MARCH 2007

virus BULLETIN

Fighting malware and spam

CONTENTS

ISSN 1749-7027

IN THIS ISSUE

SEEING IN THE DARK

There are only two reasons for traffic to be going into unallocated IP space: either due to some form of misconfiguration or because it is malicious. Jose Nazario reveals how monitoring darknet traffic can throws light on what threats are present.

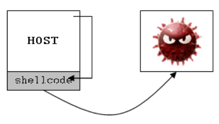
page 2

WHEN TOOLS TURN BAD

Peter Ferrie describes a malicious plug-in for one of the most popular tools of the anti-malware trade: the *IDA* disassembler and debugger.

page 4

STORM IN A P2P CUP



Earlier this year, Trojan.Peacomm (a.k.a. the 'Storm trojan') demonstrated that legitimate P2P protocols can be used effectively to coordinate virus networks. Elia Florio and Mircea Ciubotariu explain how.

page 6

vbSpam supplement

This month: anti-spam news & events; Martin Overton describes a phish with a sting in the tail; and Vipul Sharma and Steve Lewis reveal how they enhanced the efficiency of a learning-based spam filter.

virus BULLETIN COMMENT

'Monitoring darknet traffic yields great visibility into what threats are present.'

Jose Nazario, Arbor Networks

DARKNET MONITORING

The Internet has faced a sustained and significant threat from network malware since the emergence of the global *Windows* network worm in 2001. For instance, in September 2001, the Nimda worm disrupted global BGP routing tables for hours at major Internet peering points, and in January 2002 the SQL Slammer worm caused significant outages and slowdowns. DDoS attacks directed at the root DNS servers, most recently in February 2007, were launched only with the help of Internet-scale malware and botnets.

Discovering this malware and other malicious activity is key to any global monitoring approach, especially an early warning system. Honeypots are often an excellent source of data, but they rely on the attacker encountering the system. By the same token, absorbing all of the data by active client collection techniques simply doesn't scale. Clearly a balance must be found, one that can be used to highlight possible new sources of activity.

A new approach to monitoring malicious Internet traffic on a global basis is to utilize darknets – unallocated IP space owned by service providers or enterprises. This is the portion of the Internet that has not been assigned to customers, and every network has some portion of darknet space. Because it is unallocated, there are only two reasons for traffic to be going into darknet space: either due to some form of misconfiguration or because it is malicious. Darknets have little background traffic, meaning that the data captured is purely signal and can be analysed easily. A key facet of any darknet is that it is globally routed and reachable, so a host anywhere on the Internet sending traffic to it will register from any source.

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*

Darknets work because network-scanning malware is unable to predict which addresses are in use on every level of the Internet. Bots and malware are not intelligent enough to pick and choose where they go, they will simply attempt to spread to as many hosts as possible. Monitoring darknet traffic yields great visibility into what threats are present.

By some estimates, only about one third of the Internet is in active use at any time, counting from the subnet that is DHCP allocated up through the BGP allocations given by organizations such as ARIN. This leaves tremendous room for darknets to be deployed.

Over the years, we have seen only a handful of Internet worms try to avoid the largest of darknet monitors by carrying a list of networks to scan. This didn't work as well as the authors had hoped, and since then very few other malware authors have tried this. The bulk of bots and malware these days use 'island hopping' strategies to bias their scanning and attacks locally, either by hardcoding such an algorithm (first made popular with Code Red II and Nimda) or through botnet scan commands focusing on the local networks. Even in these cases darknets observe the malware due to the sparseness of IP address assignments on the local network.

Multiple levels of data can be analysed in darknets, including NetFlow-based approaches through honeypots. At the NetFlow level, routers and switches show what traffic is destined to a darknet by generating traffic summaries called 'flows'. This provides a lightweight data representation of the traffic by omitting payloads and aggregating packets into a single flow record. These records are great for trend analysis and useful in analysing global scan patterns. Packets can be collected, which, in combination with a honeypot system, can be used to discover the nature of the attack and provide further characterization.

Darknets are the network equivalent of email spam traps, dummy IM accounts and other such data collection points. The major difference between a darknet and a typical honeynet, however, is the scale of data collection. Darknets are composed of hundreds of addresses, rather than one or two hosts. This means that data analysis techniques must scale up dramatically, focusing on trends and patterns instead of deep specifics.

At *Arbor Networks*, we have found that a distributed darknet monitoring system provides global visibility into malicious traffic and probes. The scan and attack patterns indicate the prevalence of bots and malware and a network of sensors collects new malware samples continuously. Because of this, there is usually an indication of a large-scale attack before it impacts customers dramatically.

NEWS

NEWS ROUND-UP

Mobile security was something of a key theme last month, with a rash of new products released alongside updates to those already established in the market. Despite mobile malware still being a relatively minor problem, it is widely expected to become a more significant threat as the mobile market continues to grow and mobile devices become standard business tools. Indeed, last year, telecoms analyst *Juniper Research* predicted a steady rise in attacks on smart phones over the next five years, with an accompanying increase in the size of the mobile security software market – predicting revenues of $5 billion for security product vendors by 2011.

Vendors beginning their foray into mobile security last month included *Sophos*, with its *Sophos Mobile Security* product for *PocketPC* versions of *Windows Mobile 5.0*; *Kaspersky Lab*, with *Kaspersky AntiVirus Mobile 6.0*, covering mobile devices on both *Windows Mobile* and *Symbian* platforms; *Panda*, which revealed the beta version of *Panda Mobile Security* for smart phones based on *Symbian Series 60*; and *BitDefender*, which unveiled the full version of *BitDefender Mobile Security* for both *Symbian* and *Windows Mobile* platforms. Meanwhile, *Symantec*, *Trend Micro* and *F-Secure* announced new versions of their respective mobile anti-virus products.

In what was Oscars month for the film world, AV vendors also received their share of accolades. *McAfee* kicked off the list of winners when its *SiteAdvisor* product was acknowledged by the US Department of Commerce with a 'Recognition of Excellence in Innovation' award. *Eset*, meanwhile, was named an 'Info Security Hot Companies 2007' winner by the *Info Security Products Guide* in recognition of the company's products, people, performance and potential. Finally, readers of *VARBusiness* – a magazine for solution providers and technology integrators – voted *Grisoft*'s *AVG Internet Security 7.5* the number one non-market-leading security software product.

A number of vendors secured significant new business deals last month. South Korean vendor *AhnLab* has set plans in motion for expansion into Latin America after securing a deal with *Banamex*, Mexico's largest bank. *AhnLab* will supply online security services for *Banamex* customers including anti-virus, anti-spyware and firewall protection. Meanwhile, *Norman* entered into a contract to licence its *Sandbox* analysis products to an undisclosed 'major US-based IT company'. Although *Norman* has not revealed the name of the company, it has been identified as a global provider of IT security and services. Finally, *F-Secure* has announced a service agreement with cable TV operator *Canal Digital Norway* to provide security services for *Canal*'s Norwegian broadband customers.

Prevalence Table – January 2007

Virus	Type	Incidents	Reports
W32/Detnat	File	25,790,764	72.50%
W32/Netsky	Worm	3,043,263	8.55%
W32/Mytob	Worm	2,492,084	7.01%
W32/Bagle	Worm	1,313,911	3.69%
W32/MyWife	Worm	735,718	2.07%
W32/Virut	File	478,998	1.35%
W32/Mydoom	Worm	421,345	1.18%
W32/Zafi	File	385,357	1.08%
W32/Lovgate	Worm	286,194	0.80%
W32/Rbot	Worm	148,453	0.42%
W32/Mimail	File	129,622	0.36%
W32/Bagz	Worm	120,578	0.34%
W32/Funlove	File	40,409	0.11%
W32/Parite	File	36,338	0.10%
W32/Stration	Worm	24,080	0.07%
W32/Womble	File	22,803	0.06%
W32/Mabutu	Worm	20,714	0.06%
W32/Bugbear	Worm	14,883	0.04%
VBS/Redlof	Script	10,498	0.03%
W32/Yaha	File	6,594	0.02%
W32/Valla	File	5,030	0.01%
W32/Tenga	File	4,756	0.01%
W32/Sality	File	4,253	0.01%
W32/Small	File	4,144	0.01%
W32/Sober	Worm	3,827	0.01%
W32/Maslan	File	3,807	0.01%
W97M/Thus	Macro	3,493	0.01%
W32/Agobot	Worm	3,093	0.01%
W32/Sobig	Worm	1,986	0.01%
W32/Elkern	File	1,965	0.01%
W32/Darby	File	1,947	0.01%
W95/Spaces	File	1,919	0.01%
Others[1]		10,976	0.03%
Total			100%

[1]The Prevalence Table includes a total of 10,976 reports across 59 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

HIDAN AND DANGEROUS

Peter Ferrie
Symantec Security Response, USA

One of the things that almost all anti-malware researchers have in common is a copy of *Interactive DisAssembler* (*IDA*). It is perhaps the best tool we have for disassembling files, since it is capable of so many important things: it displays the file more or less as it really appears in memory, applying relocations, and resolving imports. *IDA* can follow all of the code paths and note all of the data references, comment the API parameters, and even determine the stack parameters.

Since some people have custom requirements, *IDA* also supports a plug-in interface. Plug-ins can do many things and control many of *IDA*'s actions – including directing it to infect files.

Enter the latest member of the ever-growing W32/Chiton family. The author of the virus calls this one 'W32/Hidan'.

WHERE ARE YOU HIDING?

When Hidan is started for the first time, it queries the default value of the 'HKCR\.idb' registry key. The virus author assumes that if this registry key is present, then *IDA* must also be present on the system. The registry key is created when the *Windows* GUI version of *IDA* is started for the first time. However, the command-line version of *IDA* does not create this key. What's more, there are other tools, such as *Microsoft Visual Studio*, that use the same registry key, so its presence on the system is no guarantee that *IDA* is present.

Regardless of which application created the registry key, the registry value will contain the name of the handler. In the case of *IDA*, the value is 'IDApro.Database32'. The virus then queries the default value of its 'shell\open\command' subkey. The data usually contains the following string:

```
"<path>\idag.exe" "%1"
```

The virus searches this string for the second quote, while remembering the location of the last backslash. Once the second quote is found, the virus appends the string 'plugins' after the last backslash, so the string becomes '<path>\plugins'. This is the directory in which *IDA* keeps its plug-ins.

BEND AND STRETCH

The virus decompresses and drops a plug-in file in the plugins directory, using the (fixed) filename 'hidan.plw'. The file contains only the virus code.

As with the other viruses in the Chiton family, this one is aware of the techniques that are used against viruses that drop files, and will work around all of the commonly used countermeasures: if a file exists already, its read-only attribute (if any) will be removed, and the file will be deleted. If a directory exists instead, then it will be renamed to a random name.

The structure of the dropped file is similar to that of the W32/OU812 member of the W32/Chiton family. However, Hidan differs in one way: the entrypoint of the file is inside the file header itself, which can complicate analysis slightly (particularly when using *IDA*). The file is also not constant, because the virus knows which bytes in the file header are not used, and replaces them with a value chosen randomly at the time of dropping the file.

After dropping the file, the virus runs the host code.

PLUGGING THE HOLE

Apart from dropping the file, the virus performs no other actions in infected files. It simply waits until *IDA* loads the viral plug-in.

When *IDA* starts, it loads all plug-ins that correspond to the platform on which it is running. *IDA* runs on the 32-bit and 64-bit versions of *Windows*, and the 32-bit and 64-bit versions of *Linux*. Additionally, in the *Professional* version of *IDA*, there is a 32-bit *Windows* version that supports 64-bit addressing, so it is capable of loading 64-bit files on a 32-bit machine.

Each of these versions has its own plug-in format and suffix to distinguish them. The suffix 'plw' means that the plug-in is for the 32-bit *Windows* version of *IDA*; 'x86' means that the plug-in is for the 64-bit *Windows* version; 'p64' means that the plug-in is for the 32-bit *Windows* version of *IDA* that supports 64-bit addressing. The suffix 'plx' means that the plug-in is for the 32-bit *Linux* version of *IDA*, and the suffix 'plx64' means that the plug-in is for the 64-bit *Linux* version of *IDA*.

IDA SYMBOLISM

Internally, plug-ins are ordinary DLLs on the *Windows* platform, or shared libraries on the *Linux* platform. Plug-ins must export one special symbol, called either 'PLUGIN' or '_PLUGIN', or it must be ordinal 1 on the *Windows* platform.

The symbol is a pointer to a plug-in structure. The structure contains a field which holds the version number of the SDK used to produce the plug-in. That version number must match the version that *IDA* is expecting, otherwise it will refuse to load the plug-in. There are two versions of the virus: one is compatible with *IDA 4.8*, and one is compatible with *IDA* version 4.9. While the SDK version did not change between *IDA* versions 4.9 and 5.0 (the current version at the time of writing), the behaviour of *IDA* did.

