JANUARY 2011

virus BULLETIN

Fighting malware and spam

CONTENTS

IN THIS ISSUE

RISING STAR

As in normal business, one of most effective ways for a banker trojan to gain market share is to do things better than its competitors, and if possible, make the migration from competitors to its own business as easy as possible. Carberp does both of these things very successfully. Toni Koivunen has all the details of this rising star in the banker trojan scene.

page 4

STANDING ON THE CUSP

Andrew Lee takes stock of the anti-malware industry and the world at large as we stand on the cusp of a new decade.

page 13

ESCAN INTERNET SECURITY 11

VB's lab team take a close look at eScan Internet Security 11, commenting on its funky styling and the wealth of features it has to offer.

page 17

VBSPAM CERTIFICATION

19 full solutions and one partial solution were put to the test in the VBSpam lab this month. Martijn Grooten has the details.

page 22

vb VERIFIED SPAM virusbtn.com

ISSN 1749-7027

'Whilst bigger may not always be better, it is clearly always bigger. And bigness begets bigness.'

Paul Ducklin, Sophos

HOW DO WE MAKE BIGGER BETTER?

You've probably heard of Alan Kay, the principal designer of the object-oriented language Smalltalk. The original Smalltalk implementation was written by Dan Ingalls – allegedly to settle a bet that only a single page of code would be needed to implement the language[1].

How times have changed! I know I'm being slightly unfair here, proverbially comparing apples with oranges (though in practice they compare quite well[2]), but the current *Linux* kernel sources consist of some 35,000 files totalling over 400MB in size.

I'd love to tell you how many lines of source code that is, and on any Unix-like system that should be easy:

```
$ wc $(find . -type f -print)
```

Use 'find' to enumerate a list of all the files in the *Linux* source tree. Pass that list to 'wc', counting the words and lines in each file and producing a total. With 4GB RAM at my disposal, this should be easy. But it isn't:

```
-bash: /usr/bin/wc: Argument list too long
```

Clearly, bigger is not always better.

Nevertheless, computing seems set on interminable expansion. I don't mean to criticize anyone in particular, but I need an example, and I've chosen *Adobe*. Its entry-level PDF-reading software *Reader*, is now a 70MB download which asks for over 400MB of disk space.

More interestingly (at least for a threat researcher), *Reader* now incorporates runtimes for three different programming languages: PostScript, ActionScript and ECMAScript, more commonly known as JavaScript.

In the interest of objectivity, I'll admit that anti-malware software is getting larger, too. *Sophos Endpoint Security for Windows*, for instance, is an 80MB download, expanding to around 110MB of disk space.

However, it's fair to point out that the overall complexity of products in our industry is, to some extent, determined by the sum of the rest of the parts. We aren't just trying to protect *Adobe Reader* from attack, or your browser, or your video player, or your operating system, but *all* of them.

Since your browser contains a JavaScript engine in which attacks can be played out, we need a reasonably complete analogue of that environment to deliver satisfactory malware prevention. We need an analogue of your CPU and operating system, too. We need to know how to unravel complex files (archives, images, software bundles etc.) in a way that takes into account all the known peccadillos of all the commonly used software that consumes those files.

So, whilst bigger may not always be better, it is clearly always bigger. And bigness begets bigness.

For security, this means that the gap between functionality and security in software and services is likely to remain wide, and will probably widen further.

Even if you embrace the 'thin client' model – *Google*'s *ChromeOS* project springs to mind, in which the browser is just about the only software on your PC – you won't be free from bigness. *Facebook*, for example, which exists only in your browser, is an enormous, new and fruitful stamping ground for cybercrooks.

In short, therefore, education and user awareness are always going to be important. In fact, they are probably the most important aspects of computer security, since they continue to protect us when technology cannot.

In the past year, I've heard a number of fellow security researchers writing off user education altogether. I think that's defeatist, and ultimately self-defeating. If education fails, you can blame the students. But you might equally blame the teachers.

And what about Alan Kay? In 2001, he took the intriguing step of starting an organization to investigate how far you might get in the modern world with just 20,000 lines of code. Have a look for yourself – you might be pleasantly surprised[3].

[1] http://en.wikipedia.org/w/index.php?title=Smalltalk&oldid=401932634.

[2] http://improbable.com/airchives/paperair/volume1/v1i3/air-1-3-apples.html.

[3] http://news.squeak.org/2007/02/15/complete-computing-system-in-20k-lines-of-code/.

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
Web Developer: Paul Hettler
Consulting Editors:
Nick FitzGerald, *Independent consultant, NZ*
Ian Whalley, *IBM Research, USA*
Richard Ford, *Florida Institute of Technology, USA*

NEWS

MOBILE USERS MORE VULNERABLE TO PHISHING

Recent research conducted by online banking security firm *Trusteer* has indicated that those who use a mobile device to access the Internet are three times more likely to fall victim to phishing scams than those using a standard desktop PC.

Researchers analysed the log files of several web servers that were hosting phishing websites and were able to determine how many users accessed the websites, when they visited them, whether they submitted their login information and what devices they used to access the website. They found that not only are mobile device users the first to arrive at phishing sites, but they are also three times more likely to submit their login details than other users.

Trusteer suggests that reason for mobile users being quicker to access the sites than non-mobile users is that mobile devices tend to be carried with the user, allowing the user to read emails (and take action) as and when they arrive – whereas users of a desktop PC would only read emails when they have access to their computer. The company says that the fact that mobile users appear to be more gullible than their non-mobile counterparts may be explained by the fact that it is harder to spot a phishing website on a mobile device than on a desktop computer – for example the full URL may not be visible.

‘CYBER ARMY’ FORMED IN ESTONIA

The Estonian government has established a volunteer force of programmers, computer scientists and software engineers that in wartime would function as a ‘Cyber Defense League’ under a unified military command.

The force reportedly carries out regular weekend exercises in preparation for possible cyber contingencies.

In early 2007 Estonian government and national infrastructure sites were hit with several weeks’ worth of DDoS attacks which coincided with political unrest over Russia’s history in Estonia.

Estonia is one of the most wired countries in eastern Europe, relying on the Internet for a substantial portion of everyday life – 80% of Estonians reportedly using the Internet to pay taxes and conduct financial transactions. Indeed, in 2000, the Estonian government declared Internet access a basic human right. It was this wide scale dependence on the Internet that left the country particularly vulnerable to the large-scale cyber attack in 2007.

Although the defence league is currently made up of volunteers, Estonian Defence Minister Jaak Aaviksoo has hinted that conscription is not out of the question.

Prevalence Table – November 2010[1]

Malware	Type	%
Autorun	Worm	11.32%
VB	Worm	6.84%
Conficker/Downadup	Worm	5.90%
Agent	Trojan	4.82%
OnlineGames	Trojan	4.01%
Heuristic/generic	Trojan	3.87%
Heuristic/generic	Virus/worm	3.55%
FakeAlert/Renos	Rogue AV	3.53%
Injector	Trojan	3.41%
Delf	Trojan	2.78%
Adware-misc	Adware	2.57%
Sality	Virus	2.55%
Downloader-misc	Trojan	2.52%
Kryptik	Trojan	1.99%
Zbot	Trojan	1.91%
StartPage	Trojan	1.90%
Small	Trojan	1.83%
Exploit-misc	Exploit	1.74%
Virut	Virus	1.72%
Crypt	Trojan	1.66%
Hupigon	Trojan	1.51%
Allaple	Worm	1.46%
Alureon	Trojan	1.36%
AutoIt	Trojan	1.28%
Suspect packers	Misc	1.25%
Vobfus	Trojan	1.22%
Dropper-misc	Trojan	1.21%
Iframe	Exploit	1.12%
FakeAV-Misc	Rogue AV	0.97%
Bifrose/Pakes	Trojan	0.96%
Tanatos	Worm	0.95%
Crack/Keygen	PU	0.87%
Others[2]		15.43%
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

CARBERP, A NEW BAG OF TRICKS

Toni Koivunen
F-Secure, Finland

Carberp, a rising star in the banker trojan scene, first appeared in early 2010. Until recently, the Zeus trojan has been pretty dominant in the banker scene, but lately a few new contestants have entered the arena, SpyEye and Carberp being the most notable. As in normal business, one of the most effective ways to gain market share is to do things better than your competitors, and if possible, make the migration from competitors to your own business as easy as possible. Carberp does both of these things very successfully.

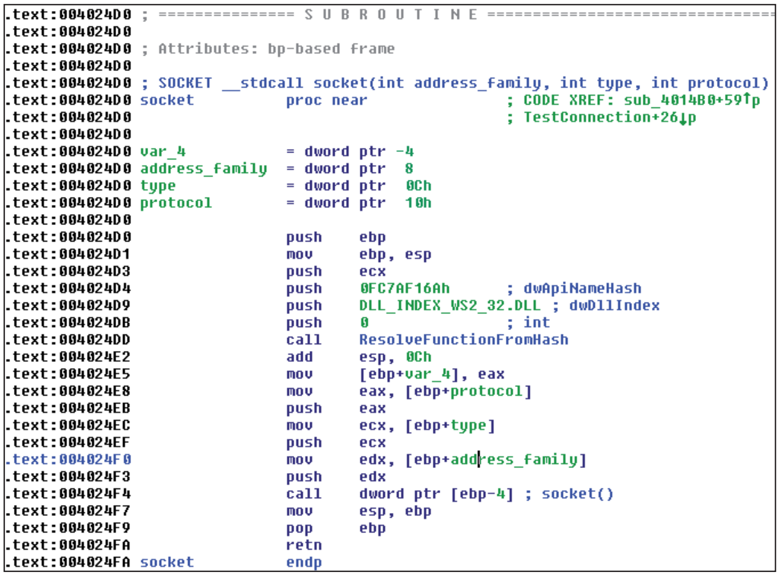
The data in this analysis was reversed from a Carberp variant with SHA-1: ba110352e6a5fb291973d4ea2f3ef3a0231f8afe. After a few days of tinkering I had a sufficiently reversed IDB in my hands (with 260 out of 350 functions named and prototyped).

Even though the main functionality of Carberp revolves around stealing bank credentials it is capable of a lot more tricks than that, mainly thanks to its modularity. Carberp is almost completely memory resident, requiring only a single file on the hard drive, and it can survive without administrative rights. The icing on the cake is that it has its own 'anti-virus' module that is designed to nuke any competition.

NO HABLO IMPORTS

One of the most striking features when taking a look at the fresh IDB after unpacking is the complete lack of imported functions. The way Carberp does its deeds is that wrappers are created for all the WINAPI calls (stdlib functions such as strcmp and so on are compiled statically).

Figure 1 shows a screenshot for the wrapper around the socket() call.



```
.text:004024D0 ; =============== S U B R O U T I N E ===============================
.text:004024D0
.text:004024D0 ; Attributes: bp-based frame
.text:004024D0
.text:004024D0 ; SOCKET __stdcall socket(int address_family, int type, int protocol)
.text:004024D0 socket          proc near               ; CODE XREF: sub_4014B0+59↑p
.text:004024D0                                         ; TestConnection+26↓p
.text:004024D0
.text:004024D0 var_4           = dword ptr -4
.text:004024D0 address_family  = dword ptr  8
.text:004024D0 type            = dword ptr  0Ch
.text:004024D0 protocol        = dword ptr  10h
.text:004024D0
.text:004024D0                 push    ebp
.text:004024D1                 mov     ebp, esp
.text:004024D3                 push    ecx
.text:004024D4                 push    0FC7AF16Ah      ; dwApiNameHash
.text:004024D9                 push    DLL_INDEX_WS2_32.DLL ; dwDllIndex
.text:004024DB                 push    0               ; int
.text:004024DD                 call    ResolveFunctionFromHash
.text:004024E2                 add     esp, 0Ch
.text:004024E5                 mov     [ebp+var_4], eax
.text:004024E8                 mov     eax, [ebp+protocol]
.text:004024EB                 push    eax
.text:004024EC                 mov     ecx, [ebp+type]
.text:004024EF                 push    ecx
.text:004024F0                 mov     edx, [ebp+address_family]
.text:004024F3                 push    edx
.text:004024F4                 call    dword ptr [ebp-4] ; socket()
.text:004024F7                 mov     esp, ebp
.text:004024F9                 pop     ebp
.text:004024FA                 retn
.text:004024FA socket          endp
```

Figure 1: Wrapper around socket() call.

The dwDllIndex refers to an array inside the ResolveFunctionFromHash() function that contains all the DLL names used by Carberp. The same hashing scheme as is used on the WINAPI names is also used on some other strings including a few process names that the malware is specifically trying to find. The hashing scheme is shown below:

```
DWORD HashString(char *pszApiName)
{
    DWORD retval = 0;
    char byte;
    char *copy = pszApiName;
    if(pszApiName == NULL)
    {
        return -1;
    }
    byte = pszApiName[0];
    while(byte != NULL)
    {
        retval = (retval << 7) | (retval >> 0x19);
        unsigned char key = copy[0];
        retval = key ^ retval;
        copy++;
        byte = copy[0];
    }
    return retval;
}
```

After locating the hash function it was pretty trivial to port it into a tool that hashes all the exports from all the used DLLs into a single text file. That file was then used along with the IDB to map all the wrapper functions.

THOU SHALT NOT HOOK

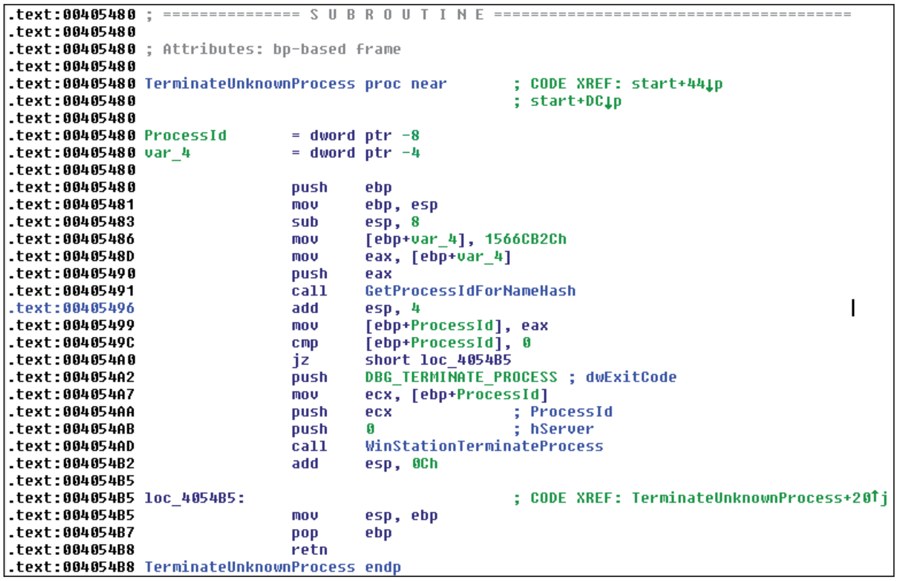
One interesting feature that was spotted when running the Carberp sample in a sandbox was that it created a lot of temporary files. Closer inspection revealed that this is part of a functionality that is designed to defeat possible user-mode hooks. At several points, including the point at which the sample starts up, Carberp performs a number of checks, pushing several WINAPI name hashes and DLL names to a function.

