criminals in the future.

With the threat landscape moving beyond email and increasing in sophistication, many companies have tightened their security, but there is still room for improvement beyond reactive security software. The reality is that traditional anti-spam and anti-virus solutions provide inadequate protection, and are circumvented easily by criminals.

Editor: Helen Martin

Technical Consultant: Matt Ham

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

BANK TAKES STEPS TO INCREASE CUSTOMER SECURITY

In an attempt to prevent online banking fraud, a British bank has signed a deal with Finnish anti-virus vendor *F-Secure* to provide anti-virus software for each of its online banking customers.

Barclays bank signed a deal last month for 1.6 million *F-Secure AntiVirus* licences as well as two years' worth of updates. The package will include anti-virus, anti-spyware, and anti-rootkit protection.

Barclays will notify each of its 1.6 million online customers that the anti-virus software is available free of charge, and will provide them with download instructions. The software will be set to update automatically. Currently, it is not clear what will happen at the end of the two years' worth of free updates – although a spokesman for *Barclays* has said that the bank may decide to include anti-virus protection as an integral feature of its online service.

ONECARE GOES LIVE

Microsoft's anti-virus software *Windows Live OneCare* is due to go on general release this month. *OneCare*, which has been available free of charge in beta form since November 2005, will be available in its final, fully supported version, for $49.95 per year. The cost will include licences for up to three *Windows XP* PCs.

Microsoft announced recently that the product had received certification from *iCSA Labs* as well as having received the *Checkmark* certification from *West Coast Labs*. This month, to complete the set, *OneCare* was submitted to *VB* for inclusion in the *Windows XP* comparative review. The results can be found on p.11.

SYMANTEC VULNERABILITY DISCOVERED AND FIXED

Symantec was quick to respond late last month to the discovery of a potentially critical vulnerability in the latest versions of its corporate anti-virus software.

The stack overflow vulnerability, which was discovered by researchers at *eEye Digital Security* in *Symantec Client Security 3.x* and *Symantec AntiVirus Corporate Edition 10.x*, would potentially allow remote attackers to execute code on the affected machine – without any user interaction.

Symantec responded to the discovery by releasing a series of intrusion prevention signatures for the affected versions of the software. The company was also quick to point out that it was not aware of any customers affected by the vulnerability, or of any exploits of the vulnerability.

Prevalence Table – April 2006

Virus	Type	Incidents	Reports
Win32/Netsky	File	70,745	43.79%
Win32/Mytob	File	42,986	26.61%
Win32/MyWife	File	14,933	9.24%
Win32/Mydoom	File	13,977	8.65%
Win32/Bagle	File	4,682	2.90%
Win32/Zafi	File	3,990	2.47%
Win32/Lovgate	File	2,599	1.61%
Win32/Bugbear	File	2,386	1.48%
Win32/Sdbot	File	1,321	0.82%
Win32/Pate	File	855	0.53%
Win32/Funlove	File	471	0.29%
Win32/Feebs	File	408	0.25%
Win32/Klez	File	269	0.17%
Win32/Reatle	File	225	0.14%
Win32/Mabutu	File	163	0.10%
Win32/Sality	File	149	0.09%
Win32/Valla	File	130	0.08%
Win32/Dumaru	File	94	0.06%
Win32/Mimail	File	92	0.06%
Win32/Bagz	File	84	0.05%
Win32/Gibe	File	81	0.05%
Win32/Maslan	File	78	0.05%
Win32/Bobax	File	57	0.04%
Win32/Areses	File	50	0.03%
Win32/Randex	File	40	0.02%
Win32/Mota	File	37	0.02%
Win32/Swen	File	37	0.02%
Win32/Kriz	File	36	0.02%
Win32/Wukill	File	34	0.02%
Redlof	Script	32	0.02%
Win32/Rontokbro	File	31	0.02%
Win32/Elkern	File	25	0.02%
Others[1]		445	0.28%
Total		161,542	100%

[1]The Prevalence Table includes a total of 445 reports across 56 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

MATH BALONEY: YET ANOTHER FIRST

Dr Vesselin Bontchev
FRISK Software International, Iceland

On 22 April 2006, Finnish anti-virus researcher Mikko Hyppönen reported that *F-Secure* had received the first virus for a new virusable platform (see http://www.f-secure.com/weblog/#00000859). The platform for which the virus is written is *MatLab*, made by *MathWorks, Inc.* (see http://www.mathworks.com/products/matlab/).

MATLAB

MatLab is a general-purpose, mathematics-oriented platform that can be used for various computations. Since mathematics is used pretty much everywhere, the applications of the product are numerous.

I happened to have easy access to the program because my mother – a professor at the Institute of Mechanics and Biomechanics in the Bulgarian Academy of Sciences – uses it for automated preprocessing of the output of her favourite CAD/CAE product. However, *MatLab* can be used for just about anything that involves computation: education, mathematics research, physics, statistics and even stock portfolio management.

The product is programmed in a proprietary language, which is vaguely C- or Pascal-like. I couldn't find an official name for the language in the product's documentation, but its users often refer to it as 'MatScript'. The programs written in this language are stored in files with the extension .M – *MatLab* calls them 'M-files'. The files are ASCII text files and can be opened with *Notepad* (although *MatLab* has a built-in editor/debugger for them).

The language is universal and powerful – not only does it have computationally oriented functions, but also a full set of file and string manipulation functions. Powerful enough to write a virus in it, that is. Which is precisely what has been done.

NAMING

As a result, the members of CARO had to come up with a name for the platform the virus infects. After some discussions (during which, regrettably, the proposal for using 'MS', as in 'MatScript', was rejected due to its similarities with the abbreviation of *Microsoft*), we eventually decided to use, respectively, 'MatLabScript' and 'MLab' for the long and the short forms of the platform name. The document describing the CARO naming scheme has been updated accordingly (see http://www.people.frisk-software.com/~bontchev/papers/naming.html).

Next, we needed a family name for the virus. Apparently its author wanted it to be named 'MatLab.Bagoly.a', as is evident from the comment lines at the beginning of the virus body:

```
%--------------------------
%MatLab .m file infector by Positron (MatLab.Bagoly.a)
%--------------------------
```

However, the members of CARO are not in the business of gratifying the egos of virus writers, so we decided to use the slightly distorted name 'Balogy' instead. (As former *Virus Bulletin* editor Nick FitzGerald pointed out, this sounds a bit like 'baloney', which pretty much reflects our opinion on the appearance of viruses for yet another virusable platform.)

So, the full CARO name of the virus is:

virus://MatLabScript/Balogy.A

REMARKABLY UNREMARKABLE

Like most proof-of-concept viruses, this virus is remarkably mediocre, full of stupidities, and has virtually no chance of spreading in the wild.

The virus can be classified as a parasitic prepender. That is, it infects other M-files by inserting its own text at the very beginning of their contents. Files that contain the string '__EndSignature__' on any line are considered already infected and are left alone. The virus has the following as its last line for self-recognition purposes:

```
e__ = '__EndSignature__';
```

The virus works using a very simple and straightforward algorithm. It starts by opening the file from which it has been executed (*MatScript* has a built-in variable that returns the name of the currently running file) and by reading its content, line by line, until the 'end signature' is found. Each line is stored in an element of a string array.

Then the virus inspects all *.M files in the current directory. The content of each file found is read (once again on a line-by-line basis) into another string array. If, during this reading, the 'end signature' is found anywhere on a line, the file is considered already infected and will not be touched any further.

Otherwise, the file is opened for writing and the virus writes into it the virus body (stored in the first string array) and then the original content of the file (stored in the second string array), after which the file is closed.

That's it – the virus only replicates, and only in the current directory. It has no payload whatsoever. Yet, despite the simplicity of the algorithm, the virus author has managed to make several logical and strategic mistakes.

HARDLY ANY GOOD

First, the virus contains two instances of the following line:

```
if tline__ == -1,  break,  end
```

The purpose of this line is to determine that the end of the file has been reached (first when reading the virus body and again when reading the original content of the file that is going to be infected). However, at least under *MatLab* version 6.1.0.450 (R12.1), this line generates the following warning once per file:

```
Warning: Future versions will return empty for empty
== scalar comparisons.
```

This means that each time the virus runs, the user will be 'rewarded' with 2(N+1) such warnings, where N is the number of M-files in the current directory. (N files, plus one file from which the virus is running, and two warnings per file because there are two instances of the line that causes the warning.) That's hardly unnoticeable.

Second, when determining whether a file is already infected, the virus continues to read it line by line until the very end – even after it has determined that the file is already infected and will have to be ignored. That's hardly wise.

Third, it is obvious that the first time the virus is run, it will infect *all* the files in the current directory. Why, then, try to do it again the next time and every time it is run? Hoping that somebody has added a new file meanwhile? That's hardly intelligent.

Fourth, the virus does not attempt to spread outside the current directory – i.e. to other directories and/or machines – despite the fact that MatScript does have the means to achieve such goals (see the next section). So, the only way in which another user can become infected with the virus is if the current directory is a shared one (e.g. on a network server), or if somebody passes them an infected M-file. That's hardly efficient.

Finally, when the virus writes to the target file, it uses '\n' as a line separator. In MatScript, this results in the lines being separated only with an LF (line feed) character. MatScript (and its built-in editor) can handle both lines that are separated only with LF characters and lines that are separated with CR/LF sequences (i.e. both Unix-style and DOS-style line endings). However, *Notepad* messes up when trying to display text files whose lines are separated only with LF characters. So, although the infected files will work in a sense, they will look messed up to the user who tries to edit them with *Notepad* or a similar unintelligent editor.

THE ANNOYANCE FACTOR

While MLab/Balogy.A is a very simplistic (and buggy) virus with virtually no chance of spreading in the real world, *MatLab* has sufficiently powerful capabilities to at least create significant annoyances for users and anti-virus researchers alike.

Indeed, *MatLab* doesn't seem to have the concept of 'autostart' script – so, a virus written in MatScript cannot hope to receive control automatically each time the product is executed. However, *MatLab* does have the concept of a 'search PATH', which means that various kinds of companion viruses are possible.

The simplest of all kinds of companion viruses are the renaming companions. A virus could store its body in a file with a name already present in the system, while renaming the original file to something else and executing it directly after the virus has finished running. Even the names of the internal *MatLab* commands can be 'overloaded' with M-files, which is almost as good as having an 'autostart' capability (e.g. by overloading the name of some often-used command like 'help' or 'edit').

Next, we can have PATH companions. By default, when the name of an M-file is typed at the command line, *MatLab* looks for a file with such a name in each of the directories on the subtree where the product is installed. However, MatScript has commands that allow the user to manipulate the search path by adding or removing arbitrary directories to/from it. Clearly, this can be exploited by a malware author.