THE TERMINATOR

The plug-in structure contains three pointers to functions, 'init', 'term', and 'run'. The init function is used to initialize the plug-in. The term function is used to terminate the plug-in. The run function is used to run the main part of the plug-in.

In *IDA* prior to version 5.0, if a plug-in init function returned a value of 0, which means that the plug-in cannot or did not want to load (perhaps because the environment is not compatible, or the file to examine is not of the correct format, etc.), *IDA* would free the plug-in directly (via the FreeLibrary() API on the *Windows* platform). Thus, neither the run nor term functions would be called, so the virus author did not include them in the virus.

However, in *IDA* version 5.0, if the term function pointer is non-zero, then *IDA* will call the term function before calling FreeLibrary(). Since the virus does not support this function (there is data at that location, but not a function pointer), the virus crashes at that point, taking *IDA* down with it. This seems a silly bug just to save four bytes of zeros.

Unfortunately, the virus has already done its work by the time it crashes, since the init function contains the replication code.

INITIATION CEREMONY

When the viral plug-in init function is called, it begins by retrieving some file infection-related API addresses from kernel32.dll, the IsFileProtected() API address from sfc.dll (or sfc_os.dll if the platform is *Windows XP/2003*), and two undocumented symbols from ida.wll (RootNode and netnode_value in the .A version, or netnode_valstr() in the .B version).

When the netnode API is called with the value held by the RootNode symbol, an ANSI-format pathname is returned. That pathname corresponds to the file being examined by *IDA*. The virus converts that pathname to Unicode format, then passes it to the IsFileProtected() API.

If the file is not protected, then it will be infected only if it passes a very strict set of filters. These filters include the condition that the file being examined must be a character mode or GUI application for the *Intel* 386+ CPU, that the file must have no digital certificates, and that it must have no bytes outside of the image.

TOUCH AND GO

If the file meets the infection criteria, it will be infected. If relocation data exists at the end of the file, the virus will move the data to a larger offset in the file, and place its own code in the gap that has been created. If there is no relocation data at the end of the file, the virus code will be placed here. The entrypoint is altered to point to the virus code. The virus will calculate a new file checksum, if one existed previously.

Once the infection is complete, the virus forces an exception to occur in order to terminate the replication code and return to the init function. The init function then returns a value of zero to *IDA*, to signal that the plug-in should be unloaded.

The interesting thing is that since *IDA* has already started to load the now-infected file, the change to the entrypoint is not visible – though the virus code can be seen if one knows where to look for it, and if the file is reloaded, the full changes are visible.

Of course, it would be possible for a plug-in to interfere with all of that – the plug-in could remain in memory and intercept disk accesses. It could also restore the host entrypoint label and remove the viral one. However, this seems like an even more pointless exercise than writing the virus in the first place.

CONCLUSION

This member of the Chiton family is just a proof-of-concept virus for a new platform, created by a virus author who specialises in them. Given its simplicity, Hidan might be considered to be hiding in plain sight.

W32/Chiton variant

Type:	Memory-resident parasitic appender/inserter.
Size:	1,321 bytes (.A), 1,330 bytes (.B)
Infects:	*Windows* Portable Executable files.
Payload:	None.
Removal:	Delete infected files and restore them from backup.

FEATURE 1

PEERBOT: CATCH ME IF YOU CAN

Elia Florio, Mircea Ciubotariu
Symantec Security Response, Ireland

When Petar Maymounkov and David Mazières designed the *Kademlia* protocol [1], they probably didn't imagine that one day it would be used to ensure the livelihood of new generation botnets. Nowadays, botnets are in a state of constant evolution and have progressed in complexity. Just three years ago, the term 'botnet' referred generically to a collection of IRC trojans; today the term could be used merely to describe the sophistication of modern networks of malicious bots.

MALWARE AND PEER-TO-PEER

Research has shown that botnet development is currently proceeding in two different areas. One area of development involves the design of new bot functionalities. Malware writers continue to add new code to their bots to make them faster in propagation and invisible on the system. While older bots were created to perform DDoS attacks, the new generation of bots can also send image spam, gather email addresses, make search queries on *Google*, log keystrokes, steal passwords and upgrade their components.

The other area of botnet development is in the design of new command and control (C&C) strategies, which is a game played at network level. A bot without control is useless, and controllers are looking for more intelligent strategies than standard IRC in order to administer their creatures without being caught. Decentralized and distributed networks, such as peer-to-peer (P2P) networks, are perfect for this purpose.

In 2006, W32.Nugache@mm represented the first concrete effort to build a malicious P2P network over TCP port 8. However, Nugache was designed with a minor flaw: the list of initial peers was hard-coded in the threat and limited to 22 servers, so it wasn't a real decentralized P2P network. But the idea was innovative, and researchers expected the next 'PeerBot' to appear soon afterwards.

In the first months of 2007 Trojan.Peacomm (a.k.a. the 'Storm trojan') confirmed the trend and showed how legitimate P2P protocols can be used effectively to coordinate virus networks.

THE 'STORM TROJAN' ATTACK

The new year's spam attack started on 18 January 2007 and was reiterated on 21 January and again later. Millions of emails were spammed to legitimate accounts with an executable attachment which turned out to be a trojan dubbed 'Trojan.Peacomm'. It was also referred to as the 'Storm trojan' due to the fact that some of the subject lines of the emails included news of severe storms that had hit Europe during January.

A previous attack, which occurred in the final weeks of 2006, had also triggered anti-virus radars due to an elevated level of spam. This was W32.Mixor.Q@mm, and the outbreak was effective because the threat was spammed using 'postcard.exe' and similar file names.

Many similarities between the Mixor and Peacomm outbreaks led anti-virus researchers to believe that the same group was behind the two incidents. Both of the attacks were, in fact, efforts to build a wide and distributed network of compromised computers running different types of trojans.

The set of malicious files downloaded by Mixor in 2006 included spam and mail-proxy trojans. In the new year this initial set was enriched by new trojans including a DDoS module, a rootkit, and the peer-to-peer client.

THE POLYMORPHIC PACKER

Peacomm, as well as all the components related to it, makes use of an improved version of the infamous packer *Tibs*. Some AV engines already detect most of the executables packed with *Tibs*, purely because they include detection for the packer itself.

Drawing an analogy with polymorphic viruses, the equivalent of the polymorphic decryptor in this case is the unpacking code, while the viral body equivalent is the original malicious executable. The analogy is also valid for detection: in some complex polymorphic viruses, detection relies on recognizing the polymorphic decryptor generically; in the same way, detection of Peacomm is possible by detecting the packer pattern.

Tibs by itself is nothing more than a regular packer, although its authors have put a great deal of effort into keeping it undetectable by AV engines. Basically, the packer adds a new section to the executable and replaces a few bytes from the entry point with a polymorphic decryptor that will restore the original executable code and data and pass control to its entry point.

In its early days *Tibs* used to encrypt the original executable using a simple function that was very easily bypassed using basic cryptanalysis. Subsequently, its authors decided to change the encryption algorithm to Tiny Encryption Algorithm (TEA), which gives much better protection against cryptanalysis. For further protection, older variants made heavy use of MMX and FPU instructions, in an attempt to break the emulators, since these kinds of instructions are used only in specialized applications. Recent variants make use of 'exotic' APIs such as User32!DdeQueryConvInfo, in order to trick emulators and virtual machines – which tend to stop emulation when encountering such unsupported APIs.

New, different executables are spread with the same functionality every once in a while, in the hope that there won't be a signature-based definition recognizing the newly created files. Even though these executables look different every time, the original packed code and data and their functionality do not change unless there are changes in the source code. By using this technique, the authors ensure a pretty good chance of evading detection that relies on specific signature recognition, while the cost of establishing and maintaining such a system is minimal (i.e. given the packer, a small script could do the job in no time). The files can be refreshed as often as every download, but it has been noted that timed intervals are preferred (for instance, the executables may be repacked every hour).

MIXOR FAMILY: A STRANGE FILE INFECTOR

Unfortunately, the source code for the Mixor virus has been widely available on the web since 2006. In fact, it is known as an 'open source' virus. It was released in March 2006 as 'X-Worm', a proof-of-concept virus for an underground magazine.

Mixor is a polymorphic file-infector virus with mass-mailing capabilities. The virus is also designed to carry a secondary executable file as payload, so it is the perfect threat for integration with any external trojan or backdoor code.

The original source code of the X-Worm was modified to create new versions of the virus, which initially incorporated Trojan.Galapoper.A and later Trojan.Peacomm. In addition, the original X-Worm added a copy of itself to .rar archives, but this part of the code was removed from the variants that have been seen in the wild.

One feature that makes the Mixor virus unique is its unusual infection strategy. Traditional file infectors append the viral body at the end of the host and patch the entry-point in order to run the malicious code first. In contrast, Mixor does not append itself to the end of the file while infecting. Instead, it creates a copy of itself in the same folder but with a random file name. Next, Mixor patches the entry-point of the host program by inserting a little shell-code that will run the external copy of the virus. As a result, there will be a secondary file, which is a copy of the pure virus body, for each infected file. Figure 1 shows the differences in the infection techniques.

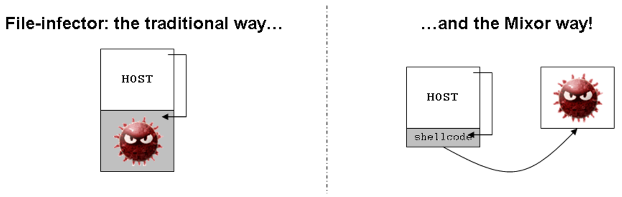


Figure 1: The file infection strategies of traditional file infectors and Mixor.

PEACOMM AND THE P2P NETWORK

Peacomm uses a kernel mode payload injector. The trojan drops the driver *wincom32.sys* and runs it as a system service. This driver injects a hidden module from the kernel mode into the user mode space of the SERVICES.EXE process via *KeAttachProcess* and *KeInsertQueueApc*. The injected executable is the component responsible for all the P2P communications and starts several threads in the SERVICES.EXE process.

The Peacomm driver was also upgraded by the authors with a full set of rootkit functionalities. In fact, the variants released after 21 January were able to hide files, registry keys and active network connections. The rootkit uses Service Descriptor Table (SDT) hooking to hide files or keys, and hijacks IRP_MJ_DEVICE_CONTROL of '\Device\Tcp' to hide active connections of SERVICES.EXE.

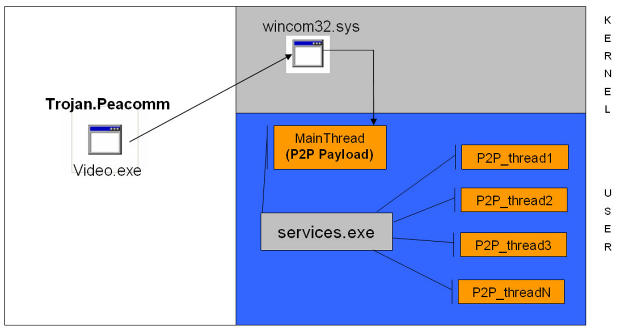


Figure 2: Module injection from kernel mode.

The first Peacomm variant was configured to communicate over UDP port 4000, but peaks reported by network probes in the days immediately after the attack indicated that later variants also used ports 7871 and 11271 (see Figure 3).

A computer infected by Peacomm sends and receives a large number of UDP packets starting with 0xE3 (227) bytes. It uses a well-known protocol in the P2P community called Overnet [2] (an implementation of Kademlia theories). The trojan creates a configuration .ini file with the

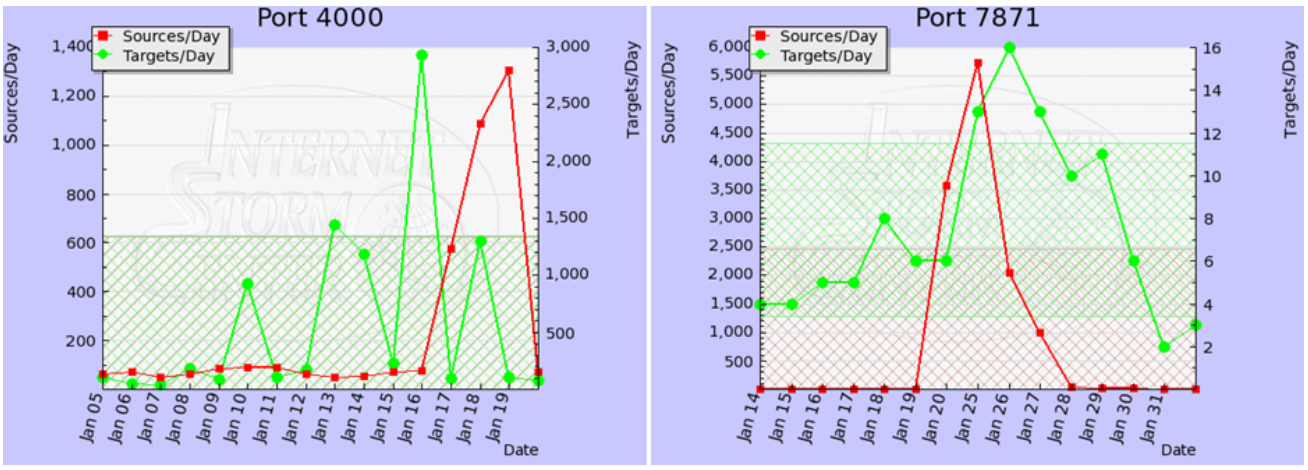


Figure 3: Spikes reported on UDP ports 4000 and 7871.

list of P2P hosts used as 'first point of contact'. The peers list is variable and contains hundreds of encoded entries. Upon investigation, the encoded entries proved to be legitimate hosts running Overnet or mlDonkey clients. So part of the peer-to-peer botnet is made up of legitimate peers, which (unknowingly) support and propagate the malicious P2P traffic generated by Peacomm.