The called function locates the specified DLL and copies it to a filename that it builds by using GetTempPathW and GetTempFileNameW. Then, using CreateFileMappingW and MapViewOfFile, the DLL in question is mapped into memory and the function starts to parse the Export directory and hash the WINAPI names, looking for the given hashes. When the correct hash is found, the function will compare the first 0x0A bytes of the function, both from the freshly mapped file and the DLL that has already been loaded into memory by the PE loader. If the bytes are different, it will copy the 0x0A bytes from the mapped DLL to the 'real' function offset, thus eliminating possible user-mode hooks in the process – extremely efficient, at least in theory. Carberp doesn't even bother to set any flags to indicate that hooks have been detected. While this is a beautiful design, its implementation gets an 'F' for fail.

There is a tiny mistake in the code. For instance, when Carberp is checking ZwSetContextThread for possible hooks, the actual bytes being checked belong to the ZwRollforwardTransactionManager WINAPI call. I guess someone missed the fact that the number of exported names can differ from the number of exported ordinals. This renders the anti-hooking functionality pretty much useless, at least for the time being.

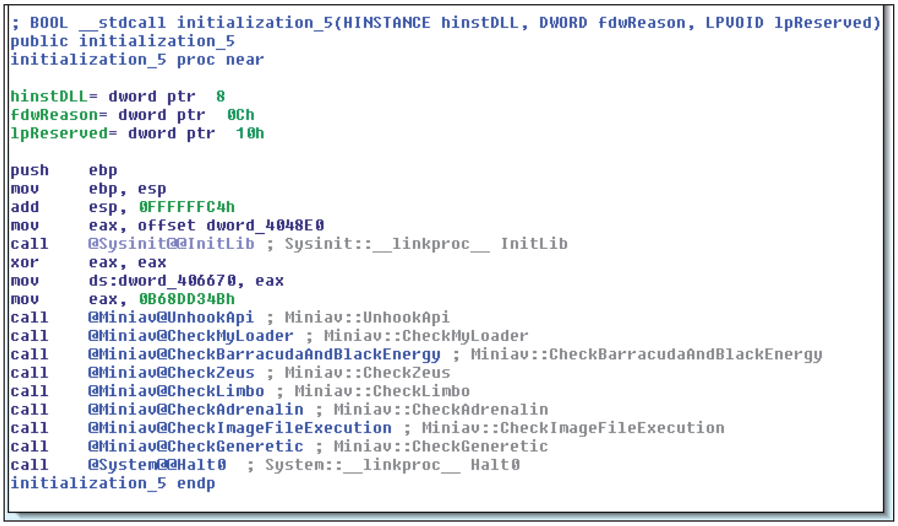
PROCESS TERMINATION

One interesting thing is that in a few places, early in the main function, Carberp calls the GetProcessIdForNameHash function (see Figure 2).



```
.text:00405480 ; =============== S U B R O U T I N E =======================================
.text:00405480
.text:00405480 ; Attributes: bp-based frame
.text:00405480
.text:00405480 TerminateUnknownProcess proc near       ; CODE XREF: start+44↓p
.text:00405480                                         ; start+DC↓p
.text:00405480
.text:00405480 ProcessId       = dword ptr -8
.text:00405480 var_4           = dword ptr -4
.text:00405480
.text:00405480                 push    ebp
.text:00405481                 mov     ebp, esp
.text:00405483                 sub     esp, 8
.text:00405486                 mov     [ebp+var_4], 1566CB2Ch
.text:0040548D                 mov     eax, [ebp+var_4]
.text:00405490                 push    eax
.text:00405491                 call    GetProcessIdForNameHash
.text:00405496                 add     esp, 4
.text:00405499                 mov     [ebp+ProcessId], eax
.text:0040549C                 cmp     [ebp+ProcessId], 0
.text:004054A0                 jz      short loc_4054B5
.text:004054A2                 push    DBG_TERMINATE_PROCESS ; dwExitCode
.text:004054A7                 mov     ecx, [ebp+ProcessId]
.text:004054AA                 push    ecx             ; ProcessId
.text:004054AB                 push    0               ; hServer
.text:004054AD                 call    WinStationTerminateProcess
.text:004054B2                 add     esp, 0Ch
.text:004054B5
.text:004054B5 loc_4054B5:                             ; CODE XREF: TerminateUnknownProcess+20↑j
.text:004054B5                 mov     esp, ebp
.text:004054B7                 pop     ebp
.text:004054B8                 retn
.text:004054B8 TerminateUnknownProcess endp
```

Figure 2: Carberp calls GetProcessIdForNameHash.



```
; BOOL __stdcall initialization_5(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)
public initialization_5
initialization_5 proc near

hinstDLL= dword ptr  8
fdwReason= dword ptr  0Ch
lpReserved= dword ptr  10h

push    ebp
mov     ebp, esp
add     esp, 0FFFFFFC4h
mov     eax, offset dword_4048E0
call    @Sysinit@@InitLib ; Sysinit::__linkproc__ InitLib
xor     eax, eax
mov     ds:dword_406670, eax
mov     eax, 0B68DD34Bh
call    @Miniav@UnhookApi ; Miniav::UnhookApi
call    @Miniav@CheckMyLoader ; Miniav::CheckMyLoader
call    @Miniav@CheckBarracudaAndBlackEnergy ; Miniav::CheckBarracudaAndBlackEnergy
call    @Miniav@CheckZeus ; Miniav::CheckZeus
call    @Miniav@CheckLimbo ; Miniav::CheckLimbo
call    @Miniav@CheckAdrenalin ; Miniav::CheckAdrenalin
call    @Miniav@CheckImageFileExecution ; Miniav::CheckImageFileExecution
call    @Miniav@CheckGeneretic ; Miniav::CheckGeneretic
call    @System@@Halt0  ; System::__linkproc__ Halt0
initialization_5 endp
```

Figure 3: The miniav.plug module.

The GetProcessIdForNameHash function calls the ZwQuerySystemInformation function, passing SystemProcessesAndThreadsInformation as a parameter. In return, it receives a list of running processes and threads in a SYSTEM_PROCESSES struct. It then iterates through the list of processes, checking all process names against the provided parameter (1566CB2Ch). If a match is found, it will call the WinStationTerminateProcess function to terminate the process. Funnily enough, WinStationTerminateProcess requires Terminal Services to be enabled in order for it to work. As to the specific process that is being targeted with this, I don't know. It was not running on my computer so I didn't see what the hash would match. I did try brute forcing it for a while, making around 22M attempts per second on a dual-core but I didn't get any hits, so the target for termination remains a mystery for now.

O PLUG-INS, WHERE ART THOU?

Carberp uses five executable modules, two of which are embedded in the binary itself and three that are downloaded from the C&C server. All of the modules are encrypted with the same algorithm (the webinjects are encrypted with it as well). The three modules that are downloaded are:

- miniav.plug
- stopav.plug
- passw.plug

Miniav.plug is a DLL file, apparently created by the author of Carberp, that is a form of AV engine designed to wipe out competition such as Zeus from the infected machine. The author of the module was apparently feeling generous and left the debug information in the module. I believe the main function of the miniav.plug file says all that is needed (see Figure 3).

Stopav.plug is another DLL file. This one is designed to disable existing AV products on the infected machine. It works by creating a process belonging to the targeted AV and then injects a small thread into that process. The injected thread is responsible for removing the file(s) belonging to the AV product, after which it will kill the process itself. I suspect that the reasoning behind this is that if the file removal command comes from within the AV product's own process, there is less chance that the AV engine will object.

The third downloaded component is passw.plug. Like miniav.plug and stopav.plug, this module is written in Delphi. The purpose of the module is to run and grab as much data worth stealing as possible. The list of software that is targeted includes:

- Camfrog
- Cached passwords (WNetEnumCachedPasswords)
- AOL Instant Messenger
- Google Talk
- Windows Live
- PalTalk
- AIMPro
- QIP.Online
- JAJC from Abstract Software
- Internet Explorer 7
- WebSitePublisher from Crye
- ICQ
- MSN Messenger
- Yahoo! Messenger
- Gadu-gadu
- Jabber
- Miranda
- &RQ
- Mozilla Firefox
- RIT The Bat!

All the data gathered by passw.plug is passed back to the main binary, which sends the data directly to the C&C server without storing it even temporarily to the disk. The passw.plug module appears to be a variant of LdPinch.

One of the executable modules that resides within the body of the main binary itself is used to take screen captures. The screen captures are also sent directly to the C&C without storing them on disk.

None of the five modules that Carberp uses ends up on the disk surface. They are decrypted on the fly and then mapped into memory, thus reducing the telltale signs of disk activity.

THE ENCRYPTION

As mentioned, Carberp uses the same encryption scheme on all the modules it uses as well as the webinjects. The following is the decryption scheme reversed from the binary (with a few lines of additional code for clarification):

```
unsigned char *pData = (unsigned char *);
malloc(dwFileLength);
fread(pData, 1, dwFileLength, hFile);
fclose(hFile);

unsigned long dwLength = 0;
if(memcmp(pData,"BJB",3) == 0)
{
        unsigned long dwKeyLength = pData[3];
        printf("KeyLength is %d.\r\n",dwKeyLength);
        unsigned char *pKey = (unsigned char *);
        malloc(dwKeyLength + 1);
        memset(pKey, 0, dwKeyLength + 1);
        memcpy(pKey,(const void *)&pData[7],;
        dwKeyLength);
        pData = pData + 7 + dwKeyLength;
        dwLength = dwFileLength - 7 - dwKeyLength;
        unsigned long i, j;
        for(i = 0; i < dwLength; i++)
        {
           for(j = 0; j <= dwKeyLength; j++)
           {
                if(pKey[j] == 0){
                           break;
                }

                pData[i] = pData[i] ^ (pKey[j] + (i * j));
           }
        }
}
else
{
        printf("This does not appear to be a Carberp;
        .plug file, aborting.\r\n");
        return 0;
}
```

So basically the XOR key is carried in the file itself. The first DWORD after the 'BJB' header specifies the length of the key, and the encryption portion starts directly after the key. All the observed keys have been between six and eight characters long, with the character set containing only digits, 0–9.

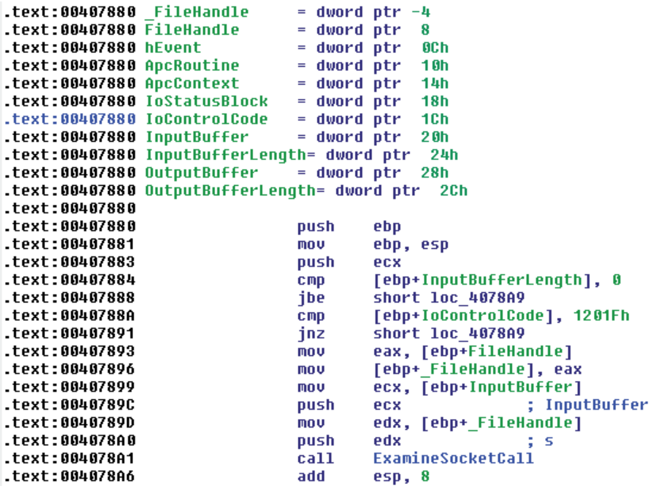
API HOOKS

Although Carberp doesn't like being hooked, it sure does like to hook others. It copies the required number of bytes from the API to a new location, writes a jmp hook (0xE9) to the original API with the jmp offset pointing to the hook, and writes another jmp to the end of the 'stolen' bytes where the jump will return to the original API.

The following is the list of hooked functions:

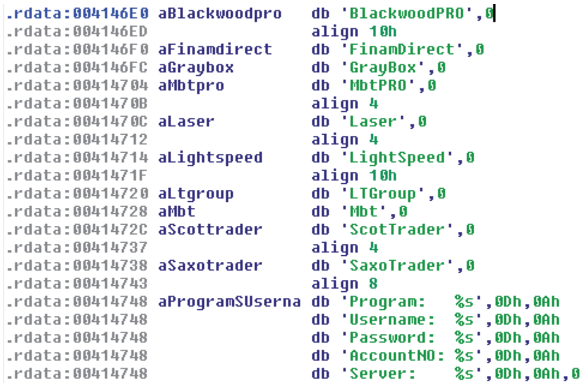
- nspr4.dll!PR_Connect
- nspr4.dll!PR_Write
- nspr4.dll!PR_Read
- nspr4.dll!PR_Close
- ssl3.dll!SSL_ImportFD
- wininet.dll!HttpSendRequestA
- wininet.dll!HttpSendRequestW
- wininet.dll!HttpSendRequestExA
- wininet.dll!HttpSendRequestExW
- wininet.dll!InternetReadFile
- wininet.dll!InternetReadFileExA
- wininet.dll!InternetReadFileExW
- wininet.dll!InternetQueryDataAvailable
- wininet.dll!InternetCloseHandle
- ntdll.dll!ZwResumeThread
- ntdll.dll!ZwQueryDirectoryFile
- ntdll.dll!ZwDeviceIoControlFile
- ntdll.dll!ZwClose

Most of the hooks are for stealing data, with a few that are used to hide the malware's presence (ZwQueryDirectoryFile for instance). One thing that seemed odd at first was the presence of the ZwDeviceIoControlFile hook. On further examination it turned out that the hook exists solely to steal FTP credentials. At the beginning of the hook a check is made as to whether the IoControlCode matches what is used when a program makes a send() call (Figure 4).



```
.text:00407880 _FileHandle      = dword ptr -4
.text:00407880 FileHandle       = dword ptr  8
.text:00407880 hEvent           = dword ptr  0Ch
.text:00407880 ApcRoutine       = dword ptr  10h
.text:00407880 ApcContext       = dword ptr  14h
.text:00407880 IoStatusBlock    = dword ptr  18h
.text:00407880 IoControlCode    = dword ptr  1Ch
.text:00407880 InputBuffer      = dword ptr  20h
.text:00407880 InputBufferLength= dword ptr  24h
.text:00407880 OutputBuffer     = dword ptr  28h
.text:00407880 OutputBufferLength= dword ptr  2Ch
.text:00407880
.text:00407880                  push    ebp
.text:00407881                  mov     ebp, esp
.text:00407883                  push    ecx
.text:00407884                  cmp     [ebp+InputBufferLength], 0
.text:00407888                  jbe     short loc_4078A9
.text:0040788A                  cmp     [ebp+IoControlCode], 1201Fh
.text:00407891                  jnz     short loc_4078A9
.text:00407893                  mov     eax, [ebp+FileHandle]
.text:00407896                  mov     [ebp+_FileHandle], eax
.text:00407899                  mov     ecx, [ebp+InputBuffer]
.text:0040789C                  push    ecx             ; InputBuffer
.text:0040789D                  mov     edx, [ebp+_FileHandle]
.text:004078A0                  push    edx             ; s
.text:004078A1                  call    ExamineSocketCall
.text:004078A6                  add     esp, 8
```

Figure 4: A check is made.



```
.rdata:004146E0 aBlackwoodpro   db 'BlackwoodPRO',0
.rdata:004146ED                 align 10h
.rdata:004146F0 aFinamdirect    db 'FinamDirect',0
.rdata:004146FC aGraybox        db 'GrayBox',0
.rdata:00414704 aMbtpro         db 'MbtPRO',0
.rdata:0041470B                 align 4
.rdata:0041470C aLaser          db 'Laser',0
.rdata:00414712                 align 4
.rdata:00414714 aLightspeed     db 'LightSpeed',0
.rdata:0041471F                 align 10h
.rdata:00414720 aLtgroup        db 'LTGroup',0
.rdata:00414728 aMbt            db 'Mbt',0
.rdata:0041472C aScottrader     db 'ScotTrader',0
.rdata:00414737                 align 4
.rdata:00414738 aSaxotrader     db 'SaxoTrader',0
.rdata:00414743                 align 8
.rdata:00414748 aProgramSUserna db 'Program:   %s',0Dh,0Ah
.rdata:00414748                 db 'Username:  %s',0Dh,0Ah
.rdata:00414748                 db 'Password:  %s',0Dh,0Ah
.rdata:00414748                 db 'AccountNO: %s',0Dh,0Ah
.rdata:00414748                 db 'Server:    %s',0Dh,0Ah,0
```

Figure 5: Strings in the code suggest Carberp might be targeting programs relating to stock trading.