Finally, a special kind of companionship is possible. If the files C:\Foo\bar.m and C:\Foo\private\bar.m both exist and the command 'bar' is typed from *MatLab*'s command line, *MatLab* will try to execute the second one, without the directory 'C:\Foo\Private' having to be present explicitly in the search path.

OVERWRITING

Of course, in addition to companion and parasitic viruses (both prepending and appending, although, due to some conventions about what the M-files are supposed to contain, the prependers are ‘easier’), the language also allows overwriting viruses to be written – although these are extremely noticeable (because the infected files stop working), and not very interesting.

However, it is also perfectly possible to write a LoveLetter-style mass-mailer in MatScript. *MatLab* associates the *.M extension with itself, so if the user receives a file with such an extension as an email attachment and double-clicks on it, *MatLab* will be launched and will try to execute the content of the double-clicked M-file.

When an M-file is first executed by *MatLab*, a pre-parsed form of it is kept in memory until the end of the *MatLab* session (or until purged explicitly from there with the proper command). This is done for speed reasons – later invocations of that file will result in *MatLab* running the pre-parsed memory image instead of trying to read and parse the original file again. This alone has some interesting implications in respect of self-modifying malware written in MatScript. However, it gets worse.

MatLab can save these pre-parsed memory images in files with the extension .P – and can execute them just like the M-files. In addition, if the files Foo.m and Foo.p both exist, and the command ‘foo’ is entered from *MatLab*’s command line, it will be the second file that will be executed; not the first one – which allows for yet another kind of companionship infection.

Even worse, while the M-files are ASCII text, the ‘P-files’ are binary files with – you’ve guessed it – completely undocumented format. At present we don’t even know whether their content is constant or whether they contain areas with variable content (e.g. like VBA).

CONCLUSION

MLab/Balogy.A is a relatively unremarkable and slightly buggy proof-of-concept virus for a new virusable platform. It poses no threat by itself, since it has virtually no chance of spreading in the real world. However, the capabilities of the platform are powerful enough and have the potential to cause some annoyance both to users and to anti-virus researchers.

virus://MatLabScript/Balogy.A

Aliases:	MLS/Lagob, Mlab.Lagob.
Type:	Parasitic prepender.
Infects:	*MatLab* ‘M-files’.
Payload:	None.

TECHNICAL FEATURE

INSIDE THE PE FILE FORMAT

Sanjay Katkar
Cat Computer Services, India

Microsoft’s PE (Portable Executable) file format has been in existence for quite a while. It is used in Win32-based operating systems.

In this article I will describe how recent malicious programs have exploited PE file format, manipulating the header fields to avoid detection. This technique has been in use for a couple of years – and, by now, most AV scanners should be able to detect the malware inside such header-manipulated PE files. However, there are still a number of scanners that can be fooled by this kind of trick.

Since I am assuming that most readers are familiar with the PE file format, I shall not discus the details of the PE header, section headers or the PE file structure here. More information about the headers and other details can be found in the various articles about the PE file format on *Microsoft*’s website (see, for example, http://msdn.microsoft.com/msdnmag/issues/02/02/PE/).

SEARCHING PE FILES

To search a PE file for malware a scanner will typically need both to scan the file and to perform some form of emulation for the detection of polymorphic viruses.

At some point every scanner must reach the file offset where the file execution begins. AV scanners that do not scan the whole PE file need to determine this file execution offset accurately in order to reach the virus code and scan for the signature.

In the detection of polymorphic viruses, the bytes at the file execution offset are used as a starting point for the emulation or code byte analysis process. So, for many reasons, the calculation of the file execution start offset is very important for AV scanners, and if the execution start offset is miscalculated the scanner will miss the detection. It has been observed that an increasing number of malicious programs are using tricks that make it difficult for the scanner to reach the file execution start offset. It has also been observed that certain executable packers (e.g. *NSPack*, *UPack*) build PE file headers that cause this calculation to go wrong.

CALCULATING THE PE FILE EXECUTION START OFFSET

First, we will look at how the file-based execution start offset is calculated for a typical PE file. For this, we need to understand the PE header and section header. The table below shows the important fields within the PE optional header and section header for NOTEPAD.EXE (*Windows XP Professional*). All values are in hexadecimal.

Table 1: Header information for NOTEPAD.EXE.

Optional header

Number of sections:	03	Section alignment:	00001000
Address of entry point:	0000739D	File alignment:	00000200
Image base:	01000000		

Section headers

Section name	Virtual size	Virtual address	Size of raw data	Pointer to raw data	Characteristics
.text	00007748	00001000	00007800	00000400	60000020
.data	00001BA8	00009000	00000800	00007C00	C0000040
.rsrc	00008958	0000B000	00008A00	00008400	40000040

We know that, on disk, PE file format resembles very closely the image when *Windows* loads it into memory. The loader uses the memory-mapped file mechanism to map the appropriate section of the file into the virtual address space. So it is very easy to calculate the file-based PE file execution start offset.

The address of entry point that is stored in the optional header is a relative virtual address (RVA), where the loader will begin execution. An RVA is simply the offset of an item, relative to where the file is memory-mapped.

The following are the usual steps that are followed to reach the file execution start offset:

1. Determine each section's virtual memory map, i.e. virtual start address and end address. The virtual address and virtual size for each section can be found in the section header.
2. Determine in which section's virtual space the address of entry point lies.
3. Check the file offset of that section as per the section header. In the section header the pointer to raw data field gives us the file-based offset where the section data/bytes begin.
4. Calculate the difference between the address of entry point and the virtual address of the section in which the entry point lies. Add this difference to the pointer to raw data, which is the file-based offset of the section, in order to get the file-based execution start offset for that file.

In the case of *Notepad*, the address of entry point lies in the .text section, as the .text section starts at 0x00001000 and ends at 0x00008748 and the address of entry point is 0x0000739D. I have not added image base to any values here – since it is common to all RVAs, I can ignore it for calculation purposes.

So the file offset for execution start is:

```
(0x0000739D – 0x00001000) + 0x00000400
```

Here, 0x400 is the pointer to raw data of the .text section, which points to the file offset of the .text section. In this case the offset comes to 0x0000679D, which is where the execution will begin.

So what we see is that the loader reads each section's bytes from the pointer to raw data into a file and maps it to the virtual address given in the section header table. Since the values are RVAs we have to add these to the image base of the file to arrive at the actual pointer. (However, in the example given above I have omitted image base because all the values are RVAs.)

In the case of *Notepad*, you can see from the section header table that the first .text section will be mapped starting from virtual address 0x01001000. This means that the .text section, which begins at 0x400 in the file (0x400 is the pointer to raw data), will be mapped at 0x01001000 in memory.

HAVING A LOOK AT NSPACK-ED PE FILES

Table 2 shows the header information of a typical malicious program that is packed using *NSPack*.

Table 2: Header information for a typical piece of malware that is packed using *NSPack*.

Optional Header

Number of sections:	02	Section alignment:	00001000
Address of entry point:	0000101B	File alignment:	00000200
Image base:	00400000		

Section Headers

Section name	Virtual size	Virtual address	Size of raw data	Pointer to raw data	Characteristics
nsp0	00004000	00001000	0000000B	0000001C	E0000060
nsp1	0000203D	00005000	00000CFD	00000200	E0000060

The file execution start offset for this file is calculated as follows:

```
(0x0000101B – 0x00001000) + 0x0000001C = 0x00000037
```

But this is not the offset where file execution actually starts. The *Windows* loader rounds the pointer to raw data to 0x00000000 because it is less than the file alignment value (which is 0x00000200). This way, the loader assumes that the first section, nsp0, starts at file offset 0 and loads the section accordingly in the memory. So if we round the pointer to raw data, as the loader does, the file execution start offset is calculated as follows:

```
(0x0000101B - 0x00001000) + 0x00000000 = 0x0000001B
```

The offset 0x0000001B proves to be somewhere in the DOS header of the PE file. It lands in the reserved part of the DOS header – which is usually filled with zeros. At this location the packer inserts a five-byte jump instruction which will transfer control to code further ahead.

AV scanners need to implement a check such that, if the pointer to raw data is not a multiple of the file alignment it must be rounded to the nearest multiple and the remaining extra bytes skipped. Malware can avoid detection by an AV scanner that has not implemented such a check. I also observed that, for files whose file alignment value is not 0x00000200, the loader rounds it to a multiple of 0x00000200.

Many AV scanners do handle *NSPack*-ed PE executables correctly and are able to detect the malware. Some have implemented a rule such that the pointer to raw data of the first section is rounded to zero only if it is less than the file alignment – otherwise it is used as it is.

I observed that, even if I modified the pointer to raw data by increasing it by a few bytes (so that it would not be an exact multiple of file alignment), the file worked properly and had no problems in loading. I also checked with executable files whose control lies in different sections (e.g. first, second, or last). Regardless of which section the file control lies in, the pointer to raw data can be changed to any odd figure not just less than file alignment.

In most of the PE files I checked, I observed that the pointer to raw data field had a value that was a multiple of file alignment, so there were no issues of rounding the values or miscalculating. But as I came across some of the recent file packers that newer trojans and other malware are using I found that the packers are using this trick to avoid proper detection or to avoid debugging by standard debugging techniques.

TEST OF AV SCANNERS

I decided to check a number of AV scanners to see whether they had implemented the rule of rounding the pointer to raw data when calculating the file execution start offset.

I decided to use an old polymorphic virus. I selected a polymorphic virus because, where signature viruses are concerned, AV scanners have different methods for detection. Some of them scan for the signature in the few kilobytes at file executable start offset, but some scan the whole file for virus signatures – and in that case we would not be able to tell whether the scanner is calculating the file executable start offset. If the scanner is scanning the entire file, then it may not miss the detection even if we change the pointer to raw data of the control (execution/code) section. In the case of polymorphic viruses, the AV scanner must calculate the file execution start offset in order to reach the virus decryption loop/engine.

I used a sample of W32.CTX, also known as Win95.Marburg.8582, and the *Virus Total* service for this test. I took one sample of W32.CTX and named it CTX_ORG.EXE, then I copied this sample to CTX_CHG.EXE and modified the pointer to raw data of the .text section by increasing it by 0x199 bytes.

The header information of both test files is shown in the tables below.

Table 3: Header information for CTX_ORG.EXE.