C&C OVER OVERNET

Just as the W32.Spybot family of backdoors build up a botnet by making use of the IRC communication channels to retrieve their commands, Peacomm uses its own P2P network to retrieve information relating to which files to download and execute.

The *counter* section in peers.ini and in wincom32.ini for the most recent variants, denotes the current state of downloads from the network. The initial setting *counter=0* specifies that no files have been downloaded yet. Using the

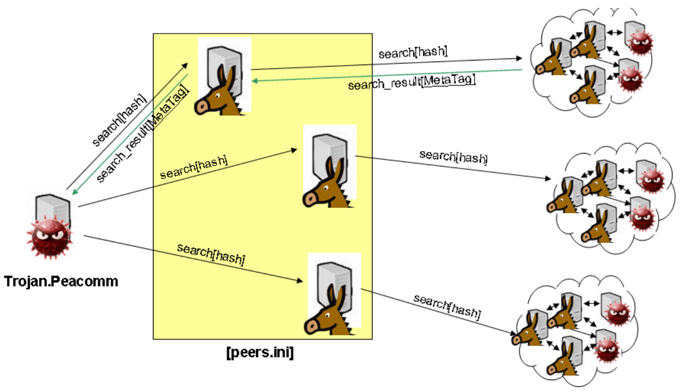


Figure 4: Search over the P2P network.

actual *counter* value Peacomm computes an encrypted 16-byte search string or hash, which additionally contains checksum information that validates the search and time information, and a random part that makes the search string look different.

The search command for this hash uses a custom search type (0x14) and expects to retrieve an encrypted tag string as seen in the following capture:

```
User Datagram Protocol, Src Port: 4665 (4665), Dst
Port: 7871 (7871)

Protocol: eDonkey (0xe3)

Message Type: Search Result (0x11)

Hash: B225564021F1B55C35FB8DA9950A6678 (search hash)

Hash: 05B3D57C0C90A3010000000000000000

Meta Tag List Size: 1

eDonkey Meta Tag

  Meta Tag Type: 0x02 (TAGTYPE_STRING)

  Meta Tag Name Size: 2

  Meta Tag Name: id

  String Length: 86

  String: 6%m[f7/$'$1vo$e:9)n"!mq2[\,;jc+!2zk*g5&<
p$1cdvn"(0c="a4;xd^j'v)!,[_,'^[%"%184qh88dj'!!
```

The original string is encrypted with RSA [3] on 64-bit blocks. It is then encoded using a custom base64 algorithm that has an additional layer of light encryption which, in fact, only changes the table of translation for the base64 encoding.

In order to decrypt the string, one would need the private key pair (d, n). By analysing the trojan's code one could easily identify the d component of the key, as it is hard-coded in plain text in the executable's data section. However, there is no sign of the modulus n.

Further analysis revealed that *n* is actually taken from the search result packet. More specifically, it is the second hash value, immediately after the search hash.

So far it has been observed that the private key (*d*, *n*) has been constant for different releases of Peacomm variants and it has the following value: (0x025F2D1619EF1ABD, 0x01A3900C7CD5B305).

Using the information above we can decrypt the given string and get the following structure:

```
Xor = 0x3B
Add = 0xAD
Counter = 0x5300
String = "205.209.179.112/game0.exe"
```

'Xor' and 'Add' are byte control check sums computed with the two named operations on the rest of the data. 'Counter' is a word represented in the network order that holds the next value for the *counter* section in the .ini file, so that the threat should know what to search for next. 'String' is the URL location of the file to be downloaded and executed.

The advantages of Peacomm's network have a great impact on the prevention of this type of attack as well as tracking the origins of the infection.

The following are a few of the properties of this network:

- It is very difficult to identify the malicious peers.
- Malicious traffic is similar to legitimate P2P traffic.
- It is serverless – even if a large number of the peers become unavailable, the network is still available.
- It is flexible – it can be easily extended with new commands and may be configured to use any port.

SERVED PURPOSE

Peacomm will attempt to download and execute a variety of custom threats on the compromised computer with one ultimate purpose: to facilitate sending spam emails or instant messages. The messages are used both for advertisements and for enlarging the bot network by sending malicious applications or links.

The following are a few of the current downloads:

- The first component being installed seems invariably to be an updated variant of Trojan.Abwiz.F, which hides its presence using rootkit techniques and, together with another installed component, acts as an SMTP server in order to relay spam and send spam through the compromised computer. In doing so, it connects to a predefined location that contains lists of email addresses as well as the message to send.
- The next component that is installed will harvest recursively the email addresses found in certain file types and send them to a predefined location, where they are added to a huge database that is used for spamming.
- Another deployed component is W32.Mixor.Q@mm, which comes bundled with the most recent variant of Trojan.Peacomm. The purpose of Mixor is to infect new computers by sending infected emails containing a copy of itself. As described earlier, Mixor also infects executables, thus making it harder to remove the threat.
- The Trojan.Mespam component is also used to expand the bot network. It will send spam IM messages, retrieved from a configuration server, that may contain malicious URLs through some of the widely used IM clients. It does not require passwords to do so, since it registers itself in the Layered Service Provider (LSP) chain of the network interface and detects the IM connections. This way, the malicious IM messages look legitimate since they come from a known contact and therefore have a good chance of tricking the target client.
- Last, but not least, a component whose purpose was only recently discovered is installed to allow directed DDoS attacks against custom IP addresses by submitting bursts of packets to them. The addresses to be attacked are retrieved from a predefined configuration server.

CONCLUSION

The Peacomm trojan represents just one element in a vast scheme designed for making money. And there is a lot of money involved, since the quick release of updates and new components must be sustained by an impressive level of resources.

Peacomm also highlights the current trends in malware evolution, which seem increasingly to be profit-oriented. Whether it is achieved through sending spam or stealing personal information, we notice an increasing and concerning growth of cyber crime.

REFERENCES

[1] Kademlia: A Peer-to-peer Information System Based on the XOR Metric. http://www.scs.cs.nyu.edu/~petar/kpos_iptps.pdf.

[2] The Overnet Protocol. https://opensvn.csie.org/mlnet/trunk/docs/overnet.txt.

[3] RSA. http://en.wikipedia.org/wiki/RSA.

FEATURE 2

REAL-WORLD TESTING OF EMAIL ANTI-VIRUS SOLUTIONS

Dr Adam J. O'Donnell
Cloudmark, Inc., USA

Testing security products can be a very complex task – especially validating the effectiveness of technology against threats that are either difficult to enumerate, or which evolve at an extremely rapid rate.

If you are testing a source code flaw-finder that examines code for specific classes of programming flaws, you can create a test set with a large number of example errors and then refine the code until all the flaws are detected and none of the correct segments are caught as false positives. Likewise, if you are writing a vulnerability scanner that purports to detect a list of known security and configuration holes, you can construct a pool of example systems where the issues can be found and confirm that the scanner detects the complete list.

In many situations, however, test vectors that are representative of the threat environment cannot be created, making the validation of a security technology somewhat challenging. What happens when the product you are testing is designed to catch threats that are evolving rapidly? Building a corpus of known threats for testing against a live system is futile if the time between creation and testing is long enough for the threat to evolve significantly. Either the live system must be tested against live data, or the system, which most likely has been fetching updates, must be 'rolled back' to its state at the time each element in the test corpora first appeared.

ANTI-SPAM TESTING METHODOLOGIES

Consider anti-spam systems, for example. The most essential ingredient for an accurate test of an anti-spam product is a stream of *live* email, rather than a stale, pre-screened corpus. If spam did not evolve at a high rate, then corpus-based testing of an anti-spam product would provide catch-rate figures that were commensurate with those seen when the filter is put into production at the mail gateway. The rate of change of spam content is so dramatic, however, that accuracy figures provided by corpus testing become misleading as the corpus ages by the hour.

The 'accuracy drift' of the anti-spam system under test would be insignificant if not for the fact that spam evolves at such an incredibly fast rate. If spam did not mutate so quickly, then Bayesian filters and sender blacklists would have been the final solution for the entire messaging abuse problem.

In the past year, *Cloudmark* has seen more and more customers realize that a live data stream is essential for evaluating a new anti-spam solution even before we begin our initial engagement. It is our experience that the differences in the expected performance, as derived from corpus testing, and the performance realized once the filter is put into production, are driving customers independently to adopt more stringent testing methodologies.

GENERAL SECURITY PRODUCT TESTING

I began this article intending to discuss security products, and thus far I have only mentioned anti-spam systems. The issues with testing became apparent in this area first because of the number of eyes watching the performance of anti-spam systems: almost every email user on the planet.

What about other filtration methods? In the past, anti-virus systems had to contend with several hundred new viruses a year. A set of viruses could easily be created that would be fairly representative of what a typical user would face for many days or weeks, as long as the rates of emergence and propagation of new viruses were low enough.

The assumption that a regularly generated virus corpus could be representative of the current threat state was mostly accurate in the days when amateurs created viruses with no motive other than fame. However, contemporary viruses are not written by 'script kiddies' trying to outdo each other, but by organized professionals attempting to build large networks of compromised desktops with the intention of leasing them as automated fraud platforms for profit.

The profit motive drives a much higher rate of malware production than previously seen, as exemplified by the volume of Stration/Warezov and CME-711/Storm variants which caused difficulties for many of the AV companies that attempted to catch the outbreaks.

IMPLICATIONS

What's the big deal if people don't perform testing correctly, and what does this have to do with new virus outbreaks? Engineers typically design and build systems to meet a specification, and they place their system under test to verify that this specification is being met. If their testing methodology is flawed, then they cannot detect design flaws that are likely to exist in the product. Eventually, these flaws will emerge in the public eye and, in the case of AV products, consumers will start to realize that products with claims of 100% accuracy have been allowing viruses through.

In other words, by testing against even a slightly stale corpus, and not against new variants, AV filter designs are able to claim considerably higher accuracy than their products actually provide. This is not to say that on-demand testing is completely neglected; rather it is relegated to a secondary role behind the de-facto industry standard of corpus testing.

I am by no means the first person to discuss the testing of anti-virus products. The subject received quite a bit of attention when *Consumer Reports* attempted to test anti-virus systems using a set of newly created viruses rather than a standard corpus. While their attempt at devising a new testing methodology may have been well intentioned for the testing of the heuristic components of scanners, it was not representative of how threats appear on the Internet. Using new, non-propagating viruses to test an AV system is almost equivalent to the proverbial tree that falls in a forest that no one is around to hear.

Additionally, the incremental changes that are usually detected by heuristics are not often characteristics of the viruses that become significant threats – it is the radical evolution in viruses and the time required for the anti-virus vendors to react that are of more concern to us. These are things that cannot be modelled via corpus testing, but only via extended testing on live traffic.

LIVE TESTING

We should ask why testing is not done primarily on live data as opposed to corpus-based analysis. I suspect there are two reasons: labour and repeatability.

With corpus testing, the tester hand-verifies that each element in the corpus is a virus. This is done once, and that cost is amortized over every test conducted using the corpus. This isn't a realistic option with live testing, since every message that is either blocked or passed by the filter must be examined by hand. Collecting repeatable test results is also challenging because, to be meaningful, the test must be conducted over an extended period of time to cover multiple, large and unique virus outbreaks. However, just because something is difficult, does not mean it shouldn't be done.

TOWARDS ACCURACY METRICS

In situations where there are a limited number of security vendors and adversaries, even live testing becomes extremely difficult. Consider the following hypothetical situation, where there is only one security vendor and multiple adversaries. Every client system is identical and running current anti-virus packages.

From the standpoint of the testing and user community, the accuracy of the system is perfect; no viruses are seen by the system since they don't even have an opportunity to propagate. At the same time, virus writers realize there is a huge, untapped source of machines just waiting to be compromised, if they can just gain a foothold. These individuals sit around and hack code until a vulnerability is found in the AV system, and upon finding it, release a virus that exploits it in the wild.