If the IoControlCode matches, the socket handle and input buffer are passed onto another function (ExamineSocketCall in the example above). Further checks are performed to check that the remote address is a loopback address and that the port is not 21. If all the checks are passed, the hook will rip out the USER and PASS fields from the stream and upload them directly to the C&C server, to a script called sni.html.

WHEN I GROW UP I WANT TO WORK ON THE STOCK MARKET…

Based on certain strings in the code (see Figure 5), it seems that Carberp might be arming up to capture credentials related to various programs that are used in stock market trading.

This is a bit worrying. The potential for financial damage (or, for the attackers, financial gain) is pretty high. Credentials and access to the systems that are listed in the code would enable the attackers to gain insider information as well as to perform fraudulent sales and/or purchases on stock markets. Luckily, this has not yet been implemented, but as Carberp is pretty new to the scene we may see it happening soon.

AFTERWORDS

Even though Carberp is relatively new, its authors have demonstrated quite a deep technical expertise and the ability to innovate. Unless they mess up significantly, they are well on the road to snatching some market share from Zeus and the rest of the gang and it is likely that Carberp will slowly but surely gain in popularity.

FEATURE

WHAT'S THE DEAL WITH SENDER AUTHENTICATION? PART 6

Terry Zink
Microsoft, USA

In the last article in this series (see *VB*, December 2010, p.12), we looked at how digital signatures in email are accomplished through the use of DomainKeys Identified Mail, or DKIM. While DKIM does have its niche applications, and is useful for whitelisting and identification in the positive case, one of the barriers to mass implementation is that it is less useful for detecting spoofing. This is because the protocol states that the failure of a DKIM validation should be treated as if there were no DKIM signature at all in the message. Since the receiver of a message doesn't know whether or not a domain should even have a DKIM signature, the lack of one doesn't indicate that a message is spoofed. There are any number of legitimate reasons why it might not have one. If a broken signature is just as 'valid' as the lack of a signature, then for most receivers who are primarily concerned about filtering spam, the usefulness of DKIM is minimal.

ADSP

Author Domain Signing Policies, or ADSP[1], is an optional extension to DKIM that allows senders and receivers to specify whether a domain signs all of its mail or only some of its mail with a DKIM signature.

The protocol is simple. Any domain that signs with DKIM can publish one of three values in the dkim= field of its DNS txt record at _adsp._domainkey.<domain>.com:

- *unknown* – the domain might sign some or all of its outbound mail. If a message from this domain arrives and is signed, it can be treated as authoritative. However, if a message arriving from this domain is not signed, then nothing can be concluded about the source of the message.

  An 'unknown' result is similar to the neutral result of an SPF check. In SPF/SenderID, a neutral result means that the message should be treated as if it had no SPF record at all. You cannot use this result for detection of spoofing, only positive identification of authorized IPs. In a similar manner, the 'unknown' field means that you are only validating signed mail if it exists.

- *discardable* – all mail from the domain is signed with an Author Domain Signature. The *lack* of a DKIM signature from this domain means that you can discard the message (mark it as spam, reject it during the SMTP transaction before the 250, etc.).

  A 'discardable' result is similar to an SPF hard fail. SPF hard fails are strong assertions about the message and indicative of spoofing. Or rather, they are supposed to be indicative of spoofing, but there are many cases when legitimate messages hard fail SPF checks. With DKIM, if a message has no signature and the domain says it is discardable, you can safely reject the message. Regardless of the reason for the lack of signature, the message cannot be trusted and should be discarded.

  Note that you can only discard the message for the *lack* of a signature. The protocol does not say that you should discard a message with a signature that does not validate.

- *all* – all mail from the domain is signed with an Author Domain Signature. The case of 'all' is ambiguous. The protocol does not indicate what you should do with a domain that publishes 'dkim= all', but what action you should take when a message arrives without a signature. Did the sender forget to sign the message? If so, he hasn't used the 'discardable' option to specify that a message without a signature can be discarded. If he was confident that he signed every single message, he would have used 'discardable'. But he didn't – so what should we do with this message?

  In this case, the 'all' option is best treated in the same way as an SPF soft fail. The soft fail indicates that the message should be accepted, but can be used as a low weight in a spam filter, perhaps as part of the content filter. The best course of action is to do the same thing with any message that fails the 'all' test – the lack of a signature means it can be used as a low weight in the content filter. If it passes, then take the normal action that you would typically take.

CONSTRAINTS

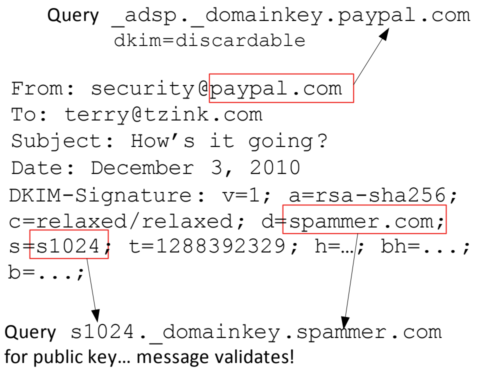
The use of ADSP nets some advantages but also imposes some constraints on the sender. One of the flexibilities of DKIM is that it allows a sender to send on behalf of someone else. The signing domain is specified in the d= field in the DKIM signature header. Thus, the From: field can specify the domain that the sender wants the recipient to see in their inbox, while reputation checks can be performed on the signing domain. For example, if travel company Oceanic contracts out the sending of its marketing messages to Big Communications, Inc.,

[1] ADSP is defined in RFC 5617, http://tools.ietf.org/html/rfc5617.

Big Communications will sign the message and put 'bigcommunications.com' in the d= field, but 'marketing@oceanic.com' in the From: field. The receiver performs the reputation check not on oceanic.com but on bigcommunications.com. Big Communications is taking responsibility for the quality of the message.

However, in order to use ADSP, the sender's From: field *must* be the same as the domain in the d= field. The reason is that the signing policy must first be looked up before checking to see if the DKIM-Signature field exists. The mail receiver first performs the lookup for the domain in the From: field to see whether or not it has a signing policy, and if it does, the receiver extracts data from the DKIM-Signature field (if it exists). If it does not exist, then the action specified by the ADSP is applied.

If the From: field is different, then this defeats the entire point of ADSP. If the domain is different, then a spammer could spoof the From: address and then sign the message with a different signing domain. For example, suppose the From: field is security@paypal.com, but the signing domain is d= spammer.com. If *PayPal* says that it signs every message, then to a mail receiver the signature in the DKIM-Signature field will check out because a spammer could easily set up his own public/private key pair.


Query _adsp._domainkey.paypal.com
dkim=discardable
Query s1024._domainkey.spammer.com
for public key… message validates!

```
From: security@paypal.com
To: terry@tzink.com
Subject: How's it going?
Date: December 3, 2010
DKIM-Signature: v=1; a=rsa-sha256;
c=relaxed/relaxed; d=spammer.com;
s=s1024; t=1288392329; h=…; bh=...;
b=...;
```

Figure 1: What can happen if From: does not have to match the d= field.

Thus, if a domain wishes to use ADSP, then it cannot have mail sent by others on its behalf[2].

This leads to some interesting nuances. Consider the following sequence of events:

1. Extract the domain from the From: address.
2. Extract the domain from the d= field in the DKIM-Signature field.
3. If they are the same, then check the ADSP record.

What about the first time a sender transmits mail to a receiver and doesn't have a DKIM-Signature, but does have an ADSP record? In this case, the receiver would extract the domain in the From: field but would be unable to extract the domain from the d= field since the d= field doesn't exist. If this sender has an ADSP record that says 'discardable', the receiver would be unable to discard the message because it can't check the ADSP record as it has only one of the two required variables. This is important for the case of spoofing and phishing. If a phisher spoofs the From: address but does not include the DKIM-Signature, even if the spoofee has an ADSP record saying that unsigned mail should be discarded, this will not help the receiver. They don't know what action to take because an unsigned message is just that – an unsigned message. The receiver must rely on traditional spam filtering techniques.

Thus, from a logistical point of view, in order to make use of ADSP a receiver needs the following:

1. It has first to see an actual, signed message from the sender and look up its ADSP record.
2. It must remember that original ADSP record and compare it to future messages that purport to be from the sender.
3. It must periodically update its memory of the ADSP record to check that it hasn't changed.

From that point forward, all incoming messages for this particular sender are checked against this local copy and the ADSP policy enforced against it[3].

WHERE IS ADSP USEFUL

As we have seen, the requirement for the d= field to match the From: field adds a serious constraint to DKIM. What sorts of senders benefit from using ADSP?

The most useful ADSP record is the 'discardable' record because it is the only one that allows a receiver to combat the problem of spoofing. But in order to use ADSP with this record, only the domain itself can send out mail for its brand – the organization cannot outsource any of its mail to marketers (or rather, if it does, it adds a lot of risk by lending its reputation to a third party).

Furthermore, the organization must control all of its outbound servers and have a strong sense of ownership. For a small organization located in one city or state, this isn't difficult as uniform IT policies can be enforced across the organization. For global organizations with

[2] This is required by the protocol in RFC 5617 section 2.7; the domain in the d= field must match the domain in the From: field.

[3] DNS cache is one mechanism for achieving this, although it isn't the only one.

IT departments in multiple countries in multiple time zones, perhaps spread across multiple continents, this imposes logistical difficulties. It is non-trivial for an IT department to control and maintain physical resources spread across different locales. Servers get upgraded, personnel come and go, and software changes. People go on vacation and sometimes software upgrades are missed; if this happens and servers are misconfigured, then it means that legitimate mail coming from a single locale that is out of spec compared to the rest of the organization can end up being discarded. Global synchronization can be accomplished, but it takes a lot of time and effort. The larger an organization becomes, the tighter its security policies must be if it wants to use ADSP.

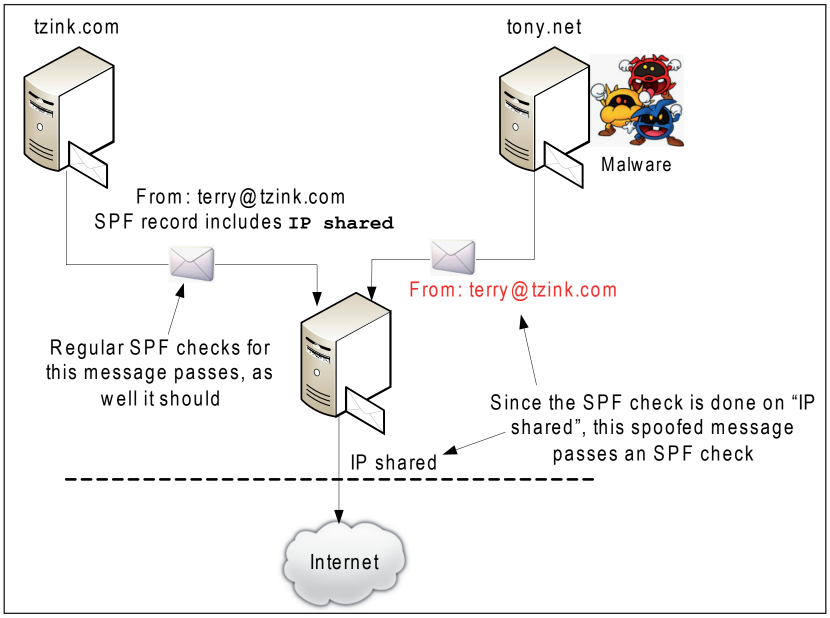

tzink.com
tony.net
Malware
From: terry@tzink.com
SPF record includes IP shared
From: terry@tzink.com
Regular SPF checks for this message passes, as well it should
Since the SPF check is done on “IP shared”, this spoofed message passes an SPF check
IP shared
Internet

Figure 2: Shared IP space and SPF weakness.

Financial institutions benefit directly from tight control over their brand identity and suffer greatly when it is abused. If a phisher spoofs a financial organization and succeeds in tricking the recipient into giving up their credentials, then that customer can lose funds. More and more these days, banks are starting to protect their users from phishing losses but this means that the bank (in many cases) absorbs the loss. However, it is not just banks that benefit from anti-spoofing; credit card companies also benefit. Credit card companies frequently advertise anti-fraud protection after the first $50 or so. Typically, they will fight with the vendors to get merchandise charges revoked, or they absorb the loss (or use insurance to offset the damages). But by ensuring that their identity is protected such that spam filters discard spoofed mail, they are reducing their vulnerability. No technology can wipe out the threat of spoofing, but organizations should be using whatever means they can to make it more difficult for phishers to trick their users.

Some of the most phished brands worldwide are *PayPal*, *HSBC*, *Bank of America*, and *eBay*[4]. These are all organizations that are strongly associated with money. They also send email communication to their user base directly instead of outsourcing it to a third party, and therefore, these organizations are excellent candidates for the implementation of DKIM and ADSP.

However, it’s not just money that attracts phishers and spoofers. Any organization that protects identity or is very popular and has a massive user base is a beneficiary of ADSP. For example, *Facebook* is one of the top 10 most phished brands. It has a user base of 500 million users. Malware authors target *Facebook* all the time and they do this because *Facebook* is so popular. The odds of being able to successfully trick at least one *Facebook* user with a spoofed email are fairly high. The odds shrink when ADSP is used because receivers can look up and see that *Facebook* signs (or could sign) all of its mail, and that mail that isn’t signed should be marked as spam.

In addition, organizations that have shared outbound IP space benefit from the use of ADSP. If one organization using the shared IP becomes infected with malware and starts to spoof another organization, then a recipient that only performs SPF checks cannot detect this as spoofing. The reason is that both organizations put the shared IP space into their SPF records, but a recipient cannot walk back through the Received headers to see which was the originating organization. Since both organizations use the same outbound IP space, the message will pass an SPF check even though it was spoofed. However, using ADSP, a receiver would be able to determine that the source of the message was not the organization it claimed to be.

Organizations that need to outsource their mail campaigns don’t benefit as much from ADSP. Without being able to assert ‘all’ or ‘discardable’, this forces organizations into the less useful[5] ‘unknown’ option. An airline, a small,

[4] http://www.ghacks.net/2010/02/02/avira-most-phished-brands-january-2010/.