Optional Header

Number of sections:	06	Section alignment:	00001000
Address of entry point:	0000E365	File alignment:	00001000
Image base:	00400000		

Section Headers

Section name	Virtual size	Virtual address	Size of raw data	Pointer to raw data	Characteristics
.text	0002912D	00001000	0002A000	00001000	60000020
.rdata	00007AF8	0002B000	00008000	0002B000	40000040
.data	000074A8	00033000	00003000	00033000	C0000040
.idata	00002092	0003B000	00003000	00036000	C0000040
.rsrc	000040E0	0003E000	00005000	00039000	40000040
.reloc	000081D1	00043000	00009000	0003E000	C2000040

Table 4: Header information for CTX_CHG.EXE.

Optional Header

Number of sections:	06	Section alignment:	00001000
Address of entry point:	0000E365	File alignment:	00001000
Image base:	00400000		

Section Headers

Section name	Virtual size	Virtual address	Size of raw data	Pointer to raw data	Characteristics
.text	0002912D	00001000	0002A000	00001199	60000020
.rdata	00007AF8	0002B000	00008000	0002B000	40000040
.data	000074A8	00033000	00003000	00033000	C0000040
.idata	00002092	0003B000	00003000	00036000	C0000040
.rsrc	000040E0	0003E000	00005000	00039000	40000040
.reloc	000081D1	00043000	00009000	0003E000	C2000040

The only difference between CTX_ORG.EXE and CTX_CHG.EXE is that the pointer to raw data of the .text section is modified from 0x1000 to 0x1199 in CTX_CHG.EXE.

After this, I confirmed that both the files could be loaded and executed properly in *Windows 9X* systems and that the virus W32.CTX was activated properly.

The modified file cannot be loaded on *Windows NT*-based platforms as it is not a valid Win32 application. The *NT* loader checks a few more things in the header than *Windows 95*-based systems and thus finds the file suspicious. The PE header can be checked and modified further such that it does work on *Windows 2000* and *XP* systems.

If an AV scanner does not round the pointer to raw data value it will calculate the file execution start offset as 0x199 bytes ahead of the actual execution start offset. Usually, CTX inserts a Jump instruction immediately at the beginning and hence if the scanner is not able to calculate the execution start offset correctly, it will miss the jump to the virus decryption polymorphic loop, will never reach the virus code and will miss the detection.

I submitted both the files for the on-line scanning services provided by *Virus Total* (www.virustotal.com). The results were that CTX_ORG.EXE was detected correctly (as infected) by 22 of the 24 scanners listed there. The file CTX_CHG.EXE was detected correctly by only 13 scanners. Nine AV scanners missed the detection – despite earlier having detected the same virus when it was not modified.

CONCLUSION

Even though the PE file format is quite old it has many more surprises in store, which are to be explored more carefully with respect to the boundary conditions and the OS loader.

There are other issues too, such as invalid information for size of raw data, virtual size or physical address, as these fields are needed both to reach the file execution start offset and often while cleaning a file to return it to its original status.

There is a need for further careful observation of the complete PE header. We still have to explore what else is there in 64-bit PE files.

ACKNOWLEDGEMENTS

I would like to thank Peter Ferrie, who helped me figure out the issue with *NSPack*-ed PE files when I first contacted him, which made me think again about all the PE file header fields and arrive at this issue.

CONFERENCE REPORT

EICAR 2006 IN A NUTSHELL

Eddy Willems
NOXS and EICAR, Belgium

The 15th annual EICAR conference took place last month in the German town of Hamburg. Set on Hamburg's harbour front with stunning views, the Hotel Hafen Hamburg provided an ideal setting for the conference.

The event started with two professional clinics, during which Vlasti Broucek demonstrated some 'Art of data visualisation' and Elizabeth Bates and Bill Haffner explained the 'Security and privacy risks in biometric deployments'.

The clinics took place in the morning, and after lunch the conference was opened officially with a welcome address from Rainer Fahs and a keynote address given by Sarah Gordon. Sarah's address reminded me of the reason I have been coming to this conference for 15 years: security knowledge lies in the details. A panel discussion came next in the schedule. Hosted by David Perry and Sarah Gordon, the discussion, entitled 'Birds of a feather flock together', gave a nice overview of the various anti-malware groups in the industry – like CARO, AVIEN, WildList, etc.

After this, the conference split into a well-planned two-stream programme, featuring some highly accomplished presenters.

I have always found it hard to decide which session to attend in these multiple-stream conferences, and this year it was even harder than before. If you know how to split yourself in two, please share it with me! The following are some of the highlights of the sessions I attended.

Two spam papers grabbed my attention. The first, by Christopher Lueg, Jeff Huang and Michael Twidale of the Universities of Tasmania and Illinois, explained nicely where spam comes from. The second spam paper – and probably the most controversial – was written by John Aycock and Nathan Friess of the University of Calgary. During their presentation they described some new spamming techniques that have not (yet) been seen in the wild. Let's hope spammers do not start to use these techniques.

The second day started with some definitions of crimeware given by Richard Ford (Florida Inst. Technology) and Sarah Gordon (*Symantec*) and spyware given by Jason Bruce (*Sophos*) and Martin Overton (*IBM*). Larry Bridwell (*iCSA Labs*) and Josh Harriman (*Symantec*) showed us some problems relating to spyware testing. Tony Lee and Jigar Mody of *Microsoft* proposed a behaviour-based automated classification method based on distance measure and machine learning.

A controversial paper by Eric Filiol (Army Signal Academy), entitled 'Malware pattern scanning schemes against black box analysis', was rather too theoretical for me, but it proved interesting for the more mathematically-minded delegates.

More practical and accessible to all delegates were the papers 'Enlisting the end-user', given by Jeannette Jarvis (*Boeing*); 'Pharming: a real threat?', given by David Sancho and François Maillard (*Trend Micro*); 'Unpacking – a hybrid approach', given by Vanja Svajcer and Samir Mody (*Sophos*) and 'Evolution from a Honeypot to a distributed honey net', given by Oliver Auerbach (*Avira*).

This year's gala dinner was unusual in that, for the first time in four years, there wasn't a new virus outbreak to talk about. There seemed to be a trend emerging, with the release of Sober.P on the first day of last year's EICAR conference, the appearance of Sasser during the 2004 conference and Bugbear.B during the 2003 conference – but thankfully this year's event was virus free.

The third day of the conference is dedicated to non-academic papers – which tend to be more commercially oriented. Nevertheless, the final day started with one of the most interesting keynote speeches I have heard for a long time: Professor Klaus Brunnstein (University of Hamburg) with 'Inherent technical risks will lead information and knowledge societies into a risk society'.

Most people assume that everything ends after the three official conference days – but not so. In what we call a post-conference programme, two task forces (Awareness and Content Security) meet to discuss and agree on real practical goals and objectives. Our RFID task force has already provided guidelines for implementing RFID technology.

The EICAR 2006 agenda was interesting and varied, and the papers were the best I have seen at an EICAR conference so far. Planning has already begun for the 2007 conference and details will be announced shortly at http://www.eicar.org/. The organizers are looking at Budapest and Barcelona as possible locations – but of course other suggestions are always welcome.

As one of the founding members of EICAR, I remember the first constitutional EICAR conference in Brussels in 1991. A lot has happened, changed and improved during those 15 years. And I fully expect this to continue over the next 15 years.

COMPARATIVE REVIEW

WINDOWS XP

Matt Ham

Yet again the *Windows XP* comparative review is upon us, with the usual throng of products arriving to be tested and to test my patience. On this occasion two new products were submitted: *TrustPort Antivirus* and the rather more famous *Microsoft OneCare*. Rude comments and/or praise for these products can be found later in the review.

As this is the last review I will conduct for *Virus Bulletin*, I had hoped for an easy run overall – sadly this was not the case for several products. Although instability was less common than in previous tests, scanning speeds for some products were even slower than they have been in the past. There were also a number of products in this test whose feature sets can only have been designed by folk who are either totally ignorant of usability or bred for enhanced sadism.

THE TEST SETS

The test sets were aligned to the February 2006 WildList. As always, the contents of the WildList can be viewed at http://www.wildlist.org/.

When I first started anti-virus testing, the WildList consisted of some 300 different viruses, one third of which were boot sector types. I have none-too-fond memories of inserting 90 floppies into a machine for scanning on demand, then repeating the process on access. Thankfully for my successor, this month's tests saw a major, if long foreseen, change in that there are no longer any boot sector viruses that are considered to be in the wild. Similarly anticipated was the fact that all but a small number of macro viruses dropped out of the test sets this month, including all *Excel* and WM/ samples.

Numerous other files also dropped out of the test set this month – and, as ever, yet more were added to replace them. Overall numbers in the test set increased marginally; more than 100 samples were added and not quite as many removed. Samples of W32/Rbot, W32/Mytob and W32/Sdbot accounted for the majority of these changes and, together, these three fill around half of the space in the WildList.

AhnLab V3Pro 2004 6.0.0.574

ItW Overall	97.51%	**Macro**	98.94%
ItW Overall (o/a)	97.51%	**Standard**	96.45%
Polymorphic	83.60%		

Starting the line-up on this occasion, *AhnLab*'s *V3Pro* managed one of the slowest installation routines I have witnessed. It also demonstrated some odd logging behaviour, so that detection was performed ultimately by deletion of infected files.

Unfortunately, a false positive and a suspicious file in the clean test set were sufficient to deny *AhnLab* a VB 100% this month, though scanning of these files was notably speedy. In addition there were numerous misses of samples in the In the Wild (ItW) test set, which suggests that slow updates could be the problem here.

Alwil avast! 4.7.829

ItW Overall	100.00%	**Macro**	99.56%
ItW Overall (o/a)	100.00%	**Standard**	99.09%
Polymorphic	93.58%		

As ever, on-access detection for *avast!* was performed by copying the test set and deleting infected files – on-access scanning is not triggered simply by opening files. *avast!* also suffered from a round of false positives – a total of three being sufficient to dash any hopes of a VB 100%. However, there were no misses during the scanning of infected files in the ItW test set, and misses elsewhere were at the same low background level as ever.

Avira AntiVir 330 7.00.00.07

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

At first glance, *AntiVir* looked very much to be taking a step backwards in this version, since many options seemed no longer to be present. Happily, it turned out that these are merely somewhat hidden in the default interface view. With this minor hitch disentangled, *AntiVir* went on to detect all infected files in all test sets – a performance that earned the product a well-deserved VB 100% award.