Before the attackers uncovered the hole in the AV engine, the system could be viewed as being 100% accurate, since no viruses propagated.

After the virus is released, havoc breaks out as 5% of all computers worldwide are infected before the vendor releases a patch. If the vendor was able to move faster, the number of compromised systems may have been only 1%; left to its own devices, with no patches applied, the virus would have compromised every system connected to the net. In this situation, the accuracy of the system is even more difficult to quantify.

Consider the three following accuracy measures:

1. Accuracy = 0%. No viruses were in circulation at the time except for the malware from the recent outbreak, on which the scanner had zero accuracy.
2. $\left(\frac{\text{Virus Corpus Size - 1}}{\text{Virus Corpus Size}}\right) * 100\%$.
   Several viruses were in circulation at the time. Detection accuracy was perfect on all viruses except for the latest outbreak.
3. Pr (a given system is not infected) *100%. The probability that any given system was not infected by the contagion.

The third of these measures seems the most appropriate, and the most flexible, given a variety of network and economic conditions and adversary styles. Anti-spam system evaluators use the measure, which is effectively the expectation of exploitation for a given host. It is a slightly more sophisticated way of expressing the probability that a piece of spam will get through.

RESPONSE TIME AND ZERO-DAY AV

From a general security standpoint, however, this measure covers a difficult and often ignored parameter that is critical to the accuracy of a security product: vendor response time. If the window of vulnerability between when the virus first appears and when signatures are issued is reduced, the accuracy expressed by this metric improves.

The Zero-Day Anti-Virus (ZDAV) industry is an emerging sub-industry that attempts to address this issue directly by shortening the time between outbreak and AV coverage by using fingerprints that are generated and issued automatically.

Although the technology cannot be used for infection remediation, ZDAV's utility for keeping a message stream clean of emergent viruses before desktop AV systems are capable of catching the content makes it an incredibly effective means of reducing the number of infected systems in the wild.

While many methods of providing so-called zero-day coverage exist, they all revolve around removing a traditionally critical component from the fingerprint-generation loop, namely the small team of highly trained malware analysts.

For example, *Cloudmark* correlates reports from a large pool of both ordinary and trusted honeypots and human respondents, and allows a decision on the disposition of the new sample content in a handful of minutes.

Both honeypot and human submitters who provided a report that agrees with the overall community's assessment gain the system's trust, which is used by the system to (1) issue fingerprints originating from those reporters more quickly in the future and (2) remove reporters who submit bogus content.

Many of these zero-day technologies are being used primarily in the message stream, but this restriction probably won't last for long. The technology appeared for messaging first because of the high rate of emergence of email virus variants, as well as the ease with which service providers – who ultimately fund these technologies – can quantify its cost. Mail is a store-and-forward technology that provides managers an opportunity to examine the number of viruses, unlike web-based trojans that fly through alongside legitimate traffic and don't provide much opportunity for analysis.

CONCLUSION

As consumers begin to demand performance estimates that match their real-world experience, technologies similar to the zero-day methodologies described will appear in areas outside of the message stream. Testing methodologies for anti-virus products must become much more rigorous and focused upon real-world scenarios such as live-stream testing, rather than a second-tier test metric compared to corpus testing.

LETTER

THE STRANGE CASE OF JULIE AMERO

In January I heard that a 40-year-old female substitute teacher, had been convicted in a US court on four counts of risking injury to minors.

The prosecution argued that Julie Amero had been surfing porn while in charge of a class of seventh-grade pupils. Julie maintained that the pornographic material kept appearing on the screen, and that whenever she tried to get rid of it, more would appear.

Reading about the case, it did not seem believable that a 40-year-old pregnant woman would spend the whole day surfing porn in front of a class of 12-year-olds. To me, the symptoms Julie described were the classic signs of a 'pornado' attack, so I started digging further.

I found many things that worried me.

The school's investigation consisted of checking the browser history and firewall logs, and then firing her. There was no chance for Julie to tell her side of the story, and no help was given to her.

The police investigation consisted of running a program to see which sites were visited. There was no search for malware or spyware, no examination of the pages visited for Javascript, no attempt to piece together the sequence of events, and no analysis of the firewall logs to discern browsing patterns.

In court, several incorrect technical arguments were made by the prosecution, including the assertion that if a link is a different colour, then it must have been clicked on deliberately.

Some of the other arguments were highly technical, yet the process continues at such a speed that there is no time during the trial to pick up on inconsistencies.

In short, as things currently stand, anyone who has pornographic images appearing on their screen as a result of malware on their PC, or being trapped in a continuous pop-up loop, is at risk of conviction if there are minors in the vicinity.

Hopefully a lot of lessons will be learned from this case, and hopefully they will be learned in time to help Julie.

The next significant date in this case is sentencing, which is currently scheduled for 2 March – by the time you read this it may already have happened.

Alex Shipp
MessageLabs, UK

[More information and updates on Julie Amero's case can be found at http://julieamer.blogspot.com/ - Ed.]

PRODUCT REVIEW

AEC TRUSTPORT WORKSTATION

John Hawes

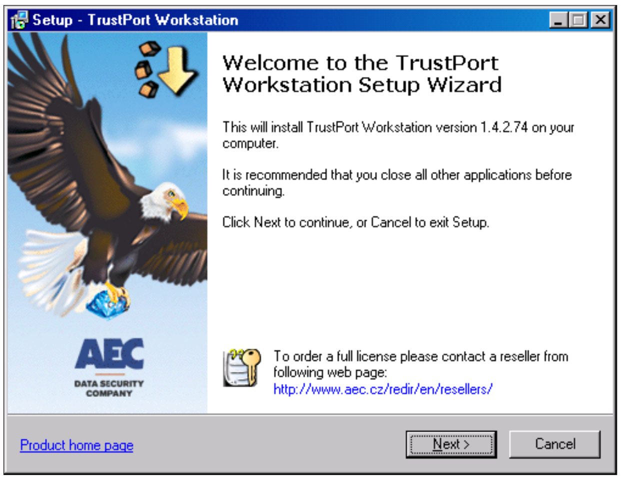
One of my first duties, on taking over the running of the VB100 testing program, was to liaise with vendors regarding the last round of tests carried out by my predecessor. One such vendor was *AEC*, whose *TrustPort* product had narrowly missed achieving VB100 certification. The product's developers were understandably somewhat bemused by this, as two other products whose engines are used by *TrustPort* had scored significantly better results. Some analysis revealed that much of the problem was due to certain file types not being scanned by default. *TrustPort* was submitted for testing again a few months later, and this time suffered no such difficulties, with splendid detection rates across the test sets.

After looking at the product in some detail during this period, I was intrigued by the concept of rolling multiple detection engines into one product. How, I wondered, did the improved detection balance with the additional overhead?

When I learnt that *TrustPort*'s anti-virus functionality was not the entire offering – and was, in fact, generally only available as part of a larger product with a range of diverse and unusual components – I resolved to get hold of a copy for deeper investigation.

COMPANY, PRODUCTS AND WEB PRESENCE

AEC, founded in 1991 and headquartered in the Czech Republic city of Brno, describes itself as a 'Data Security Company'. The company started out as a reseller of anti-virus software, and has added further security offerings to its portfolio over the years.


Setup - TrustPort Workstation

Welcome to the TrustPort Workstation Setup Wizard

This will install TrustPort Workstation version 1.4.2.74 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

To order a full license please contact a reseller from following web page:
http://www.aec.cz/redir/en/resellers/

AEC began developing its own anti-virus product in 2003, and the *TrustPort Workstation* offering is the desktop, end-user part of the *Phoenix Rebel* suite of security tools. The suite bears the bombastic tagline 'the ultimate security solution' and a logo theme which appears to show a bald eagle carrying a blue jewel. It includes anti-virus and firewall for servers, gateway products featuring anti-spam, mail and web malware filtering, web access control, and management tools for the control and deployment of these various items.

The company also offers a range of security services, including assessment and penetration testing of corporate networks, managed services, and general security consultancy.

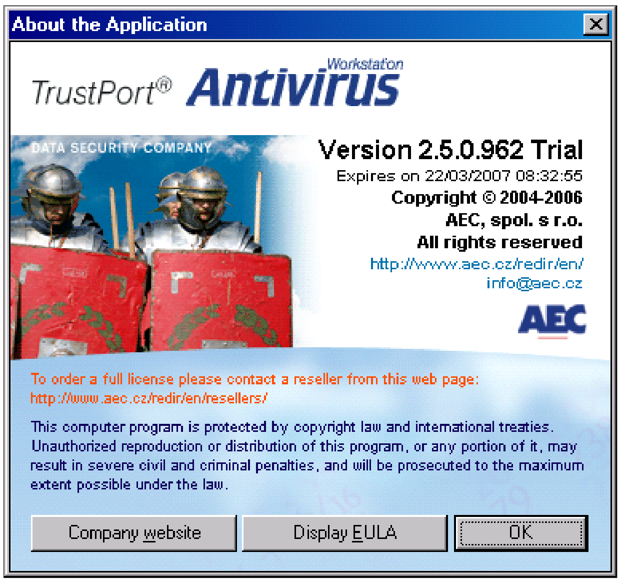
Beyond the anti-virus, which is the main focus of this review, the workstation suite contains some rather unusual items. The client firewall is fast becoming a standard part of such desktop security products, and spyware and adware are covered by some of the several engines combined to provide broad malware protection. Desktop web and email filtering (for both malware and spam) are also rolled in as part of the 'anti-virus' module. *TrustPort*'s most interesting aspect is the addition of data encryption, file encryption and signing, and secure shredding, thus placing another layer of security between hackers and sensitive data.

AEC's website can be found at http://www.aec.cz/ – there are Czech, Slovak and English versions of the site. The English version of the site is a fairly simple, pared-down affair; the home page features links to details of the various product offerings, and some items of company news. Elsewhere on the site, apart from further product details, are: a list of sales partners (most of which are based in eastern Europe, but there are also some in the Middle East and Pacific Rim regions), contact details, pages for upcoming events, and FAQs, the latter two of which were pretty blank and 'undergoing construction' when I visited (although there were some entries in the Czech version of the FAQ page).

The complex business of providing information on malware, outbreak alerts and so on is mostly left to the providers of the individual engines used by the product (which are: *BitDefender*, *Ewido*, *Grisoft* and *Norman*). A support email address is provided on the contacts page (a test query elicited a terse, but accurate response within 12 hours – support for all the products is apparently included with the licence). Trial versions of several products are available to download from the website, following a brief personal details page.

INSTALLATION AND COMPONENTS

TrustPort Workstation is available from the download site in a single 65MB file. The installer allows for a selection of



Version 2.5.0.962 Trial
Expires on 22/03/2007 08:32:55
Copyright © 2004-2006
AEC, spol. s r.o.
All rights reserved
http://www.aec.cz/redir/en/
info@aec.cz

To order a full license please contact a reseller from this web page:
http://www.aec.cz/redir/en/resellers/

This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

the modular components to be specified, but defaults to the full set. Installing these in various configurations started simply, with a EULA and selection of components, along with install locations. There followed a selection of options for the *eSign* module, a list of secure identity management systems for which support is available, including *Spyrus*, *Eutron* and *Aladdin*'s *eToken*.

After that came some setup for the encryption software, the choice of whether to create a new virtual disk or mount an existing one, and then the extraction and installation of files – a fairly rapid process.

Applying updates to the anti-virus portion of the software was the next step, and one which somewhat worryingly reported successful completion on my lab machine, despite there being no possibility that the software had connected to its update source. Thinking this could have been due to some setting of the webserver to which all requests in the lab are redirected, I tried again on an entirely unconnected machine, but achieved the same result.

When run later on with a genuine web connection present, updating was much more successful, although a rather lengthy business (which is not surprising, given that four separate engines are used).

Once the product had made contact with its home base in the Czech Republic, it informed me at once that 16MB of update was required, and this chugged slowly onto my machine over the next 45 minutes or so. Each product, it emerged, had its own progress bar, so just as I thought the update was almost complete it would revert to zero and begin on the next set of definitions, engine upgrades and so on. Although I didn't watch this process too closely, I couldn't help but notice that one engine (the one provided by *BitDefender*) reached an impressive 111% before it was completely happy. Another update, a few days later, comprised around 8MB (7MB of which were *Norman* components) and took only 10 minutes to complete, much of which was in the installation rather than downloading of data.