[5] When I say ‘less useful’ I mean from an anti-spoofing viewpoint, not an authentication viewpoint.

independent bookstore, a dance studio, or a flower shop each might decide to use an email service provider to carry out their marketing campaigns. This drives down the costs for the organization in question, but it prevents it from using a strong assertion about whether or not it signs mail for its domain. However, in these cases, it is not as important as it is for some other types of organization – an independent bookstore does not protect its users' financial assets, nor does a flower shop know its customers' social security numbers. While they all might have credit numbers or similar, they do not have the footprint of a large financial institution or a social networking site.

It ultimately comes down to a cost/benefit ratio. When an organization becomes large enough to attract the attention of spammers and phishers, the chances of its customers being phished, and how much that will cost the organization, needs to be weighed against bringing its email campaigns in house.

DKIM, ADSP AND SPF

ADSP and SPF hard and soft fails accomplish similar things. How can they be used in conjunction with one another? What happens when we toss SenderID in there?

On the sending side, if an organization wants to use ADSP with 'all' or 'discardable', then it makes sense to complement it with SPF hard fails. The reason is that a hard fail implicitly states that you know all the IPs that you will ever send from. Ergo, this means that you have tight control over the mail that originates from your organization.

Since you know where your mail originates and want receivers to discard any mail from you that isn't signed, there is a very good chance that you also know where all of your outbound IP addresses are that relay mail to the Internet. By using 'discardable', you cannot have anyone send mail on your behalf. This means that no one can ever send mail as you from anything other than your own email servers, and therefore you should use SPF hard fails in your SPF record.

On the flip side, if you use SPF hard fails, should you also use 'all' or 'discard'? It's actually fairly complicated, so let's look at some possible combinations.

Case 1: ADSP pass (DKIM check validates) and SPF pass

This is the easy case, the message is 'doubly authenticated'. For a case such as this, you might treat the message the way you would treat any authenticated message – pass it through to the fast track of filtering, or simply collect statistics.

Case 2: ADSP fail (discardable) and SPF hard fail

In this case, the receiver should mark the message as spam. Both cases of identity checks are failing and therefore the message should be assumed to be spoofed. It could be a misconfiguration on the sender's side, but they are explicitly telling the world to reject the message.

Case 3: ADSP pass (DKIM check validates) and SPF hard fail or soft fail

This situation can occur when an organization has a new set of outbound IPs that they have just brought up but have not yet added them to their SPF record. However, they are signing with DKIM. This is possible if they have a role for their outbound servers (i.e. a pre-defined image that Operations needs to simply go and deploy), perhaps in a new data centre. In this case, the role for the servers already knows where to look up the private key and grab it to sign the mail. However, the SPF records have not yet been updated.

In this case, even though there is a contradiction between the permitted IPs in SPF records and the DKIM result, the results of a DKIM check are stronger than an SPF check, at least in the positive authentication case. Since DKIM is stronger, then the result of a DKIM pass should override the result of an SPF hard fail. What is important is that the mail originated from the organization, and DKIM asserts that.

Case 4 : ADSP fail (discardable) and SPF pass

This situation is similar to the above. An organization might start sending mail outside of its normal range of IP addresses but have previously added these new IP addresses to their SPF record (e.g. they had originally allocated more than they needed). However, what if they have not yet configured all of their new outbound mail servers to DKIM sign all of their mail?

Here, the ADSP fail should take priority. DKIM is stronger than SPF, but that's not the main reason a receiver should reject the mail. As discussed previously, an organization might share outbound IP space with other organizations if they are using another service as a relay (such as an ISP). However, they won't share their private key with other organizations. If another organization on the shared IP spoofs the MAIL FROM address of the first organization, this could result in an SPF pass. However, if the first organization uses an ADSP record, that check would then fail and a spoof would be detected. Since organizations that use ADSP are trying to protect against spoofing, and shared IP space is a weakness of SPF, then when an ADSP check fails and SPF passes, the mail should still be marked as spam or at least a very heavy weight applied to it.

Case 5: ADSP fail (all) and SPF hard fail or soft fail

When an organization asserts 'all' for its ADSP record, they are not issuing as strong a statement about whether or not they sign all of their mail. Should the receiver mark all mail from them without a DKIM signature as spam? Earlier, I said that a receiver should assign it a weight. I still stand by that, and if an SPF hard or soft fail occurs, then the two of them taken together should be assigned a heavier weight than if either one were to occur in isolation.

These are not the only scenarios but they do illustrate some possibilities for harmonizing the two technologies.

THE BOTTOM LINE

In this series on authentication, I have described the two major authentication technologies as well as the pros and cons of each. Ultimately, authentication is all about establishing identity and it is by no means limited to email; DNSSEC is another technology that is used to authenticate identity. One of the reasons why there is so much abuse on the Internet is because of the Internet's inherent anonymity. When it was originally designed, its creators did not foresee that it would become as popular as it did. Thus, when they built it, they simply wanted to get it up and running as soon as possible and this meant people could send mail as anyone. However, because the ability to actually transmit mail (or perform any Internet transaction) was limited to a small set of people, abuse was relatively rare. To put it one way, geeks all trust each other to act ethically.

As the Internet grew, its vulnerability increased because malicious players also started to use it. They account for a small proportion of Internet users but they are able to do a lot of damage because of the inherent insecurity of the underlying protocols of the Internet. If the creators had to do it all over again, it is unlikely that they would allow the anonymity that is permitted today and they would likely implement some mechanism of identity.

However, the Internet is not only a technological phenomenon, it is also a cultural one; in particular it reflects *western* values[6]. In the west, freedom of speech is one the most treasured values, and the Internet is viewed as a mode of communication. Thus, to the west, the Internet is seen as a mechanism of transmitting one's points of view, be it for entertainment, economic or *political* purposes.

That last one is important because one of the United States' values is the minimization of government lest it become too large. To US citizens (and, to a lesser extent, those of other western nations), the ability to speak out *against* repressive governments requires anonymity. The Internet is the perfect tool to communicate on a massive scale while still preserving that anonymity. Thus, if the technology sector ever wanted to end the anonymity of the Internet in order to end the scale of widespread abuse, they would encounter significant pushback from the political sector – both grassroots and organized movements. How are dissidents supposed to speak out against their governments without the safety of their anonymity?

The technology sector would claim that anonymity was not the original intent of the Internet. But regardless of whether it was intended, a lot of infrastructure and dependencies have been built up on top of anonymity such that its removal is virtually impossible. *That* is the current reality.

Will we ever see a cultural shift that allows us to value identity over anonymity? It's difficult to say. It's likely that with the deployment of IPv6, identity will be *required* by receivers in order to filter mail as IP blacklists will lose their effectiveness – the theoretical hiding spots for spamming IPs will become nearly infinite. Mail receivers will start pushing for everyone on IPv6 to start authenticating their mail so they can implement technical shortcuts for reputation filtering (perhaps *allow* instead of *reject*).

If email stays on IPv4 for many more years, and organizations send mail out of a common set of shared IPs, then identity will become even more valuable as receivers will insist that senders start signing their mail in order to avoid the collateral damage of blocking good senders who are forced to share an IP with bad senders. If that occurs, then technical requirements could force a shift in values.

In any case, the value of using SPF and DKIM, at this point, should be clear. On the sending side, by establishing your identity, you enable receivers to trust you and better facilitate communication between you and your recipients. On the receiving side, it allows you to differentiate organizations and apply different policies, and it can even help you detect spoofing. The time and effort spent implementing sender authentication can certainly outweigh its costs.

[6] This next part represents my personal views.

OPINION

TRANSITIONS: WELCOME TO THE NEW OLD WORLD

Andrew Lee
Independent researcher, UK

We stand on the cusp of a new decade (figuratively speaking, of course – actually standing on a cusp might be somewhat impractical): the second decade of this century. It should therefore come as no surprise that every journal, blog, television channel and newspaper is awash with retrospectives of the last ten years and jammed fuller than a Christmas/(insert your feast day of choice) postprandial stomach with predictions for the future. Humans seek out patterns (not signatures), and the cyclic positioning of the earth and other planets around our closest star is a pattern that provides endless opportunities for pontificating on the past and future.

Indeed, plenty has happened in the last 10 years. If the news reports are to be believed (and personal experience as a frequent flyer suggests they are), we have moved into a very different world from the one we inhabited in the year 2000.

In late 1999, many of us spent countless (ultimately wasted) hours supposedly preventing the end of the world. This was to be visited upon us in the form of the 'Y2K' problem – a global apocalypse which turned out to be not much more than an excuse to laugh at very solemn news reporters who had nothing to report for 12 hours, as we (Poor Bloody Infantry in tech support) sat in our offices playing solitaire, watching our servers not exploding, and rubbing our hands in glee in anticipation of large overtime payouts. But this, it seems, was only the harbinger of a series of *real* disasters which ultimately served to demonstrate that the world (like its media organs) is irretrievably bound together by instant communications, and relies to the point of absurdity on the Internet as a means to transmit, store and disseminate information.

This turbulent past decade has brought us through tragedy: the destruction by terrorists of New York's World Trade Center (amongst other terrorist acts), the deaths from tsunamis and floods in Asia that killed hundreds of thousands and displaced millions, hurricanes in the USA that killed thousands more, wars (not only in Iraq and Afghanistan), ethnic strife in Darfur, earthquakes, famines, fires, mining disasters, oil spills, and the collapse of both the dot-com industry and (more recently) almost the entire global economy. Whether this
is truly worse than, say, the years 1914–1918, when millions were killed in global wars, or 1665–1666, when Londoners faced plague followed by fire, or the countless wars, famines and disasters stretching back through history, is perhaps only a matter of perspective – there are few people alive who can say that they lived through those times.

We also saw the emergence of incredible new technologies (or 'magics', as Arthur C. Clarke might have described them), and amazing positive change. This was the decade that truly saw the coming of age of mobile computing (smartphones, netbooks, laptops and now *iPads*) as a phenomenon. Medical science saw the mapping of the human genome, the cloning of livestock, the creation of non-embryonic stem cells for research and the elimination of the Rinderpest 'cattle plague' virus. Engineers oversaw the construction of the world's tallest building – the 828m Burj Khalifa Tower in Dubai (although quite what it's *for* is another matter). Brazil and India experienced incredible economic growth, lifting millions out of poverty. And in entertainment, James Cameron made *Titanic* and *Avatar*, the two highest grossing movies of all time (though perhaps I should have added those to my negatives list, along with the potential cloning of celebrities and the technology that enables endless replays of yesterday's Christmas hits).

PUTTING CHANGE INTO CONTEXT

So why start like this? What does any of this have to do with security, and in particular anti-malware?

Well, for one thing, it seems important to put change into context. As so many have pointed out in retrospectives, the massive 'Internetization' (let's see if that one makes it into the *Oxford English Dictionary* in the next decade) of the world has meant that we have moved from hobbyist, slow-spreading viruses that we could inspect and analyse at our leisure, to a global swarm of malicious software used for criminal exploitation. We can clearly see from history, that wherever technology leads, crime will follow. The turning points can easily be identified: Loveletter, Slammer, Blaster, Bagle/Netsky, Storm, and most recently Stuxnet (to those who still care about naming, I apologize for using populist names).

Of course, crime has always existed online – *AOL* password stealers, trojan diallers and so on have been around since the early days – but the scale now is simply staggering, as is the convergence of the 'undesirable' elements of Internet life:

phishing, spam, spyware and malware, exploits and scams. The Internet has become a monster, and the criminals have successfully ridden its back as it has rampaged across the face of our civilization.

It is fashionable in some circles – unfortunately the ones in which many of us move – to discount the anti-malware industry as a hopelessly beleaguered dinosaur, still peddling its snake oils and balsams to the gullible and guileless user. The seismic events that caused the destruction of those ancient noble beasts have their parallels in the modern day, but despite the constant buffeting and an occasional fiery meteor, the AV industry prospers and indeed thrives. Conspiracy theorists may point to the past 'signatures is all we do', and highlight our failures – but they truly fail to appreciate the incredible innovation that has driven this industry forward.

How many products can claim to have such a broad reach, be updated so often, have such versatile functionality, and yet operate efficiently at the very lowest levels of modern operating systems (themselves unbelievably complex beasts)? Apart, of course, from the peddlers of rogue anti-virus for whom the process is as simple as writing a few more Javascripts and processing the income.

A DARK CORNER

Let us, then, imagine how a world without anti-malware might look. In fact, we do not need to stretch our imaginations too far. There is one dark corner of our universe where security still plays second fiddle to 'usability'; here the users remain blissfully unaware of the dangers that lie in wait for them, and they have no way of even knowing that such dangers exist. Of course, I'm talking about the wonderfully designed world of *Apple Mac*.

Here, it is still rare to find a voice that will openly admit that there is any problem, that there could ever be any problem, or that such a problem might be worth tackling. We, as anti-malware and security practitioners, know that from such gossamer we can stitch the emperor's new clothes; and we know that all that stands between the *Mac* user and the apocalyptic floods of malware so well known to the *Windows* user is economics – market share.

Why, when they are so successfully depleting the financial reserves of hapless *Windows* users, would an attacker bother with a lowly OS that has only around 5% of users? Indeed, anyone who bothers to track these things (as I have been doing for over 10 years), will know that as the market share of *Apple* has grown, so have attacks on users of *Apple*'s products – in exactly the same way that the growth of social networks such as *Facebook* and *Twitter* have given rise to malware for those facets of what might be termed the 'Internet operating system'.

So we could come to a point where the user might have no knowledge of any malicious exploitation, simply because there would be no 'sentient' program which might inform him of such activity. Just imagine the disaster if this were true today in the *Windows* world. In an alternative reality where this industry did not exist, every day corruption and fraud would exponentially increase the negative impact on our global economy, which would stifle take-up of technology and ultimately drive millions back towards poverty. Megalomania? I don't think so. The very real truth is that, on the whole, this industry does a thankless job in a situation in which the attacker constantly has the advantage – yet, it does that job unrelentingly, and some of the world's finest minds are bent towards ensuring that the levees do not break, that the missiles do not cause widespread destruction, and that the digital hurricanes do not leave millions at the mercy of the criminal elements that would so love to exploit them.

MAKING A DIFFERENCE

As we move into a new decade (or rather, as the earth continues to take its customary route around our central star) we surely will face new challenges. And, as we face those challenges, we will struggle to do all that we can to provide innovative protection, we will do all we can to stay one step ahead of the attackers, and we will try, against the odds, to stem the tide.

Sometimes we will fail for some of our users. Sometimes we will wish we could have done better. Sometimes we will be slated for it in the media. Sometimes people will stand up and say we're worthless. Sometimes we will wish that we could have a time machine, or a flying car, or that we could develop prescient powers greater than Nostradamus.

But at other times we *will* succeed. And, those will be the times when we justify our existence. Because a user somewhere didn't lose their credit card details; because a child wasn't exposed to pornography; because a factory didn't get shut down by malware and the workers kept their jobs; because Grandma still got that email with pictures of her new baby granddaughter.

The fact that sometimes we will miss our targets, or fail to protect where we might have done, makes us no different from any army in the world – but for all that, we know that we still make a difference. So, here's to the next decade, where we will look back and wish for the quiet halcyon days of the 00s – or at least hope we won't have to balance precariously on any more cusps.