CA eTrust (InoculateIT engine) 8.0.403.0 23.71.145.0

ItW Overall	100.00%	**Macro**	99.90%
ItW Overall (o/a)	100.00%	**Standard**	99.51%
Polymorphic	99.89%		

On-access tests	ItW		Macro		Polymorphic		Standard	
	No. missed	%	No. missed	%	No. missed	%	No. missed	%
AhnLab V3Pro	19	97.51%	50	98.94%	2236	83.60%	63	96.45%
Alwil avast!	0	100.00%	18	99.56%	112	93.58%	18	99.09%
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	0	100.00%
CA eTrust (InoculateIT)	0	100.00%	4	99.90%	1	99.89%	4	99.51%
CA eTrust (Vet)	0	100.00%	10	99.88%	1	99.95%	3	99.84%
CAT Quick Heal	1	99.87%	86	97.96%	314	96.55%	153	92.81%
Central Command Vexira	3	99.61%	0	100.00%	126	92.58%	25	99.12%
Command Authentium	0	100.00%	0	100.00%	0	100.00%	4	99.67%
Doctor Web Dr.Web	0	100.00%	0	100.00%	0	100.00%	3	99.69%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet FortiClient	0	100.00%	0	100.00%	51	97.37%	6	99.79%
FRISK F-Prot	1	99.87%	0	100.00%	6	99.97%	6	99.49%
F-Secure Anti-Virus	0	100.00%	0	100.00%	0	100.00%	3	99.85%
GDATA AntiVirusKit	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG	0	100.00%	3	99.93%	257	85.97%	31	98.35%
Hauri ViRobot	0	100.00%	44	98.82%	5785	69.52%	271	83.61%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Microsoft OneCare	0	100.00%	0	100.00%	31	97.67%	12	99.37%
MicroWorld eScanWin	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Norman Virus Control	0	100.00%	0	100.00%	175	92.96%	12	99.45%
NWI Virus Chaser	0	100.00%	0	100.00%	0	100.00%	3	99.69%
SOFTWIN BitDefender	0	100.00%	13	99.69%	7	99.77%	17	99.27%
Sophos Anti-Virus	0	100.00%	8	99.80%	0	100.00%	15	99.30%
Symantec AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%
TrustPort Antivirus	22	97.47%	3	99.98%	14	99.24%	30	99.52%
VirusBuster VirusBuster	2	99.74%	0	100.00%	126	92.58%	25	99.12%

Having progressed to version 8, both the *eTrust* products now rejoice in a new interface. However, the new interface seems to prioritise looking new and trendy over being intuitive and easy to use.

Something I found to be particularly irritating was the fact that the interface is launched as HTML in a browser window which is almost unusable on any lower resolution screens.

I was hoping for an improvement in *eTrust*'s reporting of infections. However, hard to credit though it is, on-screen reporting proved to be even worse than it had been previously. In this version of the product infections are reported in a tiny text box which, by default, is truncated and cannot be resized.

It is thus impossible to tell which files are infected through the use of the on-screen display. This can be overcome by printing the log file, though there is no obvious way of obtaining a useful version of this as a file.

As in previous comparative reviews, this version of *eTrust* is not eligible for a VB 100% award, since the *InoculateIT* engine is not the product's default.

CA eTrust (Vet engine) 8.0.403.0 12.4.2191.0

ItW Overall	100.00%	**Macro**	99.88%
ItW Overall (o/a)	100.00%	**Standard**	99.96%
Polymorphic	99.95%		

Of course, the comments made in the previous section also apply to this version of *eTrust*. As mentioned, the *Vet* engine is the default for use in scanning – in fact *eTrust* reverts back to *Vet* on each restart of the GUI.

June 2006 VIRUS BULLETIN 100% www.virusbtn.com

Despite the interface woes, *eTrust*'s detection rates were up to their usual good levels, and since no false positives were detected in the clean test set a VB 100% is the result. Scanning speeds were also good for both of the engines.

CAT Quick Heal 2006 8.00

ItW Overall	99.87%	**Macro**	98.23%
ItW Overall (o/a)	99.87%	**Standard**	96.51%
Polymorphic	96.58%		

Problems for *CAT* started in the clean test sets, where the generation of a false positive denied the product any chance of a VB 100% immediately. On a truly bizarre front, *Quick Heal* reported internally that all scans of clean objects took exactly one hour each. In reality, scanning speeds were good. Unfortunately, there was a second major disappointment for *CAT* in that samples of W32/Bagle.X were missed in the ItW test set.

Central Command Vexira Antivirus 2006 5.002 33

ItW Overall	99.61%	**Macro**	100.00%
ItW Overall (o/a)	99.61%	**Standard**	99.27%
Polymorphic	90.27%		

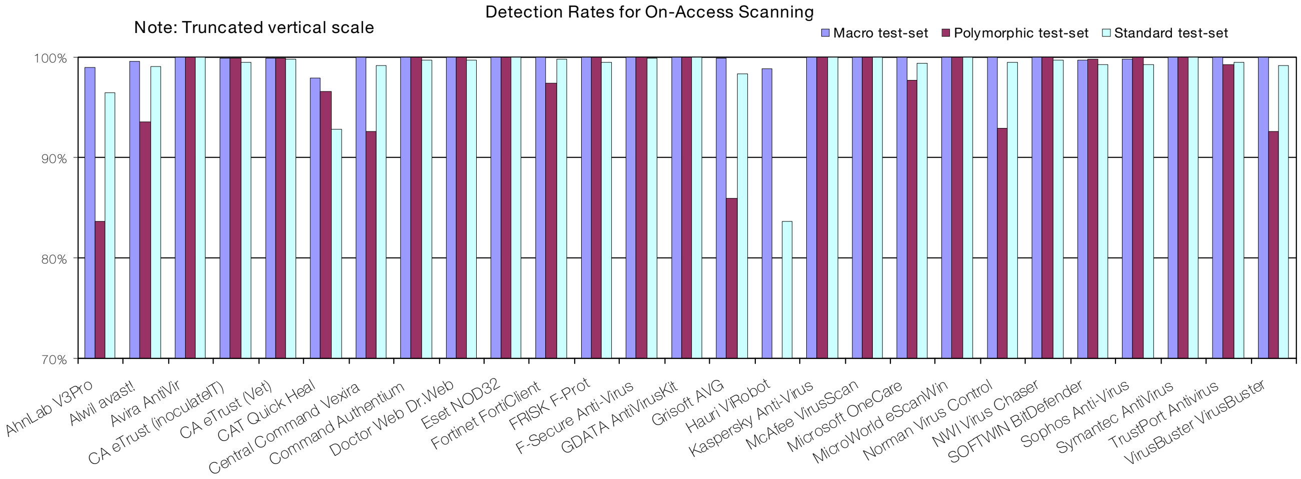
Vexira bears a very close resemblance to *VirusBuster* – which can be explained by the fact that it is a rebadged version of *VirusBuster*. Purists might point out that one product is red and the other blue, but my advanced skills of observation saw past this dissimulation.

Unfortunately stability was not a strength of this product, which caused a hang on the test machine after on-access scanning.

On demand, matters were substantially worse, with there being repeated crashes while scanning *PowerPoint* files. After this performance had been tolerated for long enough to obtain results, there remained a number of misses of samples in the ItW test set, thus the product was prevented from obtaining a VB 100%.

Command Authentium AntiVirus 4.93.7

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



Once again, the most irritating thing about this product was the log – which is available only in a very truncated RTF format. An extensive search of the machine did not help in finding a useful log, thus infected files were deleted to determine detection rates.

After having jumped through the appropriate hoops, the scanning results were good, with only very few, non-ItW, infected files being missed. As a result, *Authentium* earns itself a VB 100% award.

Doctor Web Dr.Web 4.33.2

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

On the negative side, *Dr.Web*'s on-access monitor *SpIDer Guard* lies about its configuration settings – option changes are only ever implemented after a reboot, a fact not reflected by the interface.

The story improved though, with scanning being perfect on demand, while missing only archived files on access. This performance was certainly ample for a VB 100% to be on its way to *Doctor Web*.

Eset NOD32 1.1517

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

NOD32 was the first product in this month's test with which I could find no real fault. Full detection across all test sets and a lack of false positives leave me little to comment on and earn *Eset* a well-deserved VB 100% to add to its collection.

F-Secure Anti-Virus Client Security 6.01

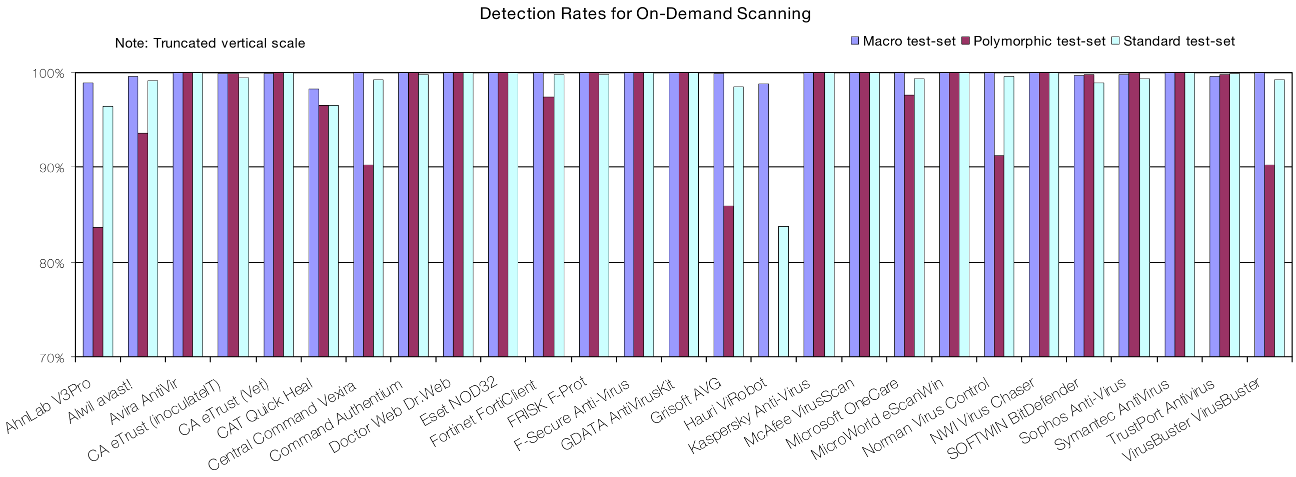
ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.98%
Polymorphic	100.00%		

Another product that displayed no remarkably bad or notably new features, *FSAV* also obtains a VB 100% for its performance. Misses here were limited to viral code, which is a stored rather than directly executable form.

Fortinet FortiClient 2.76 8.459

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.79%
Polymorphic	97.36%		

The trend of good results with few shocks is continued with *Fortinet*'s offering. Although the product missed a noticeable number of polymorphic files, detection results across other test sets were very strong. As a result, *FortiClient* adds another VB 100% to its collection.