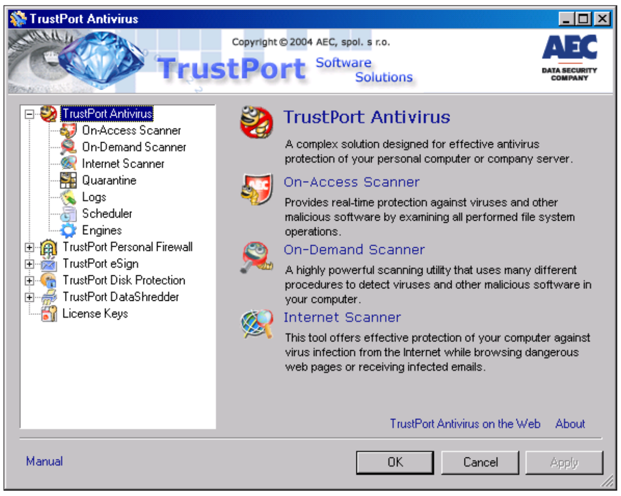
The final stage of the installation process was the creation of an encrypted 'volume', with size limitations and encryption options, including whether to use the *AES* (*Rijndael*) or *CAST-128* cipher algorithm. A username/password combo was taken for access to the stashed data (without the usual advice on password length or complexity), and a random seed generated. The volume was then created in a matter of a few seconds, and the setup was complete.

CONFIGURATION AND OPERATION

Once installation was complete, the reboot that generally follows such a process was not requested. Impressed, I looked around for the anti-virus tools and signs of on-access protection. No new items were in evidence in the system tray, and a quick run of a file-opener utility showed no slowdown in file access, indicating almost certainly that no scanning was taking place.

On searching the programs menu, I found the *TrustPort* entries, one per module, and in the anti-virus section chose to 'Open TrustPort Tray'. This added a rather attractive blue gem to the tray, which led to a raft of options for the various components, including a common configuration dialogue. It took a matter of a few clicks to discover that the on-access scanner was in a 'stopped' state, with no clear way of starting it.

Rebooting the machine cured this, another (shield-shaped) icon was added to the tray and protection was fully in place.



Copyright © 2004 AEC, spol. s r.o.

TrustPort Antivirus

A complex solution designed for effective antivirus protection of your personal computer or company server.

On-Access Scanner

Provides real-time protection against viruses and other malicious software by examining all performed file system operations.

On-Demand Scanner

A highly powerful scanning utility that uses many different procedures to detect viruses and other malicious software in your computer.

Internet Scanner

This tool offers effective protection of your computer against virus infection from the Internet while browsing dangerous web pages or receiving infected emails.

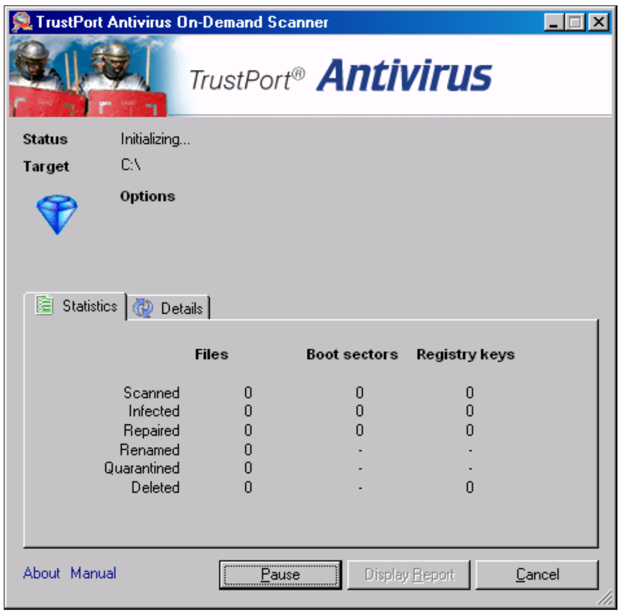
I later discovered that this issue appeared only to affect *Windows 2000* installations, and there were no such problems on *XP* or *Vista*.

The other components, the shredder and data protection utilities, seemed to be functioning perfectly well on all platforms without reboot, although *Vista* support is not yet fully rolled out for all the modules. I had a brief look through these prior to running any serious testing, and was again impressed by the range. The firewall offered an extra stash of 'Network utilities', and while both the shredder and disk encryption modules seemed fairly simple to operate, the *eSign* component was less intuitive. It offered a fair few options but it wasn't immediately obvious how it was supposed to be used, so I resolved to return to it once the main bulk of my testing was complete.

The first thing that struck me when looking into running a few scans over the *VB* test collections in the lab was that there seems to be no main GUI in the usual sense. Of the two icons dropped into my system tray on install, a shield-shaped one – quite logically used for operating the shields – allowed me only to disable/re-enable on-access and web scanning (these actions were also available to a user without admin rights under *Vista*). The second icon, a return of the blue gem, led to a tree of options for each product, appropriately shrunk down if only certain modules were installed.

Each section's options included a 'configure' link, which led in turn to a central configuration interface with pages for each module installed. From here settings could be adjusted, with options laid out clearly and logically, allowing a broad range of tweaks and changes without swamping the user with overly technical tables of checkboxes. I noted that this configuration interface was rather slow to open, taking more than 30 seconds on some older systems, but once up it was fairly responsive. The interface was divided into sections, with an opening page describing each and detailing the available options. Somewhat oddly, these descriptions did not link to the sections they covered, which could only be accessed from the tree menu to the left.

Most of the AV part of the GUI made pretty admirable sense – on-demand scans operate by default in full paranoia mode, scanning all files using all engines, including the heuristic and sandbox functionality provided by some. Some options, which I would have assumed would form integral parts of a full anti-virus product's GUI, were notable by their absence. While there were controls of all the options here, including logging and scheduling of updates, there was no option to schedule regular scans; the 'scheduler' page only offered controls for updating. I assumed the controls for scheduling, like immediate scanning, must be set up elsewhere.



Status Initializing...
Target C:\
Options

	Files	Boot sectors	Registry keys
Scanned	0	0	0
Infected	0	0	0
Repaired	0	0	0
Renamed	0	-	-
Quarantined	0	-	-
Deleted	0	-	0

For immediate scanning perhaps this is understandable, as it is something required more often by testers than the average person, but there was room for it here and it seemed sensible to have access to a scan while I was playing with the controls for it.

Logging also seemed to be lacking something – although the details of how logs were kept were controlled here, there was no 'Show me the log now' button, or even any evidence of where I could find it for myself. Looking later at the firewall section, I found the log for that was easily viewed, and an 'external file viewer' was also provided, but that wouldn't show me the scanning logs, complaining about the format despite it being plain text. A simple button or window on the AV logging screen would make a nice addition.

With the software fully set up, and the controls reasonably well mastered, it was time to run a few scans.

MALWARE SCANNING AND DETECTION

With the settings thus examined, but not changed, I moved back to the system tray icon to find the 'Scan selected area' button, and ran it over the test sets. Scans could also be initiated, I found, from the right-click menu in *Explorer*, which simplified my task somewhat, but scans using both methods took some time to get going (up to a minute on slower systems). I suppose that with such in-depth processing using multiple engines, few users would expect an instant response, and on-demand scanning would in other products mostly be used in a scheduled, once-a-week-in-the-middle-of-the-night kind of way; though I could find no scheduler here either.

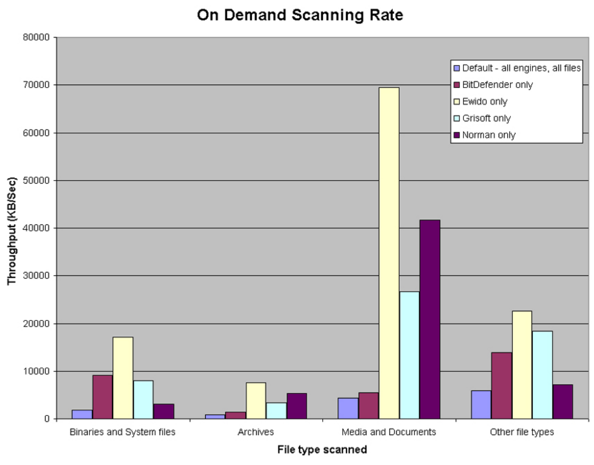
Logging, which is viewable from the scan results window if not from the configuration GUI, is presented in a clear and readable XML layout, and also stored as plain text, both of which formats came in handy during testing.

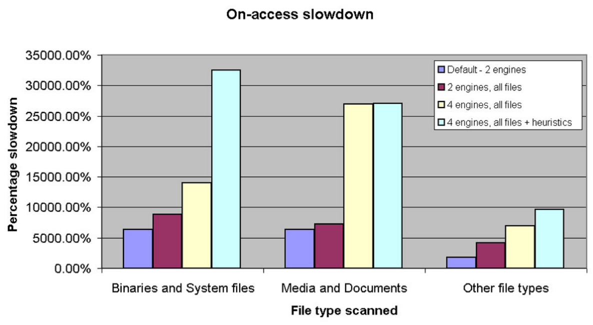
The scans of infected test sets showed an impressively thorough detection rate, with the product having no problems covering the collections. Although I would have expected each of the engines to miss a handful of files from the zoo collections, in this setup they overlapped nicely to cover each other's shortcomings and ensure full marks for the product as a whole.

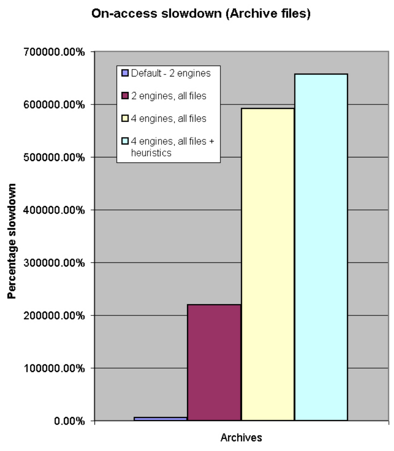
Even in a non-updated state, nothing was missed in the sets used for the last VB100 review (perhaps unsurprisingly, as the shipped version included data from early January, just prior to the deadline for the comparative review). The addition of files from the latest WildLists and other more recent samples caused no problems either; all the newer samples were also detected, with several picked up either generically, as members of known families, or using the deep and sophisticated heuristics available – it was notable that a number of these were not spotted by the on-access scanner.

On-access scanning is designed with a lower overhead in mind, and by default only the *Grisoft* and *Norman* engines are active, although the others can easily be activated from the 'engines' tab of the config dialogue. Further on-access scanning options, such as the scanning of compressed files and use of heuristics, are also easily enabled.

The use of multiple engines is, of course, certain to add to the system overhead. An on-demand scan using the full paranoia settings is likely to be fairly slow, but sure. On access, however, there is a more significant impact on one's user experience, and I was intrigued to see what kind of impact the product would make on a system.







Running speed tests identical to those included in the last VB100 comparative review, I produced results for on-demand scanning, using the default full settings and full set of engines, and then for the separate engines running alone – the results for *Norman* and *Grisoft* are particularly interesting, as directly comparable data from the *Vista* test is available (see *VB*, February 2007, p.14).

I also measured on-access slowdown, in several configurations – the default, with two engines scanning by file type; the same with all files being scanned; then all four engines, scanning all files, but without heuristic/sandbox detection enabled; and finally equivalent settings to the on-demand scan, with all detection powers up to the max. The results are shown in the graphs above.

I was pleasantly surprised to observe that, despite the quadruple engines, scanning speeds were by no means sluggish, and on-access overhead, except in the most intensive modes, compared pretty favourably with many of the products tested in the same way last month.

With all engines running, especially over archives, things did get fairly slow, but on demand with this setting a very respectable set of times were recorded. Indeed, figures for *Grisoft* and *Norman* were in some cases better than those recorded by the engine developers' own products, although this may have to do with vagaries of implementation – despite the archive settings being left in the default, with all types scanned to unlimited depth, *BitDefender* found a significantly higher number of files and, from a visual inspection at least, seemed to be recursing deeper into a wider range of compression types. The *Ewido* anti-spyware engine, run alone, seemed to have no archive-scanning features at all, accounting for some further anomalies in the speed charts.

Of course, another potential downside of having multiple engines to increase one's chance of catching malware is the similarly multiplied chance of a false positive. The *Norman* engine suffered such an event in the *Vista* comparative last month, and *Trustport* had the same result, detecting a trojan, somewhat ironically, in a piece of software provided by *BitDefender*. Consultation between *VB* and developers at *Norman* has since had this problem fixed, as a quick check with a fully updated version proved, and no further false positives were produced by the team of scanners.