ANECDOTE

'HELLO, I'M FROM WINDOWS AND I'M HERE TO HELP YOU'

Craig Johnston
Cybercrime researcher, Australia

A few weeks ago I received a number of queries from friends who had received phone calls from a man telling them that they had viruses on their computers. I told them that this was a scam and advised them not to have anything to do with anyone who calls and makes such claims.

ROUND ONE

However, one evening recently I received an unsolicited phone call at home from a man with a heavy Indian accent. He informed me he was 'from Windows in Sydney' (when I questioned him further he said he was from *Microsoft* and gave me the company's correct Sydney address). He told me that my computer had been flagged as being infected with viruses and that he was calling to help me out.

Of course I realized this was a scam, but I was very interested to learn how the scammers operated, so I played along.

The caller took me step by step through the process of opening up the Event Viewer on my home PC and told me where to look once there. He asked if I could see any error, alert or warning messages displayed – which of course I could. He told me that this confirmed that my computer was infected with viruses and that he would help me fix the problem. When I asked him why he was doing this, he said he was from *Microsoft* and that its staff had a duty to help people when they could see a computer was infected.

Next, he asked if my computer was running a little slower than it used to, and of course I said it was. He presented this as more evidence of the virus infection on the computer. He then tried to get me to log onto a website that would give him remote access to my computer to enable him to help me.

Of course, I wasn't too keen on giving him control of my system, so I hung up the phone. Two minutes later, he called back and continued to try to persuade me to allow him to take control of my system.

After about five minutes of me trying to get the caller to prove that he was actually in Sydney (by asking what the weather had been like here that morning – it's easy to look up a weather forecast for anywhere in the world, but harder to find very recent weather history) he eventually gave up and said he couldn't help me. The incident was interesting, but somewhat predictable.

ROUND TWO

Four days later, I received another call from another man with an Indian accent, who spouted the exact same lines. I strung him along for a few minutes before I got fed up of the whole exercise and told him that it was all a scam and accused him of preying on people's naïvety and abusing their trust. He asked 'So you think this is a scam?', to which I replied 'I know it's a scam!', and he simply admitted, 'Yes, it is a scam'.

For the next 15 minutes we had a very interesting conversation. The caller was more than happy to answer my questions about the group's modus operandi and admitted that his job was to cause confusion and fear in the victim, while posing as a trusted advisor, so that he could sell the victim a product. The product he said the group were selling was *Registry Mechanic* – which is a *Windows* registry optimization tool from *PC Tools* (owned by *Symantec*). While the caller admitted that the methods used to convince the 'customer' were dodgy, he was keen to assure me that the product being sold was legitimate and that it would benefit the customer.

I think that this man genuinely believed that he and his colleagues were helping people out. When I asked him if *Registry Mechanic* was an anti-virus product, he replied that it was, and told me that it would protect users from malware.

I found the conversation very interesting. The guy was more than happy to answer my questions, even though at one point I told him that I worked in the field of cybercrime research. He told me that he was based in Calcutta and that he and his colleagues had made a lot of money by targeting people in Australia recently. As we said our goodbyes, he even told me that he'd enjoyed our chat.

CONCERNS

These two related events raised some concerns in my mind. They are, in no particular order:

- Given the queries I'd had from friends, and the fact that I received two similar calls in the space of a few days, it seems that these guys were hitting the Sydney area very hard.
- I'm certain that the bogus callers would be very successful with the method they were using. There are plenty of people who are naïve and/or ignorant when it comes to computers. If a nice gentleman (apparently) from *Microsoft* calls them to help them find evidence of a virus on their computer, then offers to take over their computer and clean it up, then sell them a product to protect them in the future – and install it on their

system for them – many people would be grateful and even happy to pay a small fee for the assistance.

- The claim that *Registry Mechanic* is an anti-virus product that will protect users against malware is simply wrong. The product is a legitimate one, and it does its job very well, but it is not designed to provide full protection against malware.
- How immoral (and illegal) is it to use fear, uncertainty and doubt (FUD) and scammer-type techniques to sell what is essentially a legitimate product (even if it is not a good solution to the supposed threat)?
- Is there a reseller of *Registry Mechanic* in India who is doing a lot of business selling to customers in Australia, and if so, should someone be pulling the plug on them and their questionable operations? I understand that *Symantec* is looking into it. (Having said that, the product that was being sold may well have been a copied, hacked or outdated version of the genuine product, and it is most likely that the callers were not, in fact, genuine resellers of the product.)

CONCLUSION

It would be relatively easy to tell people simply to ignore any and all unsolicited contact from people informing them that they have spotted a malware infection on their computer. However, on 1 December this year all the big ISPs in Australia signed up to become 'icode compliant'. The icode[1] is a national voluntary code of practice which involves ISPs contacting customers that have been identified as being infected with malware to inform them that they may be quarantined or disconnected from the Internet until they clean their computer up. The ISPs will direct the infected users to a website (http://www.icode.net.au) which tells them how to avoid malware infections, how to detect and remove malware, and how to get professional help in cleaning up their computer.

So the ISPs will soon be contacting people out of the blue and telling them that their computer has been identified as having a malware infection, then offering help to clean up their computer. It goes a little like this: 'Hello, I'm from [Big ISP], and I'm here to help you!'

Hmm, sound familiar...?

(Fortunately, when the ISPs make their calls they will encourage the customer to verify the ISP's identity by calling them back on a previously published and publicly available phone number.)

[1] http://www.iia.net.au/index.php/all-members/869-get-ready-for-icode-in-force-1-december-2010.html.

CALL FOR PAPERS

VB2011 BARCELONA

Virus Bulletin is seeking submissions from those wishing to present papers at VB2011, which will take place 5–7 October 2011 at the Hesperia Tower hotel, Barcelona, Spain.

The conference will include a programme of 30-minute presentations running in two concurrent streams: Technical and Corporate.

Submissions are invited on all subjects relevant to anti-malware and anti-spam. In particular, *VB* welcomes the submission of papers that will provide delegates with ideas, advice and/or practical techniques, and encourages presentations that include practical demonstrations of techniques or new technologies.

A list of topics suggested by the attendees of VB2010 can be found at http://www.virusbtn.com/conference/vb2011/call/. However, please note that this list is not exhaustive, and the selection committee will consider papers on these and any other anti-malware and anti-spam related subjects.

SUBMITTING A PROPOSAL

The deadline for submission of proposals is **Friday 11 March 2011**. Abstracts should be submitted via our online abstract submission system. You will need to include:

- An abstract of approximately 200 words outlining the proposed paper and including five key points that you intend the paper to cover.
- Full contact details.
- An indication of whether the paper is intended for the technical or corporate stream.

The abstract submission form can be found at http://www.virusbtn.com/conference/abstracts/.

One presenter per selected paper will be offered a complimentary conference registration, while co-authors will be offered registration at a 50% reduced rate (up to a maximum of two co-authors). *VB* regrets that it is not able to assist with speakers' travel and accommodation costs.

Authors are advised that, should their paper be selected for the conference programme, they will be expected to provide a full paper for inclusion in the VB2011 Conference Proceedings as well as a 30-minute presentation at VB2011. The deadline for submission of the completed papers will be Monday 6 June 2011, and potential speakers must be available to present their papers in Barcelona between 5 and 7 October 2011.

Any queries should be addressed to editor@virusbtn.com.

PRODUCT REVIEW

ESCAN INTERNET SECURITY 11

John Hawes

We have had a chance to check out the latest *eScan* product in the last couple of VB100 comparative reviews, and have been intrigued by both the snazzy look of the new interface and the promise of a wealth of additional features. In this review we'll be taking a more thorough look at version 11 of *MicroWorld*'s *eScan Internet Security* to find out how well the funky styling works and what more it has to offer.

PRODUCT RANGE AND WEB PRESENCE

MicroWorld Technologies is a veteran player in the anti-malware world, having been founded in 1993. The company first submitted a product for VB100 testing in 2003, and since then it has become one of our most regular participants – providing support for every platform we test on and keen to face up to the challenge of our certification scheme month after month. The company's test record has been exemplary, with a solid, if not quite unbroken run of VB100 passes – even weathering the trying switch from an OEM engine to all-in-house technology with only the slightest dip in performance.

According to *MicroWorld*'s website, escanav.com, the company is incorporated in the US, but is Indian in origin and has a large presence there, as well as offices in Germany, South Africa and Malaysia. From this base is produced a complete set of solutions, ranging from home-user to enterprise level and including a selection of dedicated server and gateway products – there is even a home-user *Linux* solution. The company website focuses heavily on the product range and its promotion, with the most prominent areas given over to information on the products, trial downloads, licensing and partner resellers. The bottom of the home page is adorned with a thorough selection of certification logos, including the VB100 award.

There is, of course, some more general content, including a blog, information on the latest major threats, and a news section, which seems to focus on company news rather than threat-related information.

Also grabbing interest on the home page were links marked 'Wiki', which lead to a sub-site featuring a surprisingly well stocked knowledgebase. This is presented in FAQ format and includes some interesting general tips and advice as well as solution-specific information. It is here, we later discovered, that the help system for the product range is hosted. This appeared to be very thorough and clearly laid out with plenty of illustrations, but it takes an almost entirely 'describe-what-each-button-does' approach, rather than the generally more useful task-oriented approach (however, that may have been available elsewhere in the large wiki area).

A 'Forum' section is similarly full of questions, most of which appear to have been answered rapidly, although users with more specific queries seem to be redirected in most cases to a live chat support system. On visiting this and posing a random question, we wondered at first if the system was automated, given the stock 'welcome' and 'please provide licence info' responses, but a friendly plea for quick help brought a more human-sounding reply, with at least some effort made to resolve the problem. Phone-based support is also available, but this is time-limited and regional, and customers are encouraged to use email or live chat where possible.

One other thing of note on the website are the links scattered around all over the place to the 'eScan Anti-Virus Toolkit' – something which also does well on a number of popular software download sites. This is a simple on-demand scanner tool with cleaning and command-line control as well as GUI, which can be used on systems that are not running the full product; its set-up and usage are pretty straightforward, and it's a pretty handy, powerful little tool.

Of course, the part of the website we were most interested in was the download page for the latest full product, and from here we initiated a download via a proprietary download manager solution, which dealt well with losing connection repeatedly. When allowed to complete properly the download measured close to 150MB, but it took only a few minutes on a reasonably fast connection.

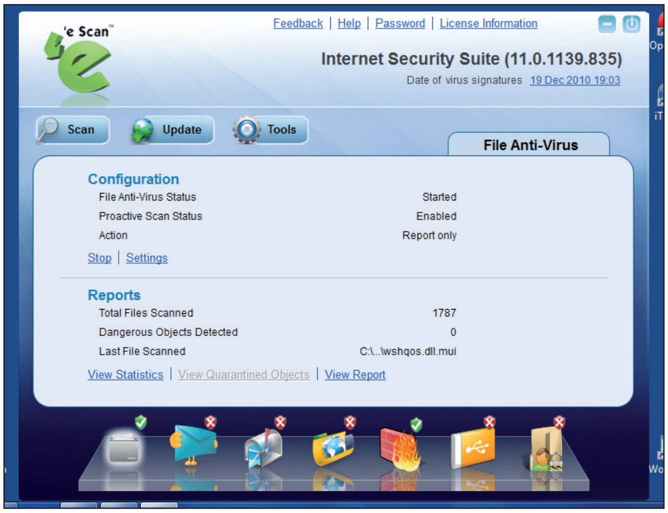
SET-UP AND CONFIGURATION

We tried the product on a selection of systems. The hardware requirements are fairly forgiving, with only 256MB of RAM required (although more is recommended). Support is still in place for *Windows 2000* as well as the latest x64 versions of *Windows 7*, so all but the most ancient of systems ought to be able to run the product. To try this out, we ran it on netbooks and wheezy old systems as well as more standard hardware.

Given the range of features we knew to be included in the product, we found the installation process fairly straightforward. It offers to disable the *Windows Firewall*, if it is running, in order to replace it with its own, but other than that there seemed to be little in the way of initial configuration or options to deal with, and after a reboot it was apparently all set and ready to go.

As mentioned previously (and repeatedly in the company's literature), the interface design is a bit of a departure from the norm – but not so extreme as to confuse anyone. There seem to be two standard approaches to anti-malware GUI layout, the ‘traditional’ style, with a menu down the left-hand side covering the various modules, and the currently modish style of having several large icons arrayed across the centre of the main window covering the main functions. There are a few radical departures from this pattern, some of which lean towards the opaque, but *eScan* manages to shake things up a bit without wandering too far from familiar patterns. The most notable thing here is the ‘dock’ at the bottom of the GUI, on which icons for all the main areas are presented. The colourful icons enlarge on rollover in the *Mac* style and it is reasonably easy to decipher what each is intended to denote. The rest of the GUI is much simpler, with a plain tab for each module providing start/stop buttons, information on status, reporting of events and so on, in plain, unadorned text.

In each of these areas there is a ‘Settings’ button, which leads to configuration controls for each module. These are fairly plain and unfussy, text-based and generally provided in exhaustive detail; just about everywhere we looked, there were more opportunities for fine-tuning to achieve just about any combination of settings. The wording of most was clear and lucid, and although in more sophisticated areas it becomes increasingly difficult to explain concepts concisely in non-technical language, we felt that the fine-tuning options would be fairly accessible to the interested, but non-expert user. Something that was lacking was context-based help rather than the generic link to the online wiki system at the top of the main interface. Being able to click straight through to a full description and guidance relating to the particular area in which one has hit a problem can be invaluable – and particularly good for encouraging those who do not have a complete understanding of what they are dealing with to take an interest, learn the ropes and take some responsibility for their own security. For many, browsing and searching through a complete help system is likely to be too much like hard work, and the opportunity to turn a moment’s interest into an educational experience for the user is easily lost.


'e Scan™
Feedback | Help | Password | License Information
Scan
Update
Tools
File Anti-Virus
Stop | Settings
View Statistics | View Quarantined Objects | View Report

Internet Security Suite (11.0.1139.835)

Date of virus signatures 19 Dec 2010 19:03

Configuration

File Anti-Virus Status	Started
Proactive Scan Status	Enabled
Action	Report only

Reports

Total Files Scanned	1787
Dangerous Objects Detected	0
Last File Scanned	C:\...\wshqos.dll.mui

With such a wide range of ground to cover, we’ll leave a closer look at the configuration system for later, after first taking a quick look at our main area of interest.

ANTI-MALWARE PROTECTION

In its earlier years *eScan* was powered by the *Kaspersky* engine, with some in-house-developed extras on top. In our tests the product’s results routinely followed those achieved by *Kaspersky Lab*’s own solutions fairly closely. When *MicroWorld* first announced it was planning to drop the OEM engine to fly solo, our first reaction was one of surprise and trepidation – having seen several other products try to go it alone in this way, with generally disastrous results, we were highly sceptical that *eScan* would be able to retain its VB100 certified status, let alone maintain its previous solid record and regular high scores. After a brief wobble and a slight drop in detection rates over the following months, however, *eScan* mounted a steady and apparently unstoppable attack on our test sets, with coverage growing more and more thorough each month. False positives have been an occasional problem, but rarely a major one, and its superb detection rates routinely place it in the high-achiever cluster on our RAP quadrants.