On-demand tests	ItW		Macro		Polymorphic		Standard	
	No. missed	%	No. missed	%	No. missed	%	No. missed	%
AhnLab V3Pro	19	97.51%	50	98.94%	2236	83.60%	63	96.45%
Alwil avast!	0	100.00%	18	99.56%	112	93.58%	18	99.09%
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	0	100.00%
CA eTrust (InoculateIT)	0	100.00%	4	99.90%	1	99.89%	4	99.51%
CA eTrust (Vet)	0	100.00%	10	99.88%	1	99.95%	1	99.96%
CAT Quick Heal	1	99.87%	73	98.23%	308	96.58%	98	96.51%
Central Command Vexira	3	99.61%	0	100.00%	624	90.27%	26	99.27%
Command Authentium	0	100.00%	0	100.00%	0	100.00%	1	99.82%
Doctor Web Dr.Web	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet FortiClient	0	100.00%	0	100.00%	55	97.36%	6	99.79%
FRISK F-Prot	0	100.00%	0	100.00%	0	100.00%	1	99.82%
F-Secure Anti-Virus	0	100.00%	0	100.00%	0	100.00%	1	99.98%
GDATA AntiVirusKit	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG	0	100.00%	3	99.93%	257	85.97%	28	98.50%
Hauri ViRobot	0	100.00%	44	98.82%	5785	69.52%	269	83.73%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%
McAfee VirusScan	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Microsoft OneCare	0	100.00%	0	100.00%	31	97.67%	12	99.37%
MicroWorld eScanWin	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Norman Virus Control	0	100.00%	0	100.00%	179	91.25%	5	99.62%
NWI Virus Chaser	0	100.00%	0	100.00%	0	100.00%	0	100.00%
SOFTWIN BitDefender	0	100.00%	13	99.69%	7	99.77%	22	98.91%
Sophos Anti-Virus	0	100.00%	8	99.80%	0	100.00%	15	99.30%
Symantec AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%
TrustPort Antivirus	4	99.95%	19	99.61%	5	99.76%	4	99.91%
VirusBuster VirusBuster	2	99.74%	2	99.98%	624	90.27%	26	99.27%

FRISK F-Prot Antivirus 3.16f

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	99.87%	**Standard**	99.82%
Polymorphic	100.00%		

Unfortunately, the run of products displaying excellent results and few faults is cut short here, since all was not perfection for *F-Prot*. Scanning speeds were fair, but unfortunately a smattering of misses across the test sets included a sample of W32/Aimbot, which is classified as in the wild.

A VB 100% award therefore is out of the grasp of *FRISK* on this occasion.

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files		Dynamic	
	Time (s)	Throughput (kB/s)	FPs [susp]	Time(s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	Time(s)	Throughput (kB/s)	Time(s)	Throughput (kB/s)
AhnLab V3Pro	51.0	12620.6	1	9.0	8814.9		150.0	1062.8	25.0	2984.3	6.0	8040.4
Alwil avast!	154.0	4179.5	3	42.0	1888.9		44.0	3623.1	21.0	3552.7	48.0	1005.1
Avira AntiVir	240.0	2681.9		8.0	9916.7		179.0	890.6	15.0	4973.8	108.0	446.7
CA eTrust (InoculateIT)	75.0	8582.0		10.0	7933.4		100.0	1594.2	21.0	3552.7	22.0	2192.8
CA eTrust (Vet)	91.0	7073.1		13.0	6102.6		100.0	1594.2	25.0	2984.3	17.0	2837.8
CAT Quick Heal	78.0	8251.9	1	25.0	3173.4		73.0	2183.8	27.0	2763.2	35.0	1378.4
Central Command Vexira	258.0	2494.8		39.0	2034.2		187.0	852.5	43.0	1735.1	70.0	689.2
Command Authentium	109.0	5905.0		5.0	15866.8		43.0	3707.4	5.0	14921.5	19.0	2539.1
Doctor Web Dr.Web	263.0	2447.3		11.0	7212.2		88.0	1811.6	14.0	5329.1	26.0	1855.5
Eset NOD32	37.0	17395.9		3.0	26444.6		31.0	5142.5	7.0	10658.2	16.0	3015.2
Fortinet FortiClient	235.0	2738.9		10.0	7933.4		145.0	1099.4	10.0	7460.7	16.0	3015.2
FRISK F-Prot	149.0	4319.8		6.0	13222.3		71.0	2245.3	8.0	9325.9	31.0	1556.2
F-Secure Anti-Virus	181.0	3556.1		16.0	4958.4		84.0	1897.8	21.0	3552.7	28.0	1723.0
GDATA AntiVirusKit	412.0	1562.3		32.0	2479.2		175.0	911.0	62.0	1203.3	61.0	790.9
Grisoft AVG	225.0	2860.7		7.0	11333.4		75.0	2125.6	9.0	8289.7	27.0	1786.8
Hauri ViRobot	457.0	1408.4	1+[1]	101.0	785.5		321.0	496.6	116.0	643.2	144.0	335.0
Kaspersky Anti-Virus	1272.0	506.0		17.0	4666.7		50.0	3188.3	18.0	4144.9	170.0	283.8
McAfee VirusScan	154.0	4179.5		9.0	8814.9		73.0	2183.8	17.0	4388.7	18.0	2680.1
Microsoft OneCare	419.0	1536.2		9.0	8814.9		246.0	648.0	14.0	5329.1	115.0	419.5
MicroWorld eScanWin	411.0	1566.1		34.0	2333.3		152.0	1048.8	64.0	1165.7	59.0	817.7
Norman Virus Control	944.0	681.8		7.0	11333.4		176.0	905.8	8.0	9325.9	148.0	326.0
NWI Virus Chaser	248.0	2595.4		12.0	6611.1		91.0	1751.8	14.0	5329.1	24.0	2010.1
SOFTWIN BitDefender	398.0	1617.2		13.0	6102.6		186.0	857.1	16.0	4663.0	66.0	730.9
Sophos Anti-Virus	99.0	6501.5		17.0	4666.7		55.0	2898.5	17.0	4388.7	21.0	2297.3
Symantec AntiVirus	176.0	3657.1		12.0	6611.1		65.0	2452.6	11.0	6782.5	11.0	4385.7
TrustPort Antivirus	1145.0	562.1		20.0	3966.7		325.0	490.5	24.0	3108.6	175.0	275.7
VirusBuster VirusBuster	288.0	2234.9	1	44.0	1803.0		191.0	834.6	49.0	1522.6	75.0	643.2

GDATA AntiVirusKit 2006 16.0.7

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

Despite a somewhat slow performance, *GDATA* managed full detection of all samples in all categories, with no false positives. *AVK*'s developers should be pleased with this performance, and a VB 100% should add to their contentment.

June 2006 100% VIRUS BULLETIN www.virusbtn.com

Grisoft AVG Anti-Virus 7.1.392

ItW Overall	100.00%	**Macro**	99.93%
ItW Overall (o/a)	100.00%	**Standard**	98.50%
Polymorphic	85.97%		

One of the more common user queries I have been faced with during my time at *Virus Bulletin* concerns how to delete infected files using *AVG*. Having tried to do so, the frequency of complaints no longer surprises me. Numerous files, although

June 2006 100% VIRUS BULLETIN www.virusbtn.com flagged as infected, were not subject to any automated deletion or disinfection.

Apart from this there were no surprises in either the clean or infected test sets, with a VB 100% being the pleasing result for *Grisoft*.

Hauri ViRobot 5.0

ItW Overall	100.00%	**Macro**	98.82%
ItW Overall (o/a)	100.00%	**Standard**	83.73%
Polymorphic	69.52%		

Unfortunately, *Hauri*'s chances of gaining a VB 100% evaporated with a false positive and suspicious file noted in the clean set – and scanning rates were not particularly speedy here either.

Misses in detecting infected files were plentiful too, although looking on the brighter side, none of the missed detections occurred in the ItW set.

Kaspersky Anti-Virus 6.0.0.299

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

KAV includes various self-protection features which turn out to be a double-edged sword. The less-than-welcome aspect is that the virus definitions are so well protected that they are, by default, unable to be updated manually. Since the update function does not allow updates from a local folder, this is somewhat irritating.

June 2006
100%
VIRUS BULLETIN
www.virusbtn.com

There also seem to have been some changes in scanning methods, the effects of which are particularly unpleasant. On-access scanning was seemingly interminable, while the clean set scanning rate is pretty indicative of the speeds seen while scanning the infected sets. This is not an effect of low scanning priorities however – during scanning *KAV* remained steadily at 99% processor usage.

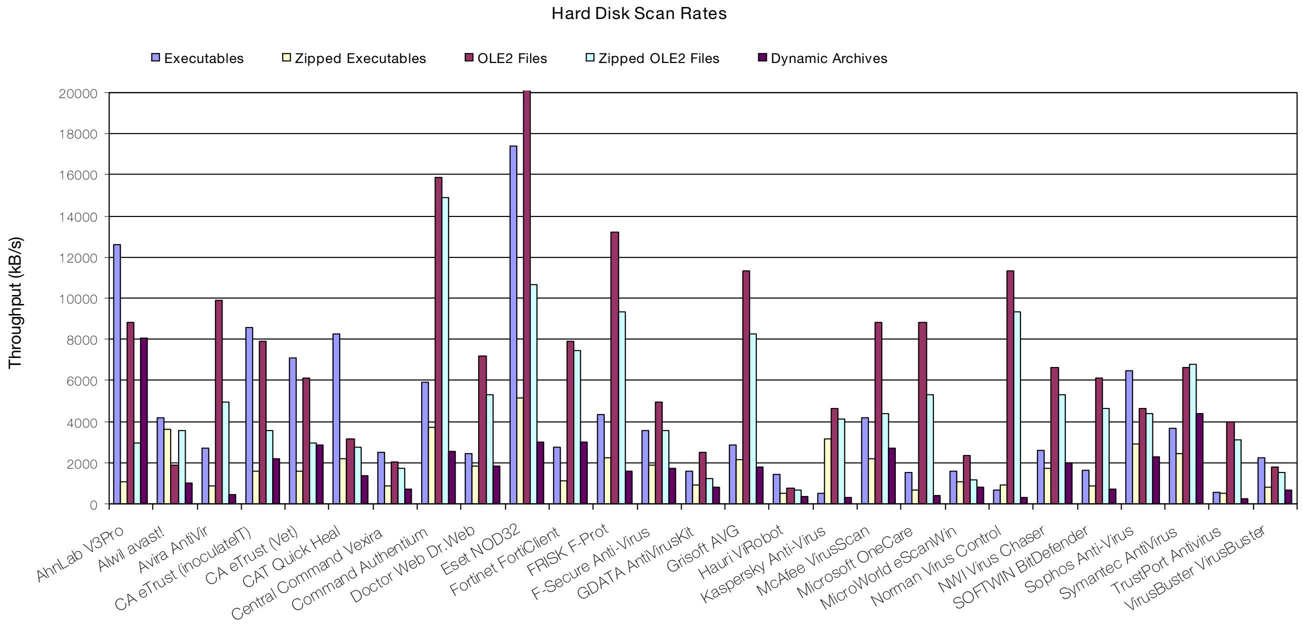
All of this work was, at least, for good reason as all files in all test sets were detected and no false positives were produced. A VB 100% award thus acts as a distraction from the various problems encountered.