OTHER FUNCTIONALITY

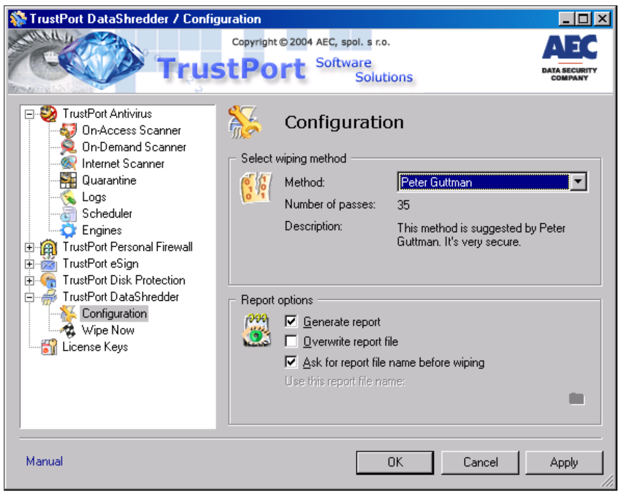
Many 'Internet Security' products make much of their multiple offerings, with anti-spyware, anti-spam, web malware filtering and firewalls rolled into a mighty product with spacious tabbed pages on the interface dedicated to controlling each of these aspects. Of these, *TrustPort* considers only the firewall to be a separate component. All the malware and spam filtering is combined, rather modestly, into the 'anti-virus' module, irrespective of the vector.

The firewall offers all the standard functionality of such things, with protocols and ports to block or allow traffic. Configuration seemed even more complex than is usual with such things, but a nice, simple slider was provided, with options from blocking all traffic to allowing everything, with allowing all outbound or applying the default rules in between.

Besides the controls for the firewall itself, this section had more surprise goodies; in addition to the standard controls over access ports and connections, a handful of extra tools were included. Alongside some statistics and details on traffic hitting the firewall, including data types and sources, are a selection of other 'network utilities', including simple interfaces to *ping* and *whois*, and a geographical lookup system.

These were fun little tools, and probably quite useful on occasion, for checking out a suspicious IP address recorded on the connections list. It would have been more usable had the individual items been more closely integrated – for example, the ability to right-click on an address listed as trying to connect to my machine, and perform lookup and geographical search operations to help figure out why, would have been more useful than needing to note down addresses and type them in afresh into a separate box.

The *DataShredder* utility is a simple tool, with a list of areas available for utter destruction. These include



Configuration

Select wiping method

Method: Peter Guttman

Number of passes: 35

Description: This method is suggested by Peter Guttman. It's very secure.

Report options

☑ Generate report

☐ Overwrite report file

☑ Ask for report file name before wiping

Use this report file name:

temporary files and folders, recycle bins, and histories, which could be shredded en masse or by individual area, and using a range of shredding methods. The default is the US Department of Defense approved method of running seven passes over the data. A quick version, with a single pass, is also possible, while for the most paranoid the Guttman method can be chosen. The description of this method reads: 'This method is suggested by Peter Guttman. It's very secure', and indeed it is, running no less than 35 passes over data to ensure absolute deletion.

I had a moment of confusion when running the program – having seen several options on the config screen relating to logging, I noticed that before starting its actions the product prompted for a log to write to, starting a browse window, for some reason, in the log folder of the anti-virus component. Browsing for a file with the required (.rpt) extension proved fruitless, and finally I realised that I had to create one for myself, with whatever name I fancied.

The shredding process now commenced, and using the DoD method I trashed all old data from an aging and very full machine (this took quite some time), and was quite astonished to see it rubbing out the traces of files I was sure I'd deleted at least two years previously. Considering the amount of data I've added to and removed from the drive since then, it seemed remarkable that any trace was still there – it has most definitely gone now. A context-menu option is also available on right-click within *Explorer*, to delete files securely at will.

The disk encryption system was similarly straightforward. A volume is created, with two types of ciphering available, and locked down with a username and password. This can then be mounted as a simple additional drive, and files stored in it in the normal way, and when unmounted this data is safe from all but the most expert and patient of information thieves. A number of such drives can be created, and transported as simple files from place to place; mounting can take place automatically on startup, and unmounting can be scheduled after a certain period of inactivity.

When I described the additional components of the *TrustPort* suite to a colleague prior to my testing, he suggested that it sounded like something that would be useful for spooks and spies. This component would certainly seem ideal for those FBI agents, and other representatives of law enforcement institutions, who seem so prone to leaving their confidential data-laden laptops in taxis, in the seat pockets of aircraft and on other forms of public transport; there is even the option to set a hotkey to unmount all volumes instantly, for use when that mysterious red dot begins tracking its way across the wall towards you.

Like the shredder, all controls were available from the central configuration tool; the only slightly frustrating aspect of the controls was that the 'Image editor' section browse button did not default initially to the directory where I had been prompted to deposit my initial image. This seems to be a minor issue across several of the products, with the browse start point often placed rather randomly.

The final component of the suite is *eSign*, an implementation of PKI for data encryption, signing and so on. Configuration options in the main GUI include options to add oneself, one's 'Supervizor', and a list of other recipients automatically to all encrypted files, and a selection of algorithms and methods for signing and encrypting your data.

Other functions are available via two further interfaces, a key management GUI and a wizard with options for signing and encrypting data; the second set of tools is available in *Explorer* context menus, to encrypt or sign selected files. Such a tool is certainly useful for secure communications and authentication of data and messages. Much of the functionality is available in the already-popular PGP/GPG system, one which does not rely on both users running specific proprietary software, and to my mind it would have made more sense to have used the OpenPGP standard, or at least to have provided the option to manage keys and encrypt data to this standard; however, I suppose that a closed system like this adds yet another layer of safety around one's data.

CONCLUSIONS

TrustPort's claims to be 'the ultimate security solution' are far from groundless. There's a lot here, far beyond the in-depth protection offered by four separate malware detection engines. These, of course, benefit not just from the spread of labs to ensure the latest items are identified and added to databases, but also using a range of sophisticated heuristic methods to catch some new items without specific definitions. While these themselves operate in a solid and efficient manner, producing some decent speed times for such a product, the addition of the extra functions provides several further layers of security.

I encountered a few minor problems in the course of my testing, mostly issues of usability, and many of them perhaps only really issues for someone in my rather unique position; few people are interested in running numerous local scans with minor changes to the settings. However, there is certainly room for a little more integration between some of the modules, and a central logging and log-viewing area would be a good place to start.

Of the more technical issues, the lack of a 'reboot required' method after install on some platforms, and the misleading 'Update successful' message, both stand out as particularly serious; both could lead a user into falsely believing they were safe from viruses – a dangerous situation to be in. The lack of integrated scheduled scanning is also a pretty significant failing; although a command-line interface is included to allow such scanning, a small addition to the interface would be a simple, but valuable improvement.

On the whole, however, I found *TrustPort* an interesting and generally well-designed set of products, and was most impressed by the integration of the multiple engines without an enormous impact on system resources. While its in-depth, multi-layered approach may not be necessary for many users, particularly those running older or less powerful hardware for whom the added overhead may be too much, the *TrustPort* suite provides a strong set of utilities useful to anyone for whom security and authenticity, of both systems and data are important concerns. With a little more development to iron out a few issues, *TrustPort* could well become a very strong specialist player in the security marketplace.

Technical details

Supported platforms: The *TrustPort Workstation* suite runs on *Windows NT*, *2000* and *XP*; when submitted for review, only the anti-virus component supported *Windows Vista*, but support has since been added for the *eSign* and *Disk Protection* components; a *Vista*-ready version of the *DataShredder* module is expected soon. The *eSign* and *DataShredder* modules also support *Windows 9x* and *ME*.

Tests were variously run on: *Intel Pentium 4*, 1.6 GHz, 512MB RAM, dual 20GB hard disks, running *Windows 2000 Professional SP4* and *Windows XP SP2*; *AMD K6*, 300 MHz, 256MB RAM, dual 10GB hard disks, running *Windows 2000 Professional SP4;* and *Intel Celeron* laptop, 256MB RAM, 10GB hard disk, running *Windows XP SP2*.

All speed tests were run on: *AMD Athlon64* 3800+ dual core, 1024MB RAM, dual 40GB and 200GB hard disks, running *Windows Vista Business Edition*.

END NOTES & NEWS

Websec 2007 will take place 26–30 March 2007 in London, UK. For programme information and details of how to register see http://www.mistieurope.com/.

Black Hat Europe 2007 Briefings & Training will be held 27–30 March 2007 in Amsterdam, the Netherlands. For programme details see http://www.blackhat.com/.

HITBSecConf2007 - Dubai will take place 2–5 April 2007 in Dubai, UAE. The conference will include presentations by respected members of both the mainstream network security arena as well as the underground or black hat community. For details see http://conference.hackinthebox.org/.

Infosecurity Europe 2007 takes place 24–26 April 2007 in London, UK. Full details of the exhibition and online registration can be found at http://www.infosecurity.co.uk/.

RSA Conference 2007 Japan takes place 25–26 April 2007 in Tokyo, Japan. For more details see http://www.rsaconference.com/.

The 16th annual EICAR conference will be held 5–8 May 2007 in Budapest, Hungary. For programme details and online registration see http://conference.eicar.org/.

DallasCon VI will take place 7–12 May 2007 in Dallas, TX, USA. Programme details and online registration are available at http://www.dallascon.com/.

The 22nd IFIP TC-11 International Information Security Conference takes place 14–16 May 2007 in Sandton, South Africa. For more details see http://www.sbs.co.za/ifipsec2007/.

The 4th Information Security Expo takes place 16–18 May 2007 in Tokyo, Japan. For more details see http://www.ist-expo.jp/en/.

The 8th National Information Security Conference (NISC 8) will be held 16–18 May 2007 at the Fairmont St Andrews, Scotland. For the conference agenda and a booking form see http://www.nisc.org.uk/.

The CISO Executive Summit & Roundtable takes place 6–8 June 2007 in Nice, France. The event will focus on how today's CISO can drive and integrate security into the very core of the business. For details see http://www.mistieurope.com/.

The 19th FIRST Global Computer Security Network conference takes place 17–22 June 2007 in Seville, Spain. For full details see http://www.first.org/conference/2007/.

The Information Security Asia 2007 Conference & Exhibition takes place on 10 and 11 July 2007 in Bangkok, Thailand. For details see http://www.informationsecurityasia.com/.

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. For more details see http://www.haisa.org/.

Black Hat USA 2007 Briefings & Training takes place 28 July to 2 August 2007 in Las Vegas, NV, USA. Registration is now open. All paying delegates also receive free admission to the DEFCON 15 conference, which takes place 3–5 August, also in Las Vegas. See http://www.blackhat.com/.

HITBSecConf2007 - Malaysia will be held 3–6 September 2007 in Kuala Lumpur, Malaysia. For more details see http://conference.hackinthebox.org/.

VB's 17th International Conference, VB2007, takes place 19–21 September 2007 in Vienna, Austria. Full details can be found at http://www.virusbtn.com/conference/.

COSAC 2007, the 14th International Computer Security Forum, will take place 23–27 September 2007 in Naas, Republic of Ireland. See http://www.cosac.net/.

RSA Conference Europe 2007 takes place 22–24 October 2007 in London, UK. See http://www.rsaconference/2007/europe/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

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

Corporate rates include a licence for intranet publication.

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

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

vb Spam supplement

CONTENTS

NEWS & EVENTS

PHISHING TECHNIQUES

Despite the fact that phishing is receiving increasing amounts of media coverage, and people are more aware than ever of the threat, the phishing 'business' seems still to be very attractive for fraudsters, and new phishing tricks appear on a regular basis. From this month, the *VB* website will feature regular in-brief reports on new phishing and spam techniques reported in the wild. See http://www.virusbtn.com/resources/phishing/.

EVENTS

The 2007 Spam Conference will take place on 30 March 2007 in Cambridge, MA, USA. See http://spamconference.org/.

The Authentication Summit 2007 will be held 18–19 April 2007 in Boston, MA, USA. See http://www.aotalliance.org/.

The EU Spam Symposium takes place 24–25 May 2007 in Vienna, Austria. See http://www.spamsymposium.eu/.

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

The 10th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will take place 5–7 June in Dublin, Ireland (members only) and a further meeting (open to both members and non-members) will be held 3–5 October in Washington D.C., USA. See http://www.maawg.org/.

CEAS 2007, the 4th Conference on Email and Anti-Spam, takes place 2–3 August 2007 in Mountain View, CA, USA. Full details can be found at http://www.ceas.cc/.

The Text Retrieval Conference (TREC) 2007 will be held 6–9 November 2007 at NIST in Gaithersburg, MD, USA. See http://plg.uwaterloo.ca/~gvcormac/spam.

FEATURE 1

A PHISH WITH A STING IN THE TAIL

Martin Overton
Independent researcher, UK

Phishing is big news at the moment, not only from the point of view of the victim, and the spiralling costs of this type of fraud to the banks and other financial institutions, but also from the perspective of the cyber-criminal. There is money to be made, and lots of it, from these scams.

The following is the Anti-Phishing Working Group's definition of 'phishing' [1]:

> 'Phishing attacks use both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials. Social engineering schemes use "spoofed" emails to lead consumers to counterfeit websites designed to trick recipients into divulging financial data such as credit card numbers, account usernames, passwords and social security numbers. Hijacking brand names of banks, e-retailers and credit card companies, phishers often convince recipients to respond. Technical subterfuge schemes plant crimeware onto PCs to steal credentials directly, often using Trojan keylogger spyware.'