With plenty of testing history behind us, we ran a few additional detection tests using some newer and older sample sets that are not routinely used in VB100 testing, as well as some extra clean software. We saw a pretty similar pattern of excellent coverage and minimal false alarms. Indeed, this latest version of the product boasts a whitelisting system which, the company claims, should entirely eliminate any risk of false alarms on important operating system files (the type which generally cause the most havoc and distress). This system certainly appeared to have been working well in the last few VB100 tests, with no problems encountered despite some major expansion of our clean sets, and even when run over expanded sets containing large quantities of unchecked material we saw no evidence of false alarms.

The product’s scanning speed is not super-fast and, as we observed in the most recent comparative (see *VB*, December 2010, p.27), can be a little unpredictable – with some scans taking notably more or less time than the same job run just moments earlier. Running a standard basic scan – covering only memory, registry and so on – took a few minutes at most, but a thorough check of a large, well-stocked system can take several hours and is usually best left for an overnight run when the machine is not in use (of course, a comprehensive and well-designed scheduler system is available to simplify and automate this). That said, where speed really counts is in on-access mode, as this is what will affect the running of the machine. Here, all seems pretty good, with measures taken in the last test proving very low indeed, both in terms of lag times imposed on accessing files and in the amount of RAM and CPU used. Some further tests in this area showed consistently light system impact when running a variety of common tasks.

In addition to the standard protection offered by this static file scanning and monitoring, and the additional features we'll look at in the next section, there are a couple more elements covered in the main 'File Anti-Virus' section of the interface which merit a mention here. Buried on the second tab of the settings dialog for this section is a checkbox marked 'Enable Proactive Scan', which, according to the description in the help pages, applies an extra layer of heuristics for picking up on suspicious activities at run-time, including monitoring registry changes and so on. This option appears to be disabled by default, but when trying some of the few malware samples we found which were not picked up by the standard scanner, some additional protection was seen here – almost all samples performed some activities which caused them to be spotted and warned about. This additional protection was counterbalanced by a slightly increased tendency to warn about legitimate items performing unusual actions though – so this option is perhaps best reserved for experts and the extremely paranoid.

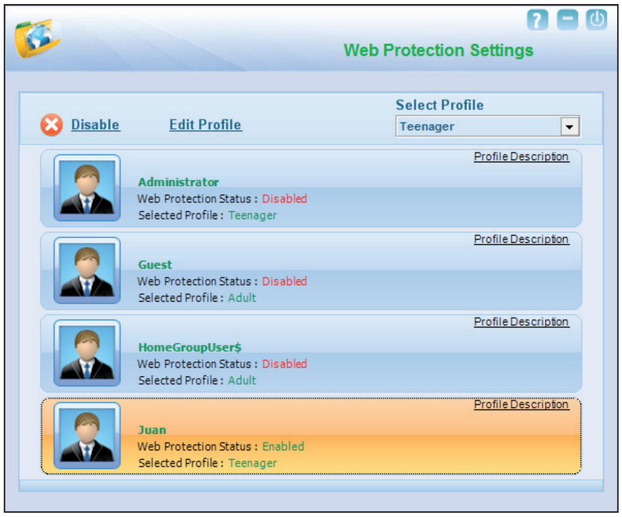
The final item is labelled 'Folder Protection', and locks down specified folders, preventing any changes to the files stored in them. This seems like a nifty idea, allowing one to corral one's most important items and keep them safe from damage. A brief test showed this feature to work nicely against a range of attempts to subvert it.

OTHER FEATURES

As if this solid level of protection at the file level was not enough, the suite includes a comprehensive range of additional layers of protection. The second and third tabs on the GUI dock are mail anti-virus and anti-spam, their icons differentiated by the presence of a mailbox as well as an envelope in the anti-spam one. The mail scanning is pretty straightforward, defaulting to blocking a wide range of executable file formats as well as some specific filenames commonly used by malware. Additional options include checking inside compressed files, looking out for attachments with multiple file extensions and watching out for attempts to exploit vulnerabilities in mail clients – there is an option to block all HTML emails containing scripts. There is also a surprise extra in the form of a mail backup system, which can be set to archive all mails and attachments to a selected location. All of this is clearly laid out and generally pretty self-explanatory.

The anti-spam settings are, of course, a little more complex: at the most basic level they can be set to add warnings to various parts of mails detected as spam, to check for certain character sets or tell-tale spammy characteristics, to check outgoing messages, forwards and replies (all disabled by default), and even to block or whitelist messages containing phrases specified on a user-configurable list. Somewhat oddly, the main tab is labelled 'Advanced', but also has a button marked 'Advanced' which leads to the truly advanced settings. Here, spam-filtering geeks can really let themselves go, tweaking which parts of mails to include in checks, which RBL servers to use and whether or not to check SPF compliance and SURBLs. There is also the mention here of *eScan*'s 'Non-Intrusive Learning Pattern', a Bayesian filtering system which the help system describes as 'revolutionary', without giving away a great deal of information about what it does or how it works. Our anti-spam testing system is currently not geared up to test desktop solutions, but we hope to introduce this facility soon, and will add *eScan*'s offering to the list of solutions to look at, to find out how this system fares.

The middle icon on the dock is tagged 'Web Protection', and consists mainly of parental controls. These are nicely laid out and fairly simple to use, with a set of levels which can be applied on a per-user basis, including 'adult', 'teen', 'adolescent' and 'walled garden'. Each is infinitely configurable, with enormously long lists of both whitelisted sites and keywords associated with half a dozen categories of potentially unwanted content. The


Web Protection Settings

Select Profile: Teenager

Profile Description
Administrator
Web Protection Status : Disabled
Selected Profile : Teenager

Profile Description
Guest
Web Protection Status : Disabled
Selected Profile : Adult

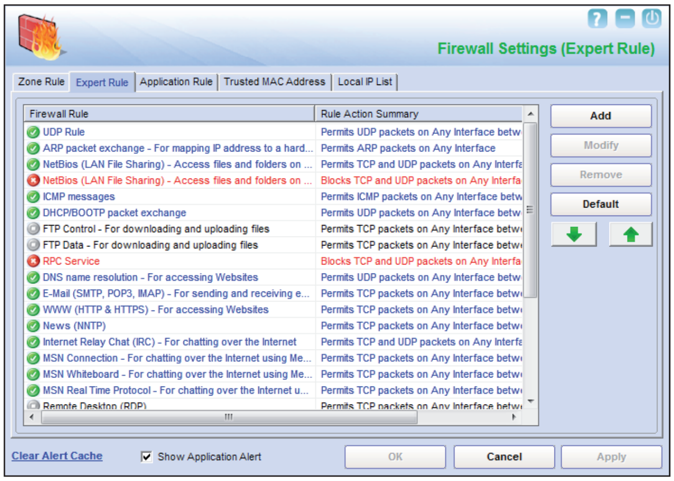
Profile Description
HomeGroupUser$
Web Protection Status : Disabled
Selected Profile : Adult

Profile Description
Juan
Web Protection Status : Enabled
Selected Profile : Teenager

keywords list includes the names of a bewildering range of beverages in the 'alcohol' section, and such a wide selection of military terms and weaponry listed under 'violence' that it may well hamper any kind of historical research if applied injudiciously. The process of adding keywords and adjusting settings is clear and simple, and it all seems to work very nicely. Attempting to access a site featuring one of the blacklisted words triggers a warning pop-up and a replacement web page. Users should note that when switching on this system they also need to set a master password via the link at the top of the interface. A helpful tool tip on the alert pop-up lets on that if this step is forgotten, the password 'admin' will grant access to any and all content.

One other section in this area is labelled 'Phishing Filter' – in most other areas configuration is all-encompassing, but here, other than a simple start/stop button, no further control or information is given, even in the help system.

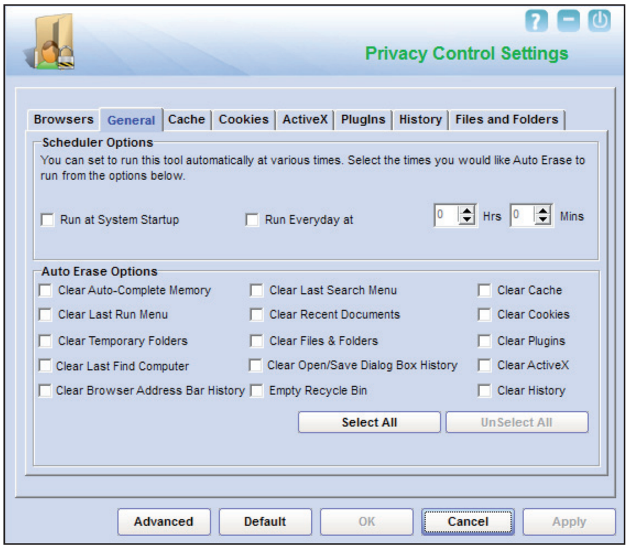
Moving swiftly onto the firewall, this is a fairly standard implementation, with a scale of protection ranging from allow all to block all, via 'Limited filter' (only applying a basic set of rules) and 'Interactive filter' (prompting for user decisions when something is not defined). The basic settings seem fairly well defined and work reasonably well, while the interactive mode is much more thorough but presents a lot of pop-ups if not configured carefully. Fortunately, a good depth of configuration is provided once again, and quite some effort has been made to assist less skilled users. Rather than just providing stark and unhelpful acronyms, descriptions of what each protocol is needed for are given, and proper sentences are used in the rule descriptions, which should make things a little easier for technophobes (although some understanding of networking basics will be required to operate it properly). The tweaking and adding of rules is a nicely presented and smooth process.



Firewall Settings (Expert Rule)

Firewall Rule	Rule Action Summary
UDP Rule	Permits UDP packets on Any Interface betw
ARP packet exchange - For mapping IP address to a hard...	Permits ARP packets on Any Interface
NetBios (LAN File Sharing) - Access files and folders on ...	Permits TCP and UDP packets on Any Interfa
NetBios (LAN File Sharing) - Access files and folders on ...	Blocks TCP and UDP packets on Any Interfa
ICMP messages	Permits ICMP packets on Any Interface betw
DHCP/BOOTP packet exchange	Permits UDP packets on Any Interface betw
FTP Control - For downloading and uploading files	Permits TCP packets on Any Interface betw
FTP Data - For downloading and uploading files	Permits TCP packets on Any Interface betw
RPC Service	Blocks TCP and UDP packets on Any Interfa
DNS name resolution - For accessing Websites	Permits UDP packets on Any Interface betw
E-Mail (SMTP, POP3, IMAP) - For sending and receiving e...	Permits TCP packets on Any Interface betw
WWW (HTTP & HTTPS) - For accessing Websites	Permits TCP packets on Any Interface betw
News (NNTP)	Permits TCP packets on Any Interface betw
Internet Relay Chat (IRC) - For chatting over the Internet	Permits TCP and UDP packets on Any Interfa
MSN Connection - For chatting over the Internet using Me...	Permits TCP packets on Any Interface betw
MSN Whiteboard - For chatting over the Internet using Me...	Permits TCP packets on Any Interface betw
MSN Real Time Protocol - For chatting over the Internet u...	Permits TCP packets on Any Interface betw
Remote Desktop (RDP)	Permits TCP packets on Any Interface betw

Clear Alert Cache ☑ Show Application Alert



Privacy Control Settings

Scheduler Options

You can set to run this tool automatically at various times. Select the times you would like Auto Erase to run from the options below.

☐ Run at System Startup ☐ Run Everyday at 0 Hrs 0 Mins

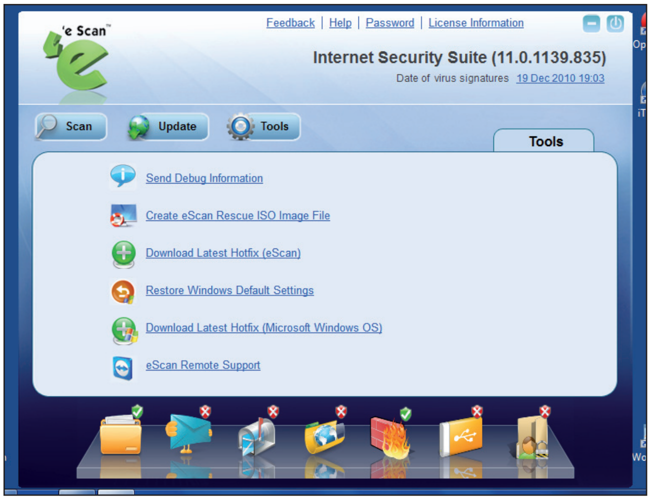
Auto Erase Options

☐ Clear Auto-Complete Memory ☐ Clear Last Search Menu ☐ Clear Cache
☐ Clear Last Run Menu ☐ Clear Recent Documents ☐ Clear Cookies
☐ Clear Temporary Folders ☐ Clear Files & Folders ☐ Clear Plugins
☐ Clear Last Find Computer ☐ Clear Open/Save Dialog Box History ☐ Clear ActiveX
☐ Clear Browser Address Bar History ☐ Empty Recycle Bin ☐ Clear History

The 'Endpoint Security' section covers application and USB device control. The USB section scans flash drives for malware and disables autoplay; it can also be set to block or insist on a password for all USB devices, and to whitelist known devices. Application control is considerably more complex, with a monster list of known applications grouped into categories and blockable on a per-application basis. The categories cover gaming, messenger, media player and P2P software, extending the already thorough parental controls settings to a business-ready level.

The final major sub-section is 'Privacy Control', which focuses on browsers and can clear history, caches, cookies etc. on demand or on a schedule. It can also be used to browse history and caches and delete individual items, and to clear *Windows* recycle bins and recently accessed lists. However, it seems to cover only *Internet Explorer* and *Firefox*, with no mention of any of the other browsers – including *Chrome*, *Opera* and *Safari* – installed on our test system. Once again, the controls are clear, simple and comprehensive, and most of the bases seem to be covered to ensure that whatever you've been up to can't be tracked – ideal for international secret agents and the paranoid.

That's all for the main set of modules (note that each has a button to create a report of all activity noted or performed, for total traceability), but of course, not all the suite has to offer. A button at the top of the interface, next to the 'Help' button, is labelled 'Tools' and provides a screenful of further items. Several of these are related to product maintenance, including generating debug information to help support issues, downloading patches, and allowing a support engineer remote access. There is also the option to create a boot CD to provide a clean environment for disinfection; this is an increasingly common offering, and something



Internet Security Suite (11.0.1139.835)

Date of virus signatures 19 Dec 2010 19:03

eScan
Feedback | Help | Password | License Information
Scan
Update
Tools
Tools
Send Debug Information
Create eScan Rescue ISO Image File
Download Latest Hotfix (eScan)
Restore Windows Default Settings
Download Latest Hotfix (Microsoft Windows OS)
eScan Remote Support

we've been planning to introduce into our comparative testing. In this case, it is apparently *Windows*-based, and is provided as part of the installation media for those purchasing boxed copies of the product.