McAfee VirusScan Enterprise 8.0i 4400 4753

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

Happily, with *VirusScan* we return to a product that had no nasty surprises in store and gave a good performance with full detection of infected samples across all test sets. With no false positives noted in the

June 2006
100%
VIRUS BULLETIN
www.virusbtn.com



clean test sets either, *VirusScan* is awarded a well deserved VB 100%.

Microsoft Windows Live OneCare 1.0.0971.12

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.37%
Polymorphic	97.67%		

As might be expected of a *Microsoft* product, *OneCare* operates in the guise of paranoid nanny. The user is not trusted to make many decisions of their own, which made certain parts of the test process frustrating.

The progress counter that is displayed during scans is particularly laughable, reaching 99% in ten minutes and then remaining at that point for approximately another 20 minutes or so. This is a result of the automatic disinfection and quarantine (the user has no say in the matter). Indeed, *Microsoft*'s idea of quarantining is somewhat novel, consisting of appending what looks like a checksum to the end of the file name.

What with constantly resetting the areas to be scanned and hanging after the on-access scan, this product cannot be said to be one of my favourites. However, its detection rates were sufficient for a VB 100% to be in order.

MicroWorld eScanWin 8.0.659.1

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

eScan is a rebadged version of *GDATA*'s *AntiVirusKit*, so it should come as no great surprise that the results for *eScan* include full detection of samples across all test sets, a VB 100% award and no adverse comment.

With little else to say, let's move on to a product that behaved badly instead.

Norman Virus Control 5.81

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.62%
Polymorphic	91.25%		

Having been a source of frustration in previous reviews (see *VB*, April 2006, p.17), *Norman Virus Control* continued to manifest new problems on this occasion.

On-access scanning was subject to repeated crashes, whether dealing with infected or previously disinfected files. The effects were sufficient to reduce *Windows* to a state of complete paralysis, in which only a hard reboot had any effect on the test machines.

Upon reboot the splash screen displays the question 'Would you go for anything but green?' (green being *Norman*'s corporate colour). My answer would be that *anything* would be better than this.

Unfortunately for the forces of truth and justice, after strenuous efforts scanning results were sufficient to warrant a VB 100% for this shockingly behaved product.

New Technology Wave (NWI) Virus Chaser 5.09

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

Since *Virus Chaser* is a rebadged version of *Dr.Web*, it should come as little surprise that it shares both the irritations and praise of that product.

With faultless detection rates across all the test sets and no false positives noted in the clean test set, a VB 100% can be included in the shared experience.

SOFTWIN BitDefender 9 7.06632

ItW Overall	100.00%	**Macro**	99.69%
ItW Overall (o/a)	100.00%	**Standard**	98.91%
Polymorphic	99.77%		

There were few notable moments during the testing of *BitDefender*, though the scanning of clean executables was certainly slow enough to be tedious to oversee.

As far as detection was concerned, *BitDefender* had a small number of missed detections, although no real pattern was discernable among them. Happily for *SOFTWIN*, however, there were no misses in the ItW set and no false positives were picked up in the clean test set, thus *BitDefender* also earns a VB100%.

Sophos Anti-Virus 5.2.0

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

Sophos's product was as well behaved as ever. Whether it was practice with the GUI or some small changes in it, something made its use seem very much simpler than I can remember it having been recently, which is always a plus point. With an admirable performance across the test sets, a VB100% is in order for the *Sophos* product.

Symantec AntiVirus 10.0.0.359

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

The *Symantec* GUI has remained the same for many years and on this occasion the product's full detection rate across all test sets leaves little scope for discussion. Not even my pathological hatred of the colour yellow can detract from the fact that the product's performance was ample for *SAV* to be awarded a VB 100%.

TrustPort Antivirus 1.6.0.807

ItW Overall	99.95%	**Macro**	99.61%
ItW Overall (o/a)	97.47%	**Standard**	99.91%
Polymorphic	99.76%		

Since this product is based on a combination of *BitDefender* and *Norman* scanning engines, I was fearful, when I first launched *TrustPort*, that its scanning performance would resemble blue whales forced into pogo-stick races. Thankfully, scanning speeds were not absolutely terrible, just pretty bad.

The combination of the two engines may be responsible for one of *Trustport*'s oddities, namely that it reported many more files as having been scanned than actually existed in the test sets. A further mystery was the variation in the actions taken upon detection of a virus. Using the default settings, samples were deleted, disinfected, quarantined, renamed and simply left to fester, all in the course of one scan.

All this aside, the detection rates demonstrated by the product came close to decent, but there were a sufficient number of ItW misses to deny *TrustPort* a VB 100%.

VirusBuster Professional 2006 5.2 33

ItW Overall	99.74%	**Macro**	99.98%
ItW Overall (o/a)	99.74%	**Standard**	99.27%
Polymorphic	90.27%		

Not surprisingly, *VirusBuster* suffered some of the same woes as *Vexira*, though thankfully to a lesser extent. Instability on demand resulted in scanning simply not being available after existing scans aborted while in progress. Only a reboot solved this broken state. Misses of samples in the ItW test set merely added to these woes, meaning that *VirusBuster* was denied a VB 100% on this occasion.

As a side note, after discussion with the developers, the reason for the scanning speed issues which plagued *VirusBuster* in the *Linux* comparative review (see *VB*, April 2006, p.13) was determined to be the handling of alert messages. In the default setting, alerts are sent to the client and if the client is set such that it will not accept these alerts, then the sending will wait until it times out. Since the client is set, by default, not to accept these alerts, this causes a dramatic slowdown in scanning rates. Clearly this problem can be solved easily by some simple changes in the client or scanner configuration.

CONCLUSION

My final words should be statements, grave judgements and moments of prescience, so as to leave a lasting memory of the quality of my reviews. Unfortunately for this line of thinking, the only thoughts I have to offer are of a cynical nature.

The names and descriptions of the threats may change, but the anti-virus industry remains pretty much the same as it ever has been. The major companies are the same, user ignorance is unchanged and the hyperbolic press releases are the same. Even the claims that 'soon all will change' are simply repeats of the past. If I should return to the anti-virus field in the future, I really don't think it would take more than a few minutes to become re-acclimatised – I just hope that *NetWare* is extinct by then.

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 XP Professional SP2*.

Virus test sets: Complete listings of the test sets used can be found at http://www.virusbtn.com/Comparatives/WinXP/2006/test_sets.html.

A complete description of the results calculation protocol can be found at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

END NOTES & NEWS

CSI NetSec '06 takes place 12–14 June 2006 in Scottsdale, AZ, USA. Topics to be covered at the event include: wireless, remote access, attacks and countermeasures, intrusion prevention, forensics and current trends. For more details see http://www.gocsi.com/.

A security briefing entitled 'Understanding and overcoming security risks' will be held on 13 June 2006 in Stirling, Scotland, and again on 20 June 2006 in London, UK. The briefings, hosted by Dimension Data in association with *CipherTrust*, *Bluecoat* and *PGP* will cover topics including how to overcome threats to email systems, web infrastructure, information integrity and compliance. For further information or to reserve a place contact Dan Trotman on dan.trotman@ciphertrust.com or call 01235 448563.

The SecureLondon Seminar will be held on 20 June 2006 in London, UK. The SecureParis event has been postponed until 1 February 2007. For details see https://www.isc2.org/cgi-bin/isc2event_information.cgi.

The First Conference on Advances in Computer Security and Forensics (ACSF) will be held in Liverpool, UK, 13–14 July, 2006. The conference aims to draw a wide range of participants from the national and international research community as well as current practitioners within the fields of computer security and computer forensics. For details, see http://www.cms.livjm.ac.uk/acsf1/.

Secure Malaysia 2006 will be held 24–26 July 2006 in Kuala Lumpur, Malaysia. Secure Malaysia is co-hosted by National ICT Security & Emergency Response Centre (NISER).The show will be held alongside CardEx Asia and Smart Labels 2006. See http://www.protemp.com.my/.

Black Hat USA 2006 will be held 29 July to 3 August 2006 in Las Vegas, NV, USA. See http://www.blackhat.com/.

The 15th USENIX Security Symposium takes place 31 July – 4 August 2006 in Vancouver, B.C., Canada. A training programme will be followed by a technical programme, which will include refereed papers, invited talks, work-in-progress reports, panel discussions and birds-of-a-feather sessions. A workshop, entitled Hot Topics in Security (HotSec '06), will also be held in conjunction with the main conference. For more details see http://www.usenix.org/.

ECCE2006 will be held 12–14 September 2006 in Nottingham, UK. This will be the second E-Crime and Computer Evidence Conference to be held in Europe. For full details, including a call for papers, see http://www.ecce-conference.com/.

The Gartner IT Security Summit 2006 takes place 18–19 September 2006 in London, UK. For full details see http://europe.gartner.com/security/.

HITBSecConf2006 will take place 18–21 September 2006 in Kuala Lumpur. Further details and a call for papers will be announced in due course at http://www.hackinthebox.org/.

T2'06 will be held 28–29 September 2006 in Helsinki, Finland. The conference focuses on newly emerging information security research. All presentations will be technically oriented, practical and include demonstrations. See http://www.t2.fi/uutisia.en.html.

The SecureLondon Workshop will be held on 3 October 2006 in London, UK. For details see https://www.isc2.org/cgi-bin/isc2event_information.cgi.

Black Hat Japan 2006 takes place 5–6 October 2006 in Tokyo, Japan. Unlike other Black Hat events, Black Hat Japan features Briefings only. For more information see http://www.blackhat.com/.

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. The full programme is now available at http://www.virusbtn.com/.

RSA Conference Europe 2006 takes place 23–25 October 2006 in Nice, France. See http://2006.rsaconference.com/europe/.