A recent news story brings this into very sharp focus [2]:

> 'Russian hackers have stolen 800,000 euro from Sweden's largest bank *Nordea* after a sophisticated phishing attack tricked some of its Internet customers into downloading a Trojan horse that recorded their account login details.
>
> 'The first attack took place in August 2006 and was detected a month later. Around 250 of *Nordea*'s customers have been hit by the attack to date.
>
> 'Hackers targeted the bank's customers with emails, purporting to be from *Nordea*, which told them to download an anti-spam tool. But those who downloaded the attachment were infected by the trojan haxdoor.ki.' [3]

The incident described above reminds me of a rather unusual phishing scam that I came across towards the end of 2006. Let me tell you the tale of the one that *didn't* get away.

A 'PHISHY' TALE

Over the last year the number of phishing scams I have seen each month has risen. I usually report around 150 to 500 phishing scams in a 'normal' month (if there is such a thing as a 'normal' month in computer security), but during November 2006 I reported over 3,000 new phishing URLs, many of which were trawling for customers of the UK-based *Barclays* bank.

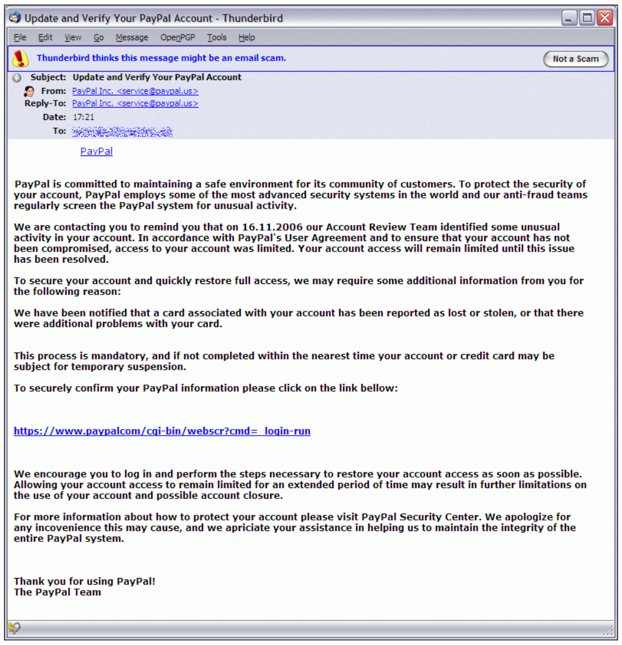
Each phishing scam email I receive is investigated thoroughly. All links are tested against the *Netcraft* toolbar (as well as other toolbars and anti-phishing solutions), and any new ones that it doesn't yet know about are submitted to *Netcraft* for inclusion in the database. This means that many of the new phishing scams have only a very small window of opportunity to hook any new victims that use the *Netcraft* toolbar.

However, once in a while I spot something that makes a new phish stand out from the rest of the shoal. One such time was in November 2006. This article explains why I considered this phish to be something different from the run-of-the-mill phishing scams.

Now, pretend I'm a typical user (it's easy if you try).

FIRST WE BAIT THE HOOK ...

Figure 1 shows a screenshot of the 'baited hook' received by the typical user (who luckily happened to use *PayPal*).


Update and Verify Your PayPal Account - Thunderbird
File Edit View Go Message OpenPGP Tools Help
Thunderbird thinks this message might be an email scam.
Not a Scam

Subject: Update and Verify Your PayPal Account
From: PayPal Inc. <service@paypal.us>
Reply-To: PayPal Inc. <service@paypal.us>
Date: 17:21
To: [redacted]

PayPal

PayPal is committed to maintaining a safe environment for its community of customers. To protect the security of your account, PayPal employs some of the most advanced security systems in the world and our anti-fraud teams regularly screen the PayPal system for unusual activity.

We are contacting you to remind you that on 16.11.2006 our Account Review Team identified some unusual activity in your account. In accordance with PayPal's User Agreement and to ensure that your account has not been compromised, access to your account was limited. Your account access will remain limited until this issue has been resolved.

To secure your account and quickly restore full access, we may require some additional information from you for the following reason:

We have been notified that a card associated with your account has been reported as lost or stolen, or that there were additional problems with your card.

This process is mandatory, and if not completed within the nearest time your account or credit card may be subject for temporary suspension.

To securely confirm your PayPal information please click on the link bellow:

https://www.paypalcom/cgi-bin/webscr?cmd=_login-run

We encourage you to log in and perform the steps necessary to restore your account access as soon as possible. Allowing your account access to remain limited for an extended period of time may result in further limitations on the use of your account and possible account closure.

For more information about how to protect your account please visit PayPal Security Center. We apologize for any incovenience this may cause, and we apriciate your assistance in helping us to maintain the integrity of the entire PayPal system.

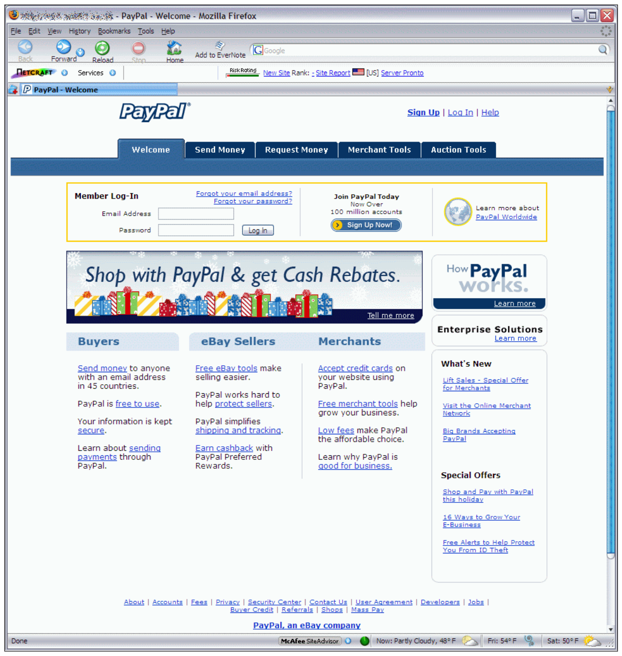
Thank you for using PayPal!
The PayPal Team

Figure 1: The 'baited hook' – the email received by our typical user.

There's nothing too unusual about this, it looks like a typical phishing email aimed at *PayPal* customers. The usual social engineering tricks are used, complete with a fake URL. As you may have guessed, the website the browser ends up at if you click on the link is not the one shown in the email, but it will look very much like the real website.

WE HAVE A BITE ... STRIKE!

Well, the typical user did bite. After panicking and running around like a headless chicken for a while, they clicked on the link. Figure 2 shows what they saw in their web browser (the site has now been closed down).



Member Log-In
Forgot your email address?
Forgot your password?
Email Address
Password

Join PayPal Today
Now Over
100 million accounts

Learn more about PayPal Worldwide

Buyers

Send money to anyone with an email address in 45 countries.

PayPal is free to use.

Your information is kept secure.

Learn about sending payments through PayPal.

eBay Sellers

Free eBay tools make selling easier.

PayPal works hard to help protect sellers.

PayPal simplifies shipping and tracking.

Earn cashback with PayPal Preferred Rewards.

Merchants

Accept credit cards on your website using PayPal.

Free merchant tools help grow your business.

Low fees make PayPal the affordable choice.

Learn why PayPal is good for business.

What's New

Lift Sales - Special Offer for Merchants

Visit the Online Merchant Network

Big Brands Accepting PayPal

Special Offers

Shop and Pay with PayPal this holiday

16 Ways to Grow Your E-Business

Free Alerts to Help Protect You From ID Theft

About | Accounts | Fees | Privacy | Security Center | Contact Us | User Agreement | Developers | Jobs | Buyer Credit | Referrals | Shops | Mass Pay

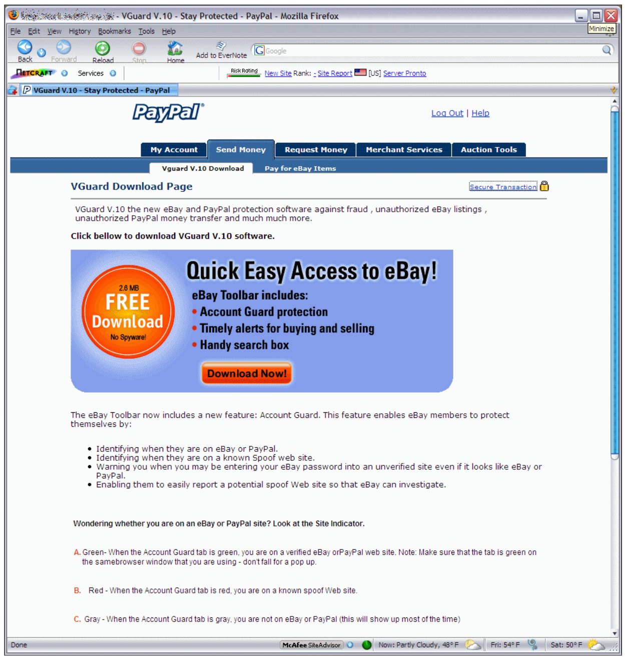
PayPal, an eBay company

Figure 2: 'Phishy' PayPal website.

You can see clearly that at the time I took this screenshot the site was not detected by the *Netcraft* toolbar, or even the *Firefox* anti-phishing functions which are now built into the browser.

As with the original phishing email, there's nothing too surprising here; this is a typical *PayPal* phishing scam site – very believable to those that are not paying attention.

The typical user logged in and filled out the required forms with their name (*Mickey Mouse*), address (*Disneyland, etc.*), social security number (*123-45-6789*), date of birth (*01-01-35*), credit card details including CVV (the three-digit security code) and PIN (when researching this phish I used false credentials, including a computationally valid credit card number from a 'non-existent' credit card).



VGuard Download Page

VGuard V.10 the new eBay and PayPal protection software against fraud , unauthorized eBay listings , unauthorized PayPal money transfer and much much more.

Click bellow to download VGuard V.10 software.

Quick Easy Access to eBay!

eBay Toolbar includes:
- **Account Guard protection**
- **Timely alerts for buying and selling**
- **Handy search box**

The eBay Toolbar now includes a new feature: Account Guard. This feature enables eBay members to protect themselves by:

- Identifying when they are on eBay or PayPal.
- Identifying when they are on a known Spoof web site.
- Warning you when you may be entering your eBay password into an unverified site even if it looks like eBay or PayPal.
- Enabling them to easily report a potential spoof Web site so that eBay can investigate.

Wondering whether you are on an eBay or PayPal site? Look at the Site Indicator.

A. Green- When the Account Guard tab is green, you are on a verified eBay orPayPal web site. Note: Make sure that the tab is green on the samebrowser window that you are using - don't fall for a pop up.

B. Red - When the Account Guard tab is red, you are on a known spoof Web site.

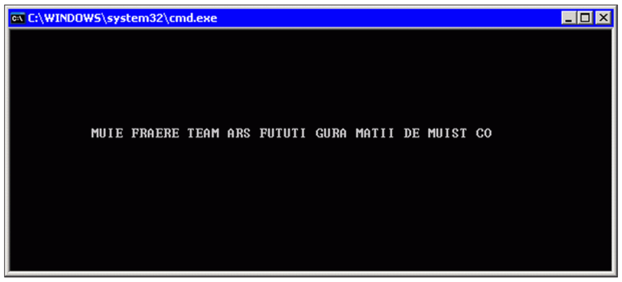
C. Gray - When the Account Guard tab is gray, you are not on eBay or PayPal (this will show up most of the time)

Figure 3: 'Phishy' PayPal website: the confirmation page.

Everything was just like most other *PayPal* phishing sites, until the confirmation page. Figure 3 shows what our typical user saw.

'Oh goody', the typical user thought, 'they're offering me a free download of an *eBay* Toolbar called *VGuard*, and it is at version 10, that's awfully decent of them!' Of course, the typical user downloaded and installed it immediately – as most users do, don't they?

All seemed to be fine – until the rude message in Romanian appeared and the machine rebooted.



```
MUIE FRAERE TEAM ARS FUTUTI GURA MATII DE MUIST CO
```

Figure 4: 'The sting in the tail'. (Picture courtesy of F-Secure. Apologies to Romanian readers.)

THE 'TYPICAL USER' HAS [BEEN] LANDED

Not only has our typical user been landed, they have been gutted and prepared to be devoured – or at least their computer has (more on that later).

So, let me now be the *real* me. What did I do once I had downloaded the 'useful' *eBay* toolbar?

Of course, I started to analyse it. The following is the file information:

```
FileName: Guardv10.exe
FileDateTime: 16/11/2006 17:44:35
Filesize: 149254
MD5: 2fadb5a4f3c80e78197d733255136ba7
CRC32: 7B3A6C60
File Type: PE Executable
Packer: Standard PE File
```

That's interesting, I thought, it isn't even packed using the usual malware authors' tools, such as *UPX*, *FSG*, and so on. Next, I had a quick peek at the internals of the file and discovered that it would create some files and execute them. Not just any files, but a DOS batch (.BAT) file – which was very suspicious.