A couple of items in the 'Tools' tab are a little different however – one downloads and applies the latest *Windows* updates, using the company's own download manager tool. This seems to be provided as an alternative to the standard *Windows Update* process, but appears to be lacking in configurability or scheduling tools. A similarly basic item is labelled 'Restore Windows default settings', and really does just that – we clicked on the link expecting to be presented with some options and so on, but were surprised to see it start running the normal malware scan dialog, apparently trolling through the system looking for changes which could have been made by malware, and reverting them to their default values.

Leaving the interface, the final few items are found on the context menu from the system tray icon. Included here, as well as the option to open the main 'console' and other standard items such as the 'about' information, is a network activity monitor – a little application that lists everything connected to the network and provides some basic details on each. Right-clicking any process gives the option to kill it or look up more information on the *eScan* website, where most common items are explained in some detail. A similar application is labelled 'System Information'. This provides detailed data on the host machine – much of which can be found in the *Windows Computer Management* console on most platforms, but here it is presented in a nice, orderly fashion with plenty of helpful, colourful icons. The final item is a virtual keyboard, designed to defeat any keyloggers which may have sneaked past the battery of protective layers. This is a fairly simple, but by no means unappealing little gadget.

CONCLUSIONS

After a rather exhausting trawl through the vast wealth of add-ons, our final impressions of *eScan 11* are generally pretty favourable. It maintains, and even improves upon the excellent levels of detection and proactive protection provided by previous versions, and adds a little extra sparkle in the form of the funky dock gizmo (and there's an extra bit of show business when closing windows, as they twirl away in a rather unnecessary spiral form). Beyond the shiny front page, things are much more businesslike and functional, with some of the best configuration and controls we have seen in any product lately, both in terms of the depth of fine-tuning available and in their clarity and simplicity. The range of extras is truly impressive, with the parental controls particularly complete and thorough, especially when supplemented by the application control module – combined, they add up to a system suitable for both worried parents and corporate policy enforcers.

It's not all perfect of course. The help system does let the product down a little; a proper contextual system providing instant and targeted advice would be much better, and some kind of task-based guidance in addition to the current list of options is a must. There were also places where the product seemed to lack internal consistency, with some of the extras appearing tacked on and not properly integrated into the main suite. Some features also seemed short on description or controls, particularly compared to the thoroughness displayed elsewhere, and occasionally we were taken by surprise when the click of a button kicked off an activity rather than taking us to the expected set-up stages. When really treating the product badly, thrashing at the buttons and overloading systems, we managed to make it 'hang' a couple of times (at least, by the *Windows 7* definition – in reality, it just had to calm itself down for a few seconds).

These are fairly minor gripes though, mainly to do with usability, and in general it does an excellent job in this area, making some serious tools accessible to all. On top of the superb protection built in, this makes it a solid and impressive suite all round.

Technical details

MicroWorld Technology's *eScan Internet Security Suite 11* was tested on:

Intel Pentium 4 1.6 GHz, 512MB RAM, running *Microsoft Windows XP Professional SP2*.

AMD Athlon64 3800+ dual core, 4GB RAM, running *Microsoft Windows XP Professional SP3* and *Windows 7 Professional*.

Intel Atom 1.6GHz netbook, 2GB RAM, running *Microsoft Windows XP Professional SP3*.

COMPARATIVE REVIEW

VBSPAM COMPARATIVE JANUARY 2011

Martijn Grooten

2011 started off with some good news as in the final weeks of December, the amount of spam circulating globally decreased significantly – adding to the general decline in spam volumes seen during the second half of 2010.

However, it would be wrong to suggest that spam is going away any time soon – or even that it will cease to be a problem in the future. Spammers are already finding ways to make up for the decrease in spam quantity – for instance by individually targeting their victims. Indeed, several cases of spear-phishing have recently made the news.

Thus in 2011, organizations will still need solutions to deal with massive streams of unsolicited emails and *VB* will continue to test such solutions in the VBSpam certification scheme.

Readers will notice that among the 18 products that have earned a VBSpam award in this month's review, the differences in performance are often very small. To discover which of these products works best for a particular organization, running a trial might be useful (most vendors offer this possibility) – such a trial would also provide the opportunity to evaluate a product's usability and additional features. The Messaging Anti-Abuse Working Group (MAAWG) has produced a useful document that explains how organizations can conduct such tests in-house and, in general, how to evaluate email anti-abuse products. The document can be found at https://www.maawg.org/system/files/news/MAAWG_Anti-Abuse_Product_Evaluation_BCP.pdf.

THE TEST SET-UP

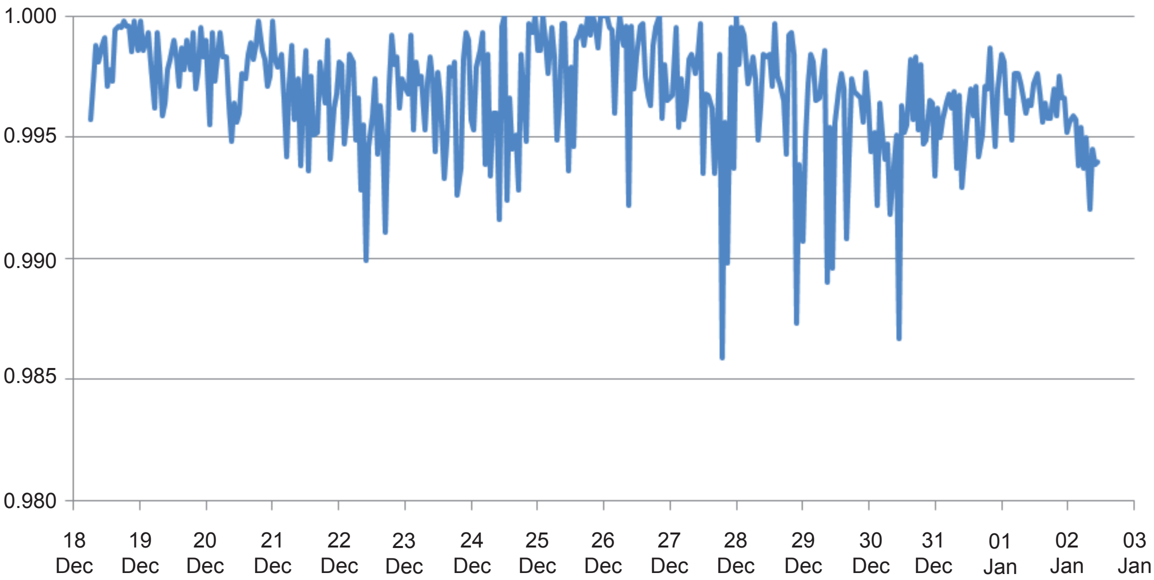
The VBSpam test methodology can be found at http://www.virusbtn.com/vbspam/methodology/. As usual, email was sent to the products in parallel and in real time, and products were given the option to block email pre-DATA. Four products chose to make use of this option.

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.

To compare the products, we calculate a 'final score', which is defined as the spam catch (SC) rate minus five times the false positive (FP) rate. Products earn VBSpam certification if this value is at least 97:

$$SC - (5 \times FP) \geq 97$$

Note that this is different from the formula used in previous tests, where the weight of the false positives was three and the threshold was 96.



Average catch rate of all full solutions throughout the test.

THE EMAIL CORPUS

The test ran for just over 15 consecutive days, from around 6am GMT on Saturday 18 December 2010 until midday on Sunday 2 January 2011.

The corpus contained 91,384 emails, 89,027 of which were spam. Of these spam emails, 32,609 were provided by *Project Honey Pot* and the other 56,418 were provided by *Abusix*; in both cases, the messages were relayed in real time, as were the 2,357 legitimate emails. As before, the legitimate emails were sent in a number of languages to represent an international mail stream.

The graph on the previous page shows the average catch rate of all full solutions throughout the test. As one can see, the decline in global spam volume coincided with a small decline in the average product's performance.

In the previous review we looked at the geographical origin of the spam messages and compared those of the full corpus to those spam messages missed by at least two solutions. We saw, for instance, that spam from the US appears to be relatively easy to filter, while spam from various Asian countries is more likely to make it to users' inboxes.

This time, we looked at the content of the messages, based on the MIME type of the message body. We distinguished five categories: messages with a body consisting of plain text; those with a pure HTML body; those with both plain text and HTML in the body; those with one or more embedded images; and other kinds of messages, including DSNs, messages with attached documents and those with an unclear and possibly broken MIME-structure.

The table below shows the relative occurrence of the categories; left among the full spam corpus, right among those messages missed by at least two solutions. As in the previous test, the latter concerned slightly fewer than 1 in 60 spam messages.

An interesting conclusion is that plain text messages – which in theory are the easiest for a content filter to scan – are significantly more likely to cause problems for spam filters than other message types. On the other hand, essages containing HTML in the body – especially those with a pure HTML body – tend to be easier to filter.

1	Text and HTML	34.4%
2	Text	31.9%
3	HTML	30.8%
4	Image	1.6%
5	Other	1.3%

1	Text	52.3%
2	Text and HTML	26.9%
3	HTML	14.4%
4	Other	5.1%
5	Image	1.3%

Left: Content of messages seen in the spam feeds.
Right: Content of spam messages missed by at least two full solutions.

Whether this really says something about spam filtering or whether this is a side effect of current filtering techniques (it could well be that the bots sending out HTML spam are easier to detect for other reasons) remains to be seen. We will certainly keep an eye on future results to see if this observed behaviour changes over time.

RESULTS

AnubisNetworks Mail Protection Service

SC rate: 99.38%

FP rate: 0.00%

Final score: 99.38

Project Honey Pot SC rate: 98.63%

Abusix SC rate: 99.81%

AnubisNetworks achieved the highest final score in the previous test – and although the Portuguese product did not manage to repeat the achievement this month, it did score an excellent spam catch rate and, like last time, no false positives. *AnubisNetworks* thus easily earns its fourth VBSpam award.

vb VERIFIED SPAM virusbtn.com

BitDefender Security for Mail Servers 3.0.2

SC rate: 99.79%

FP rate: 0.04%

Final score: 99.58

Project Honey Pot SC rate: 99.52%

Abusix SC rate: 99.95%

BitDefender's final score was slightly improved this month, thanks to the number of false positives having been reduced to just one. *BitDefender* thus continues its unbroken run of VBSpam awards, having earned one in every test to date.

vb VERIFIED SPAM virusbtn.com

Fortinet FortiMail

SC rate: 99.80%

FP rate: 0.13%

Final score: 99.17

Project Honey Pot SC rate: 99.77%

Abusix SC rate: 99.82%

vb VERIFIED SPAM virusbtn.com

If there were an award for the greatest improvement then *FortiMail* would

	True negative	False positive	FP rate	False negative	True positive	SC rate	Final score
AnubisNetworks	2357	0	0.00%	556	88471	99.38%	99.38
BitDefender	2356	1	0.04%	186	88841	99.79%	99.58
FortiMail	2354	3	0.13%	175	88852	99.80%	99.17
GFI VIPRE	2347	10	0.42%	1379	87648	98.45%	96.33
Kaspersky	2356	1	0.04%	373	88654	99.58%	99.37
Libra Esva	2357	0	0.00%	24	89003	99.97%	99.97
McAfee Email Gateway	2356	1	0.04%	19	89008	99.98%	99.77
McAfee EWS	2357	0	0.00%	543	88484	99.39%	99.39
MessageStream	2349	7	0.30%	82	88945	99.91%	98.42
OnlyMyEmail	2357	0	0.00%	16	89011	99.98%	99.98
Pro-Mail	2339	11	0.47%	675	88352	99.24%	96.90
Sophos	2356	1	0.04%	97	88930	99.89%	99.68
SPAMfighter	2351	6	0.25%	613	88414	99.31%	98.04
SpamTitan	2357	0	0.00%	25	89002	99.97%	99.97
Symantec Brightmail	2356	1	0.04%	53	88974	99.94%	99.73
The Email Laundry	2357	0	0.00%	116	88911	99.87%	99.87
Vade Retro	2355	2	0.08%	230	88797	99.74%	99.32
Vamsoft ORF	2357	0	0.00%	545	88482	99.39%	99.39
Webroot	2354	3	0.13%	85	88942	99.90%	99.27
Spamhaus*	2357	0	0.00%	1178	87849	98.68%	98.68

* As the only partial solution tested, the results for Spamhaus are listed separately from those of the full solutions.

certainly win it: the hardware appliance saw improvements to both its false positive rate and, most impressively, its spam catch rate, earning the product its tenth consecutive VBSpam award.

GFI VIPRE

SC rate: 98.45%

FP rate: 0.42%

Final score: 96.33

Project Honey Pot SC rate: 98.23%

Abusix SC rate: 98.58%

In this test, *GFI*'s *VIPRE* missed just over 1.5 per cent of all spam emails. That in itself was not a problem (though it does, of course, leave some room for improvement), but ten legitimate emails were blocked by the product too. With such a high false positive rate users may be less likely to forgive the product for allowing the odd spam message through to their inboxes. With the lowest final score of all products in the test, *VIPRE* fails to win a VBSpam award this month.

Kaspersky Anti-Spam 3.0

SC rate: 99.58%

FP rate: 0.04%

Final score: 99.37

Project Honey Pot SC rate: 99.04%

Abusix SC rate: 99.90%

VERIFIED SPAM virusbtn.com

As in the previous test, *Kaspersky*'s anti-spam solution managed to keep the number of incorrectly classified legitimate emails down to just one, so the extra weighting on false positives introduced in the final score calculations this month did not cause much of an issue. In fact, with an improved spam catch rate, the product saw its final score increase and thus it easily wins its ninth VBSpam award.

Libra Esva 2.0

SC rate: 99.97%

SC rate pre-DATA: 98.96%

	Project Honey Pot		Abusix		pre-DATA[†]		STDev[‡]
	FN	SC Rate	FN	SC Rate	FN	SC Rate	
AnubisNetworks	448	98.63%	108	99.81%	N/A	N/A	0.75
BitDefender	157	99.52%	29	99.95%	N/A	N/A	0.51
FortiMail	75	99.77%	100	99.82%	N/A	N/A	0.31
GFI VIPRE	578	98.23%	801	98.58%	N/A	N/A	1.55
Kaspersky	314	99.04%	59	99.90%	N/A	N/A	0.76
Libra Esva	21	99.94%	3	99.99%	928	98.96%	0.14
McAfee Email Gateway	16	99.95%	3	99.99%	N/A	N/A	0.11
McAfee EWS	512	98.43%	31	99.95%	N/A	N/A	0.94
MessageStream	54	99.83%	28	99.95%	N/A	N/A	0.32
OnlyMyEmail	6	99.98%	13	99.98%	N/A	N/A	0.10
Pro-Mail	454	98.61%	221	99.61%	N/A	N/A	0.82
Sophos	83	99.75%	14	99.98%	N/A	N/A	0.44
SPAMfighter	247	99.24%	366	99.35%	N/A	N/A	0.78
SpamTitan	8	99.98%	17	99.97%	N/A	N/A	0.09
Symantec Brightmail	40	99.88%	13	99.98%	N/A	N/A	0.17
The Email Laundry	110	99.66%	6	99.99%	381	99.57%	0.34
Vade Retro	179	99.45%	51	99.91%	N/A	N/A	0.75
Vamsoft ORF	358	98.90%	187	99.67%	N/A	N/A	0.78
Webroot	25	99.92%	60	99.89%	20165	77.35%	0.28
Spamhaus[*]	718	97.80%	460	99.18%	1189	98.66%	1.13

[*] As the only partial solution tested, the results for Spamhaus are listed separately from those of the full solutions.