AVAR 2006 will be held 4–5 December 2006 in Auckland, New Zealand. See http://www.aavar.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

GONE PHISHING IN JAPAN

Police in Kyoto have arrested eight men in Japan's first crackdown on organized phishing. The men are suspected of belonging to a phishing ring which stole personal information from some 1,000 visitors to a fake *Yahoo!* auctions site.

Visitors were lured to the fake site through emails which were designed to look as if they had been sent from administrators of the genuine *Yahoo!* site. However, a URL in the email directed unsuspecting users to the fake auction site, where they were required to enter their IDs and passwords. The group collected these login details and used them to access the genuine auction site, putting a number of dummy items up for sale. Those who made seemingly successful bids on the items unwittingly sent their payments to the group's bank accounts – and of course, did not receive their goods.

The phishing ring is believed to have defrauded 700 people of around 100 million yen (approx. £474,000).

BLUE FROG CROAKS BUT MAY RISE AGAIN

Last month, *Blue Frog*, the anti-spam service offered by *Blue Security*, was forced to roll over and accept defeat after suffering a retaliatory attack from a spammer.

Blue Security championed a DIY-style anti-spam campaign in which the company's half a million customers were encouraged to send replies to the spam they received. The idea was that the resulting traffic would overload the spammers' servers and hamper their email-sending activity severely. Indeed, some spam companies did agree to stop mailing *Blue Security*'s customers.

Last month, however, the company's website, along with those of many of its partners, was hit by a denial-of-service attack, which is believed to have originated from a particular Russian spammer. In addition to the DoS attack the company was threatened with a second attack that the attacker claimed would include a computer virus unless the company ceased its activity. The company felt that it had no choice than to close its anti-spam operations.

Now, however, two software developers are attempting to recreate a more robust, open source version of *Blue Security*'s anti-spam service. The developers announced their intentions in a *CastleCops* forum, and are searching for interested parties to participate in the project and lend support.

The project is named the *Okopipi Project*, Okopipi being the Amazon Indian name for the blue poison dart frog found in Suriname, South America. According to the project's founders, 'The rules of engagement would be the same as *Blue Frog*. One spam equals one opt-out request. No DDoS. We [will] use bandwidth throttling [that is] sufficiently low to not overwhelm the site. It proved effective before. We see no need to change this. All actions will be approved by a steering committee.' Comments and suggestions have been invited on the fledgling project – for full details, or to sign up to development and general discussion mailing lists, see http://www.okopipi.org/.

EVENTS

The EU Spam Symposium will be held 15 June 2006 at the University of Maastricht, the Netherlands. Full details can be found at http://www.spamsymposium.org/.

The European Summer General Meeting and SuperSummit of the Anti Phishing Working Group (APWG) will be held 27–28 June 2006 in Brussels, Belgium. Full details can be found at http://www.antiphishing.org/.

The third Conference on Email and Anti-Spam, CEAS 2006, will be held 27–28 July 2006 in Mountain View, CA, USA. The conference encompasses a broad range of issues relating to email and Internet communication. Full details can be found at http://www.ceas.cc/.

The Text Retrieval Conference (TREC) 2006 will be held 14–17 November 2006 at NIST in Gaithersburg, MD, USA. More details of the TREC 2006 spam track including information on how to participate can be found at: http://plg.uwaterloo.ca/~gvcormac/spam/.

FEATURE

SpamOrHam

John Graham-Cumming
Independent consultant, France

Many readers will recall the popular website *HotOrNot* (www.hotornot.com), where visitors could view pictures submitted by the public and vote (on a scale of 1 to 10) whether the person depicted was ‘hot’ or not. *SpamOrHam* (www.spamorham.org) uses the same principle to sort a large collection of emails into those that are spam and those that are genuine messages, or ham.

PUBLIC SPAM CORPUS

SpamOrHam’s first task is to check the sorting of the 2005 TREC Public Spam Corpus. In the January 2006 issue of *Virus Bulletin*, Gordon Cormack described the results of the spam track of the 2005 Text Retrieval Conference (TREC) – for which the 2005 TREC Public Spam Corpus was created (see *VB*, January 2006, p.S2).

The spam track tested a range of spam-filtering technologies against four corpuses of spam and ham. Three of the corpuses were from private individuals and were not released, the fourth, now known as the 2005 TREC Public Spam Corpus, consists of ham messages released during the course of the *Enron* investigation and spam messages drawn from a public archive. All 92,189 messages in the public corpus are available for download at http://plg.uwaterloo.ca/~gvcormac/treccorpus/.

Using a variety of techniques – starting with various existing spam filters, and calling upon humans where the spam filters disagreed – the messages were sorted into spams and hams. The public corpus download includes a file that describes this ‘gold standard’. Details of the creation of the public corpus can be found in Cormack and Lyman’s 2005 CEAS paper ‘Spam corpus creation for TREC’ (http://www.ceas.cc/papers-2005/162.pdf).

PLACE YOUR VOTES

A visitor to *SpamOrHam* is presented with emails drawn randomly from the 2005 TREC Public Spam Corpus in two forms: an image of the email rendered using *Microsoft Outlook 2002* and the complete raw message including full headers and body. The user is invited to click on one of three buttons: ‘This is Spam’, ‘This is Ham’ or ‘I’m not sure’. Each vote by a user is recorded for later comparison with the gold standard.

To ensure that the site is responsive, all 92,189 emails were rendered by importing them into *Microsoft Outlook 2002* and then each message was opened and a screen shot taken which was saved as a GIF with a filename that matches the name of the message in the 2005 TREC Public Spam Corpus. When a user visits the site the server side code reads the raw message from a copy of the corpus on the site and displays the GIF generated.

The generation of the GIF files was one of the most time-consuming tasks in the creation of the *SpamOrHam* site. Ignoring the time taken to write the necessary code and deal with various errors along the way, importing all 92,189 messages into *Microsoft Outlook* took 34 hours on a 2.4 GHz PC running *Windows 2000* with 1 Gb of RAM. Rendering the screen shots of each email took 46 hours with the CPU running at 100% utilization throughout. The rendering generated just under 3 Gb of GIF files.

CAPTCHA

To prevent abuse of the site the user is challenged periodically with a CAPTCHA. The CAPTCHA asks the user to enter a sequence of letters displayed in an image on a fuzzy background (there is also a link to an MP3 file so that disabled users can take part). If the password is not entered correctly the user’s vote is not recorded and they are presented with another CAPTCHA to solve. Once they pass the CAPTCHA test the user will be presented with up to ten emails to vote on before being asked to prove that they are a human with another CAPTCHA image.

One interesting feature of the site is that it stores no state on the server side. The entire state for each user is stored in hidden form fields that are protected using a secure hash. Any attempt to tamper with the form fields, or submit forged information, is detected by the value of the hash. Such fraudulent votes are discarded and a record is kept of the abusive IP address. Further details of this mechanism can be found in this blog entry: http://www.jgc.org/blog/2006/04/stateless-web-pages-with-hashes.html.

Examining the error logs of *SpamOrHam* has shown that although some potential attempts to subvert the purpose of the site have been detected, the biggest problem is that humans have a hard time with the CAPTCHA. Around 20% of the CAPTCHA images presented to users are interpreted incorrectly, leading to a second CAPTCHA being presented.

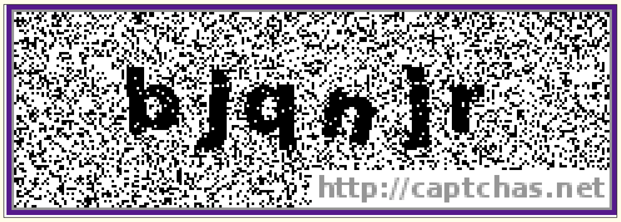


Figure 1: An example CAPTCHA as used by SpamOrHam.

The biggest problem seems to be distinguishing the letters i and l against the fuzzy background.

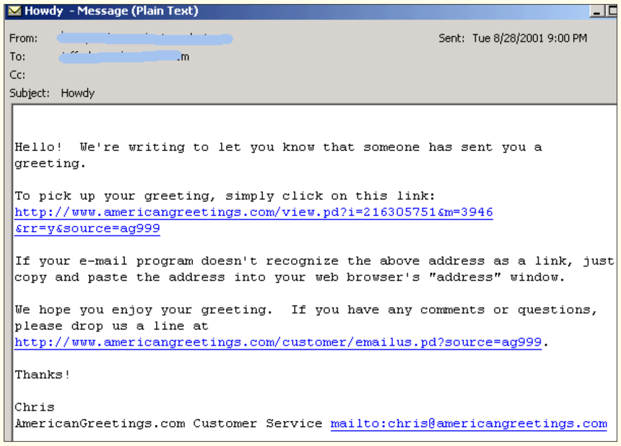
INITIAL RESULTS

SpamOrHam launched on 29 April 2006 and at the time of writing, over 207,000 votes have been cast against the 92,189 messages in the dataset. Around 11,000 messages have not yet been voted on (the expected value for a truly random selection across the messages would be around 9,700; however, due to a bug in the random selection code some messages were not initially being selected – the bug has now been fixed). *SpamOrHam* aims to collect one million votes with the goal that each message be voted on multiple times.

Although the site is only one fifth of the way towards its goal some initial conclusions can be drawn. Of the 81,013 emails voted on by the general public, 53,802 have been voted on more than once and the votes agree with the TREC gold standard. A further 20,707 have been voted on just once while still agreeing. That means that the public and the machine classification of the messages agree on 91.7% of the messages.

The remaining 6,504 messages are divided into three groups: there are 1,894 messages that have been voted on once and the voters disagreed with the gold standard; there are 2,992 messages that have been voted on multiple times but the votes cancel out (for example, one person says spam and another says ham); and there are 1,618 messages where multiple voters have seen an email and the overall votes show disagreement with the TREC gold standard.

Focusing on just these 1,618 messages shows some surprising results (at the time of writing, not all 1,618 have been examined). The overall impression is that, although *SpamOrHam* has found some errors in the gold standard, the ability of people to spot the difference between genuine messages and spam or phishes is open to question.