So, like a good malware analyst, I sent it off to be run in a sandbox. The following are the results (from *Norman Sandbox*):

```
Guardv10.exe : Not detected by Sandbox Signature:
NO_VIRUS)
[ General information ]
* File length: 149254 bytes.
* MD5 hash: 2fadb5a4f3c80e78197d733255136ba7.
[ Changes to filesystem ]
* Creates file C:\TEMP\bt8323.bat.
* Deletes file C:\TEMP\bt8323.bat.
[ Process/window information ]
* Creates an event called .
```

The results from the sandbox confirmed that the downloaded executable created a batch file.

My next question was: what anti-malware tools detect it? To find out I scanned the file using over 30 'up-to-the-minute' updated anti-malware tools. Here are the results (from *AV-Test*):

```
Scan report of Guardv10.exe
@Proventia-VPS Malicious (Cancelled)
AntiVir -
Avast! -
AVG -
BitDefender -
ClamAV -
Command -
Dr Web -
eSafe -
eTrust-INO -
eTrust-INO (BETA) -
eTrust-VET -
eTrust-VET (BETA) -
Ewido -
```

```
F-Prot -
F-Secure -
F-Secure (BETA) -
Fortinet -
Fortinet (BETA) -
Ikarus -
Kaspersky -
McAfee -
McAfee (BETA) -
Microsoft -
Nod32 -
Norman -
Panda Suspicious file
Panda (BETA) Suspicious file
QuickHeal -
Rising -
Sophos -
Symantec -
Symantec (BETA) -
Trend Micro -
Trend Micro (BETA) -
UNA Trojan.BAT.Small.BC0B
VBA32 -
VirusBuster -
WebWasher -
YY_Spybot Jupilites,,Installer
```

As you can see, hardly any of them detected anything at all. I sent the file off to all the anti-malware companies so that they could add detection for it to their products.

PREPARE TO FEAST!

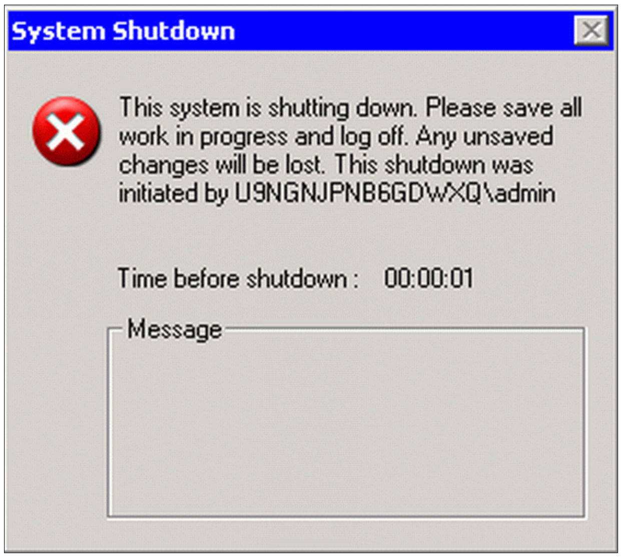
The sting in the tail mentioned in the title of this article is not that the phishers have used a bit of social engineering to get a phished target to give away their personal and financial data, but that they have also got them to download and run a piece of malware – which the typical user thinks is a useful toolbar.

In fact, the 'useful toolbar' does the following [4]:

- It attempts to remove the first four boot configurations from the 'boot.ini' file and then delete the 'hal.dll' file in the Windows '%System%' directory.
- It copies itself to the Windows 'Startup' folder and proceeds to shutdown (reboot) the computer.
- If it is successful, this will make the infected computer unbootable and it may also show a rude message in Romanian on the screen.

Not only have the phishers made off with the user's data, but they are also trying to cover their tracks by making the system unusable.

Any half decent security professional or system administrator could, of course, resolve the matter fairly easily, but most average users would be completely stumped as to how to proceed at this point.


System Shutdown

This system is shutting down. Please save all work in progress and log off. Any unsaved changes will be lost. This shutdown was initiated by U9NGNJPNB6GDWXQ\admin

Time before shutdown : 00:00:01

Message

Figure 5: 'The sting in the tail' going down! (Picture courtesy of F-Secure.)

In most cases they would probably take it to their local PC expert or repair centre and wouldn't realise that it was that 'useful tool' that did the dirty deed.

CONCLUSIONS

As illustrated by the news snippet at the beginning of this article, it seems that typical users are being fooled by this type of phishing scam in which malware is used either to make stealing personal or financial data easier, or to cover the attackers' tracks.

Meanwhile, back at the bank ... well you know how this story ends, and at the moment it's not often a happy ending. The typical user ends up not with a fish supper, but as 'phish phood' instead.

REFERENCES & NOTES

[1] Source: http://www.antiphishing.org/.

[2] Source: http://www.businessweek.com/globalbiz/content/jan2007/gb20070119_387969.htm.

[3] Haxdoor.ki was, allegedly, authored by '*Corpse*' a Russian malware author of some notoriety. More details can be found here: http://www.f-secure.com/weblog/archives/archive-012007.html#00001096.

[4] http://www.f-secure.com/v-descs/killwin_ar.shtml.

FEATURE 2

ENHANCING THE EFFICIENCY OF LEARNING-BASED SPAM FILTERS

Vipul Sharma and Steve Lewis
Proofpoint, Inc., USA

The effectiveness of content-based spam filters is directly related to the quality of the *features* used in the filter's classification model. Features are the specific attributes examined by the spam filter [1, 3]. Highly effective filters may employ an extremely large number of such features (in the order of hundreds of thousands), which can consume a significant amount of both storage space and classification time.

In the ongoing battle between spammers and spam filter developers, new techniques and technologies are continually being introduced by both sides. This means that the number and importance of the features needed to classify spam accurately is subject to continual change. A given feature might be very important at one point in time, but become irrelevant after a few months as spam campaigns and their associated techniques change.

Discarding on a regular basis features that have become ineffective ('bad features') will benefit the spam filter with reduced classification time (reduced model training time and email delivery time), reduced storage requirements, increased spam detection accuracy and a reduced risk of overfitting of the model. (Overfitting occurs when the model trains on a sample set that is skewed by samples that are not representative of real-world threats – while the filter's performance against the training samples will continue to increase, its performance against new, unseen samples will worsen.)

This article reports the results of experimentation with continuous feature-selection methods in real-world spam filters.

FEATURE SELECTION

In machine learning, features are the inherent representation of an instance (email messages in our case). To handle efficiently the continuous introduction of new types of spam emails, it is important to add new features or attributes to the spam filter model. One very important step to keep classifiers efficient is to keep track of these attributes and to monitor their discriminative ability.

It is essential to keep good (highly discriminative) features to ensure ongoing classification accuracy. But it is also important to discard bad (irrelevant or ineffective) features for the following reasons:

- Bad attributes increase the error rate in classification, thus bringing down the overall effectiveness of the filter.
- As an increasingly large number of attributes are added, the complexity of the model grows, resulting in increased computation cost (classification time).
- There is a risk of overfitting the model, caused by redundant or useless attributes.

Being able to distinguish between good and bad features is essential for ensuring the long-term effectiveness of the model. The factors involved in differentiating between good and bad features are described below.

UNDERSTANDING GOOD AND BAD SPAM FEATURES

The logic behind any feature extraction in spam filtering is that the feature should occur frequently in spam messages and infrequently in ham messages (i.e. legitimate, non-spam emails), or vice versa. An ideal feature would occur *only* in spam or *only* in ham messages.

The methods used to evaluate the quality of features are extremely important to ensure effectiveness and low false positive rates.

One well known example of a content-based spam filter is the open source *SpamAssassin* (*SA*), which calculates the effectiveness of a feature using the S/O (spam/overall) metric. The S/O of a feature is the proportion of total occurrences of the feature that were spam messages (i.e. the number of times the feature occurs in spam messages divided by the total number of times the feature occurs). A feature with S/O 1.0 occurs only in spam messages, while a feature with S/O 0.0 occurs only in ham messages.

Measuring the quality of features based purely on their S/O value would bias the classification model towards 'all spam' features, since this metric will only select features that occur frequently in spam emails. It is important also to select features that are indicative of ham.

Table 1 compares the effectiveness of features based on their S/O values. The second column of the table reports the percentage of messages that are spam when a given feature is present. The table shows that the feature '*visit online*' has a higher S/O value than the feature '*X_NO_RULES_FIRED*', since the former is seen in more spam messages than the latter (50% as compared to 20%). However, rating these features purely by their relative S/O values ignores the fact that '*visit online*' is present equally in both spam and ham messages, hence it is of no use in discriminating between the two types.

Feature	Spam	Ham	S/O
Viagra	92.1%	7.9%	0.921
Buy Viagra	99.8%	0.2%	0.998
MSGID_RANDY[*]	82%	18%	0.82
Vi@gr@@[**]	100%	0%	1.0
visit online	50%	50%	0.5
X_NO_RULES_FIRED[***]	20%	80%	0.2

Table 1: Features and their S/O values.

[*]*MSGID_RANDY is a SpamAssassin rule that checks for patterns in headers of spam messages.*

[**]*Vi@gr@@ is a common obfuscation of the drug trade name 'Viagra'.*

[***]*X_NO_RULES_FIRED occurs when no rule or Meta fires, and is indicative of ham messages.*

On the other hand, the feature *'X_NO_RULES_FIRED*' is found significantly more often in ham messages than in spam messages, hence it is a good feature. Using a metric like S/O alone will not select such a feature and the final model will have a higher false positive rate as a result.

In order to address this aspect of feature selection, we benchmarked several statistical feature selection techniques and discuss one of them in the next section.

INFORMATION GAIN

Information Gain (IG) is a widely used method of feature selection in machine learning. The goal of IG is to reduce the overall size of the feature space (i.e. dimensionality reduction). In this way, IG is essentially used as a preprocessing stage prior to training.

IG measures the change in entropy (or randomness) of the model due to a given feature [6]. (A model is more predictive if it is less random. If the randomness decreases due to a feature, it is believed to be a good feature, and if the randomness increases, then the feature is considered to be a bad one.)

Generally, for a training set *S* that consists of positive and negative examples of some target concept (such as spam/ham), the information gain of an attribute *A* that can take values from *value(A)* is given by:

$$IG(S, A) = Entropy(S) \cdot \sum_{v \in value(A)} (|S_v| / |S|)\ Entropy(S_v)$$

S_v is the subset of *S* for which attribute *A* has value *v*

$S_v = \{s \in S | A(s) = v\})$

For a given training set *S*, the Entropy is defined as:

$$Entropy(S) = \sum_{i \in Class} -p_i \log_2 p_i$$

where p_i is the proportion of *S* belonging to class *i*.

In our ongoing investigations of information gain as a method of feature selection, an IG threshold was chosen to produce the best accuracy on the training corpora (the set of spam and ham messages used to train the learning-based spam filter). Features that had an IG below the threshold were discarded.

We observed a performance boost of around 8% after discarding the features with IG below the threshold. Some of the rules were optimized after understanding the tricks spammers were using to bypass them, and many other rules were optimized for improved time performance.

We also noticed that after removing bad features, the error rate on the training data was reduced – meaning that we were producing better models that resulted in greater anti-spam effectiveness. Employing this process on a regular basis ensures that the feature set is cleaned of ineffective features, thereby ensuring a high level of effectiveness over time.

CONCLUSION

Regular feature extraction is required to keep spam filters functioning at the highest levels of effectiveness, but this can also result in an ever-increasing feature set and accompanying increases in processing time.

IG has been shown to be an effective method for measuring the quality of features and determining those which should be discarded. Using our technique, we were able to improve our spam filter's performance substantially and increase its accuracy. The use of feature selection also decreases the risk of overfitting as the filter no longer trains itself on bad features.

REFERENCES

[1] SpamAssassin. http://www.spamassassin.org/.

[2] Dalvi, N.N.; Domingos, P.; Mausam; Sanghai, S.; Verma, D. Adversarial classification. KDD2004.

[3] Rios G.; Zha, H. Exploring SVM and Random Forest for Spam Detection. CEAS 2005.

[4] Androutsopoulos, I.; Paliouras, G.; Michelakis, E. Learning to filter unsolicited commercial email. Technical Report, National Centre for Scientific Research \Demokritos 2004.

[5] Harremoeës, P.; Topsøe, F. Maximum Entropy Fundamentals. Entropy 2001, 3[3], 191-226.

[6] Mitchell, T.M. Textbook – 'Machine Learning'. McGraw Hill, ISBN 7-111-11502-3/TP. 2760.

[7] Veysset, F. Honeypot Technologies, SAR Conference, 2005.