Howdy - Message (Plain Text)

From: [redacted] Sent: Tue 8/28/2001 9:00 PM
To: [redacted].m
Cc:
Subject: Howdy

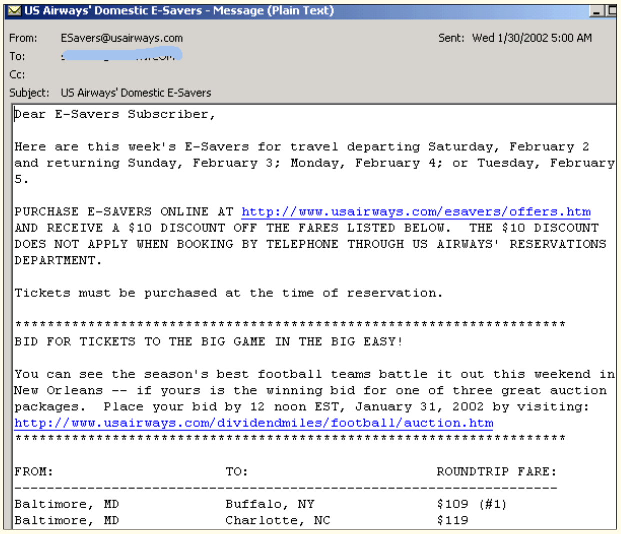
```
Hello!  We're writing to let you know that someone has sent you a
greeting.

To pick up your greeting, simply click on this link:
http://www.americangreetings.com/view.pd?i=216305751&m=3946
&rr=y&source=ag999

If your e-mail program doesn't recognize the above address as a link, just
copy and paste the address into your web browser's "address" window.

We hope you enjoy your greeting.  If you have any comments or questions,
please drop us a line at
http://www.americangreetings.com/customer/emailus.pd?source=ag999.

Thanks!

Chris
AmericanGreetings.com Customer Service mailto:chris@americangreetings.com
```

Figure 2: An e-card that many voters think is spam.

Bill Yerazunis, creator of the *CRM114* spam filter, has measured his own accuracy at determining whether a message is spam or not and indicates that he achieves 99.84% accuracy (see http://www.paulgraham.com/wsy.html). In my 2005 MIT Spam Conference presentation 'People and Spam' I reported on a previous test of the general public's ability to sort email messages (see http://www.jgc.org/pdf/spamconf2005.pdf), which yielded an accuracy of 99.46%. The error rate for the *SpamOrHam* test looks like it will be much higher, with humans being able to identify only around 98% of messages correctly.

People's perception of what constitutes spam should worry legitimate email marketers. Figure 2 is an example of a legitimate e-card that members of the public consider to be spam; there are multiple instances of *SpamOrHam* voters considering e-cards to be spam.

The same was true of the legitimate email from *US Airways* shown in Figure 3; multiple *SpamOrHam* voters see it as spam.



US Airways' Domestic E-Savers - Message (Plain Text)

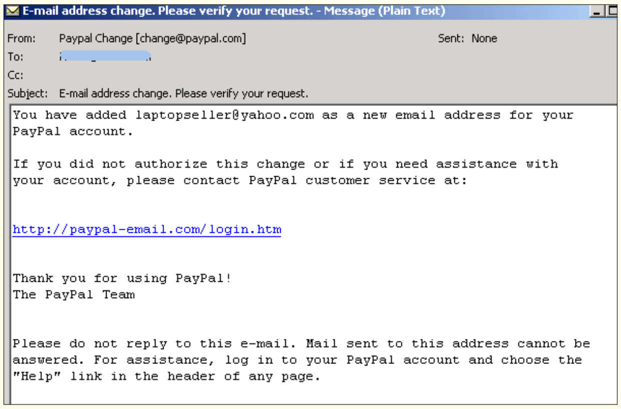
From: ESavers@usairways.com Sent: Wed 1/30/2002 5:00 AM
To: [redacted]
Cc:
Subject: US Airways' Domestic E-Savers

```
Dear E-Savers Subscriber,

Here are this week's E-Savers for travel departing Saturday, February 2
and returning Sunday, February 3; Monday, February 4; or Tuesday, February
5.

PURCHASE E-SAVERS ONLINE AT http://www.usairways.com/esavers/offers.htm
AND RECEIVE A $10 DISCOUNT OFF THE FARES LISTED BELOW.  THE $10 DISCOUNT
DOES NOT APPLY WHEN BOOKING BY TELEPHONE THROUGH US AIRWAYS' RESERVATIONS
DEPARTMENT.

Tickets must be purchased at the time of reservation.

*******************************************************************
BID FOR TICKETS TO THE BIG GAME IN THE BIG EASY!

You can see the season's best football teams battle it out this weekend in
New Orleans -- if yours is the winning bid for one of three great auction
packages.  Place your bid by 12 noon EST, January 31, 2002 by visiting:
http://www.usairways.com/dividendmiles/football/auction.htm
*******************************************************************

FROM:                     TO:                      ROUNDTRIP FARE:
------------------------------------------------------------------
Baltimore, MD             Buffalo, NY              $109 (#1)
Baltimore, MD             Charlotte, NC            $119
```

Figure 3: Legitimate US Airways marketing mail considered to be spam.

And the dangers of phishing are illustrated clearly by the fraudulent *PayPal* message shown in Figure 4, which many voters think is legitimate.

Happily, since this was the original goal, users of *SpamOrHam* have found some errors in the 2005 TREC Public Spam Corpus. The email shown in Figure 5, which was sent to an alumni mailing list, is listed incorrectly as spam in the public corpus, but multiple voters agree that it is legitimate.

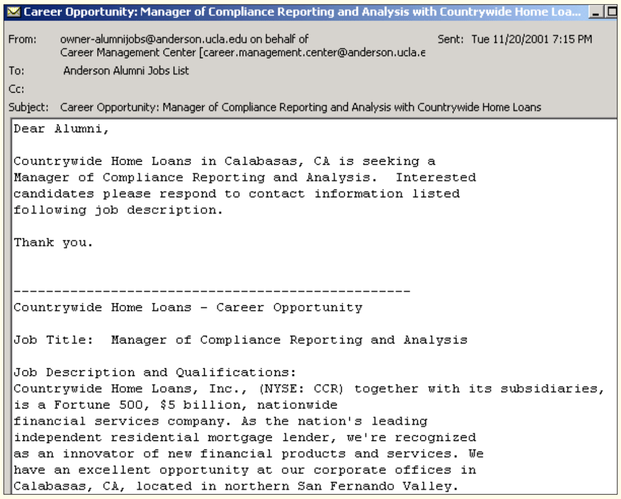


E-mail address change. Please verify your request. - Message (Plain Text)

From: Paypal Change [change@paypal.com] Sent: None
To: [redacted]
Cc:
Subject: E-mail address change. Please verify your request.

```
You have added laptopseller@yahoo.com as a new email address for your
PayPal account.

If you did not authorize this change or if you need assistance with
your account, please contact PayPal customer service at:


http://paypal-email.com/login.htm


Thank you for using PayPal!
The PayPal Team


Please do not reply to this e-mail. Mail sent to this address cannot be
answered. For assistance, log in to your PayPal account and choose the
"Help" link in the header of any page.
```

Figure 4: Many voters have fallen for this PayPal phish.



Career Opportunity: Manager of Compliance Reporting and Analysis with Countrywide Home Loa...

From: owner-alumnijobs@anderson.ucla.edu on behalf of Career Management Center [career.management.center@anderson.ucla.e Sent: Tue 11/20/2001 7:15 PM
To: Anderson Alumni Jobs List
Cc:
Subject: Career Opportunity: Manager of Compliance Reporting and Analysis with Countrywide Home Loans

```
Dear Alumni,

Countrywide Home Loans in Calabasas, CA is seeking a
Manager of Compliance Reporting and Analysis.  Interested
candidates please respond to contact information listed
following job description.

Thank you.


-------------------------------------------------
Countrywide Home Loans - Career Opportunity

Job Title:  Manager of Compliance Reporting and Analysis

Job Description and Qualifications:
Countrywide Home Loans, Inc., (NYSE: CCR) together with its subsidiaries,
is a Fortune 500, $5 billion, nationwide
financial services company. As the nation's leading
independent residential mortgage lender, we're recognized
as an innovator of new financial products and services. We
have an excellent opportunity at our corporate offices in
Calabasas, CA, located in northern San Fernando Valley.
```

Figure 5: Legitimate email spotted by SpamOrHam voters.

The inaccuracy of humans when sorting email messages has an important effect on the testing of spam filters and the design of anti-spam technologies.

The first difficulty highlighted by *SpamOrHam* is in the creation of test corpuses. If spam filters are to be tested reliably, it is essential that test data (both spam and ham messages) is available and that the test data is split accurately into spams and hams.

Since all tests performed so far on humans filtering messages show that they cannot be trusted to provide 100% accurate results, the results of spam-filtering tests need to be viewed with caution. If a spam filter test says that filter A is 99.2% accurate and filter B is 98.5% accurate, it's not possible to tell which filter is better without knowing the margin for error in the original test dataset.

Taking into account human fallibility probably means that humans have an error rate of up to 2% over large sets of messages. Results of spam filter tests need to account for that initial error rate.

Secondly, many anti-spam products contain a quarantine where suspected spam messages are placed, and users are invited to review the captured messages in an attempt to spot false positives (legitimate messages that have been quarantined mistakenly). Equally, some spam-filtering products invite users to teach the system which messages are spam by forwarding spam messages that they have received mistakenly.

However, if the error rate for humans is high, this feedback loop with the anti-spam product may cause the spam filter to perform more poorly than a filter that receives no feedback. For example, if users report that a legitimate email (such as the *US Airways* marketing mail) is spam, a spam filter may begin quarantining all *US Airways* marketing mail for all users sharing the same anti-spam system. This may mean that per-user configuration is necessary to prevent users from interfering with each other's preferences.

On the other hand, users who fall for phishing emails may be allowing more phishing messages to be delivered if their erroneous retrieval of phishing mails from quarantine causes a spam filter to start letting them through.

Finally, there is much disagreement about the definition of spam (a commonly heard adage in anti-spam circles is: 'one man's spam is another man's ham'). This may be reflected in the treatment of marketing messages in the *SpamOrHam* tests, and anecdotal evidence indicates that users will feed back emails sent from legitimate mailing lists, marking them as spam, as a way to unsubscribe without going through the email marketer's actual unsubscribe option.

This behaviour has been made worse by the practice of some spammers to include unsubscribe links in spam; users who try to unsubscribe in fact receive more spams, having 'confirmed' their email address for the spammer.

CONCLUSION

The *SpamOrHam* test is still in progress. Once one million votes have been registered the complete data from *SpamOrHam* will be made public in the form of raw vote data so that anyone can use it for their own research in conjunction with the 2005 TREC Public Spam Corpus.

In addition to gathering the raw votes, *SpamOrHam* is also recording information about the amount of time people spend examining mail before making a decision about whether a message is spam or not.

This timing data will also be made public. Finally, *SpamOrHam* is actively looking for suggestions on how to analyse the data gathered. Please feel free to drop me a line at jgc@jgc.org with your thoughts.