[†] pre-DATA filtering was optional and was applied on the full spam corpus. There were no false positives in the pre-DATA filtering.

[‡] The standard deviation of a product is calculated using the set of its hourly spam catch rates.

(Libra Esva 2.0 contd.)

FP rate: 0.00%

Final score: 99.97

Project Honey Pot SC rate: 99.94%

Abusix SC rate: 99.99%

As I have said in previous reviews, in the business of spam filters, the devil is in the details. However, sometimes tiny details fall within the statistical error margin: *Libra Esva* missed just four more spam emails than the best performing product in this test and, as the virtual solution did not block any legitimate email, it achieved the second highest final score. *Esva*'s Italian developers should consider their product one of the winners of this test.

McAfee Email Gateway (formerly IronMail)

SC rate: 99.98%

FP rate: 0.04%

Final score: 99.77

Project Honey Pot SC rate: 99.95%

Abusix SC rate: 99.99%

McAfee's *Email Gateway* appliance saw its spam catch rate improve to just 19 missed spam emails, giving the product the joint best spam catch rate. But perhaps more impressive is the fact that its false positive rate was reduced greatly – to just one mislabelled legitimate email. With the fifth highest final score in this test, *McAfee*'s *Email Gateway* wins yet another VBSpam award.

McAfee Email and Web Security Appliance

SC rate: 99.39%

FP rate: 0.00%

Final score: 99.39

Project Honey Pot SC rate: 98.43%

Abusix SC rate: 99.95%

The second *McAfee* appliance also saw improvements to both its spam catch rate and its false positive rate; here it is the total lack of false positives that is most impressive. A very decent spam catch rate combined with that lack of false positives means that the product easily achieves its ninth consecutive VBSpam award.

vb VERIFIED SPAM virusbtn.com

MessageStream

SC rate: 99.91%

FP rate: 0.30%

Final score: 98.42

Project Honey Pot SC rate: 99.83%

Abusix SC rate: 99.95%

Fewer than one in 1,000 spam messages sent through *MessageStream*'s hosted solution made it to our MTA, which just shows the impact a spam filter can have. The product scored more false positives than the average solution in this test, which should be some concern for the developers and something for them to work on, but *MessageStream* still easily won its tenth VBSpam award.

vb VERIFIED SPAM virusbtn.com

OnlyMyEmail's Corporate MX-Defender

SC rate: 99.98%

FP rate: 0.00%

Final score: 99.98

Project Honey Pot SC rate: 99.98%

Abusix SC rate: 99.98%

OnlyMyEmail's *MX-Defender* made an impressive debut last month with the highest spam catch rate in the test. Not only did it manage to repeat that achievement this time, but it also achieved a score of zero false positives. This gives the product the highest final score this month, along with a very well deserved VBSpam award.

vb VERIFIED SPAM virusbtn.com

Pro-Mail (Prolocation)

SC rate: 99.24%

FP rate: 0.47%

Final score: 96.90

Project Honey Pot SC rate: 98.61%

Abusix SC rate: 99.61%

False positives caused problems for *Pro-Mail*'s hosted solution last month and continued to do so in this test – the product missed more legitimate email than any other solution in the test. While the product's spam catch rate was good, the high false positive rate was enough to keep its final score just below the threshold of 97, thus denying it a VBSpam award.

Sophos Email Appliance

SC rate: 99.89%

FP rate: 0.04%

Final score: 99.68

Project Honey Pot SC rate: 99.75%

Abusix SC rate: 99.98%

Sophos's hardware appliance combined another very decent spam catch rate with just one false positive (compared to four in the previous test) to give the product an excellent final score and earn it its sixth VBSpam award in as many tests.

vb VERIFIED SPAM virusbtn.com

SPAMfighter Mail Gateway

SC rate: 99.31%

FP rate: 0.25%

Final score: 98.04

Project Honey Pot SC rate: 99.24%

Abusix SC rate: 99.35%

SPAMfighter Mail Gateway saw its spam catch rate improve for the second time in a row, and while its false positive score increased – something the developers hope will be dealt with better by a new scanning engine – it easily achieved its eighth VBSpam award.

vb VERIFIED SPAM virusbtn.com

SpamTitan

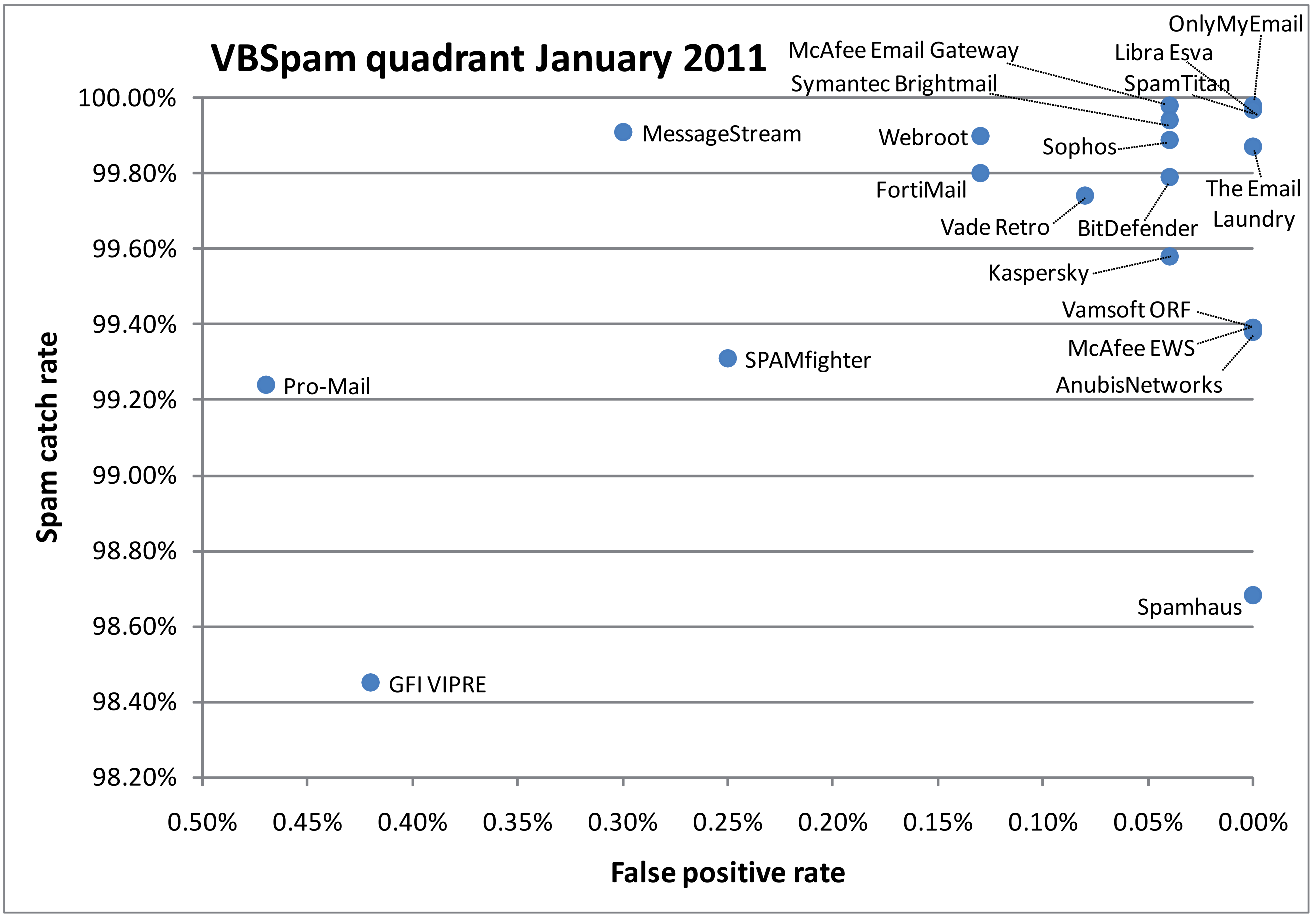
SC rate: 99.97%

FP rate: 0.00%

Final score: 99.97

Project Honey Pot SC rate: 99.98%

Abusix SC rate: 99.97%



SpamTitan not only equalled the stunning spam catch rate it displayed in the previous test, but the virtual appliance also correctly identified all legitimate emails. With close to the highest final score, *SpamTitan* is among the winners of this test.

Symantec Brightmail Gateway 9.0

SC rate: 99.94%

FP rate: 0.04%

Final score: 99.73

Project Honey Pot SC rate: 99.88%

Abusix SC rate: 99.98%

This month's test proved that the relatively high false positive rate displayed by *Symantec*'s virtual appliance in the last test was a one-off incident. With just one false positive this time, and a decent spam catch rate, the product is among the better performers in the test.

The Email Laundry

SC rate: 99.87%

SC rate pre-DATA: 99.57%

FP rate: 0.00%

Final score: 99.87

Project Honey Pot SC rate: 99.66%

Abusix SC rate: 99.99%

In its pre-DATA filtering, *The Email Laundry* blocks more spam than several solutions do overall. During the content scanning phase, more than two-thirds of the remaining spam was blocked. This, combined with the fact that there wasn't a single false positive, gave the hosted solution the fourth highest final score.

Vade Retro Center

SC rate: 99.74%

FP rate: 0.08%

Products ranked by final score	Final score
OnlyMyEmail	99.98
Libra Esva	99.97
SpamTitan	99.97
The Email Laundry	99.87
McAfee Email Gateway	99.77
Symantec Brightmail	99.73
Sophos	99.68
BitDefender	99.58
McAfee EWS	99.39
Vamsoft ORF	99.39
AnubisNetworks	99.38
Kaspersky	99.37
Vade Retro	99.32
Webroot	99.27
FortiMail	99.17
Spamhaus	98.68
MessageStream	98.42
SPAMfighter	98.04
Pro-Mail	96.90
GFI VIPRE	96.33

(Vade Retro Center contd.)

Final score: 99.32

Project Honey Pot SC rate: 99.45%

Abusix SC rate: 99.91%

Both the spam catch rate and false positive rate for *Vade Retro* were slightly better in the last test, but the French hosted solution had some leeway: both are still good, and with a more than decent final score, the product wins its fifth consecutive VBSpam award.

Vamsoft ORF

SC rate: 99.39%

FP rate: 0.00%

Final score: 99.39

Project Honey Pot SC rate: 98.90%

Abusix SC rate: 99.67%

ORF's false positive rates have always been among the lowest of the products we've tested so we were not surprised to see it among those that scored zero false positives in this month's test. With another very good spam catch rate, *ORF* earns its fifth VBSpam award in as many tests.

Webroot

SC rate: 99.90%

SC rate pre-DATA: 77.35%

FP rate: 0.13%

Final score: 99.27

Project Honey Pot SC rate: 99.92%

Abusix SC rate: 99.89%

Webroot's spam catch rate has been close to 100% for many tests in a row and this one was no exception. I was pleased to see a good reduction in the number of false positives as well, giving the hosted solution an improved final score even using the more challenging formula, and earning the product its tenth consecutive VBSpam award.

Spamhaus Zen+DBL

SC rate: 98.68%

FP rate: 0.00%

Final score: 98.68

Project Honey Pot SC rate: 97.80%

Abusix SC rate: 99.18%

This test showed that the false positives produced by *Spamhaus* last month really were just a hiccup: in this test (as in all other tests but the previous one) no legitimate email was blocked using the combination of IP (*Zen*) and domain (*DBL*) based blacklists. Meanwhile, a lot of spam was blocked and with far from the lowest final score, *Spamhaus* demonstrates that, even as only a partial solution, it is well up to the task.

CONCLUSION

Developers of the two products that failed to win a VBSpam award this time around will be hard at work in the interim between this test and the next to improve their products' performance and ensure they make it to the winners' podium next time. Meanwhile, the developers of the other products will have to demonstrate that they are capable of keeping up with the way spam changes.

And so will we at *Virus Bulletin*. Just as the developers of anti-spam solutions can never rest on their laurels, we will keep looking at ways to improve our tests and make sure their results are as accurate a reflection of real customer experience as possible. Watch this space for an announcement of the exciting new additions we are planning.

END NOTES & NEWS

Black Hat DC takes place 16–19 January 2011 in Arlington, VA, USA. For details see http://www.blackhat.com/.

The 10th Ibero-American Seminar on Information Technology Security will be held 7–11 February 2011 in Havana, Cuba. For details see http://www.informaticahabana.cu/en/home.

RSA Conference 2011 will be held 14–18 February 2011 in San Francisco, CA, USA. For more information see http://www.rsaconference.com/2011/usa/.

The 12th annual CanSecWest conference will be held 9–11 March 2011 in Vancouver, Canada. See http://cansecwest.com/.

The 8th Annual Enterprise Security Conference will be held 14–15 March 2011 in Kuala Lumpur, Malaysia. The theme for the 2011 conference is 'Improving digital security to protect your assets from malicious cybercrime'. For details see http://www.acnergy.com/.

Black Hat Europe takes place 15–18 March 2011 in Barcelona, Spain. For more information see http://www.blackhat.com/.

Infosecurity Europe will take place 19–21 April 2011 in London, UK. For more details see http://www.infosec.co.uk/.

SOURCE Boston 2011 will be held 20–22 April 2011 in Boston, MA, USA. For more details see http://www.sourceconference.com/.

The New York Computer Forensics Show will be held 26–27 April 2011 in New York, NY, USA. For more information see http://www.computerforensicshow.com/.

The 5th International CARO Workshop will be held 5–6 May 2011 in Prague, Czech Republic. The main theme of the conference will be 'Hardening the net'. A call for papers has been issued, with deadlines for submissions of 31 January. Abstracts and information requests should be sent to workshop@caro2011.org. Other details will be available soon on the conference website at http://www.caro2011.org.

The 20th Annual EICAR Conference will be held 9–10 May 2011 in Krems, Austria. This year's conference is named 'New trends in Malware and Anti-malware techniques: myths, reality and context'. A pre-conference programme will run 7–8 May. For full details see http://www.eicar.org/conference/.

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/.

The 2011 National Information Security Conference will be held 8–10 June 2011 in St Andrews, Scotland. Registration for the event is by qualification only – applications can be made at http://www.nisc.org.uk/.

The 23rd Annual FIRST Conference takes place 12–17 June 2011 in Vienna, Austria. The conference promotes worldwide coordination and cooperation among Computer Security Incident Response Teams. For more details see see http://conference.first.org/.

SOURCE Seattle 2011 will be held 16–17 June 2011 in Seattle, WA, USA. For more details see http://www.sourceconference.com/.

Black Hat USA takes place 30 July to 4 August 2011 in Las Vegas, NV, USA. DEFCON 19 follows the Black Hat event, taking place 4–7 August, also in Las Vegas. For more information see http://www.blackhat.com/ and http://www.defcon.org/.

The 20th USENIX Security Symposium will be held 10–12 August 2011 in San Francisco, CA, USA. See http://usenix.org/.

VB2011 will take place 5–7 October 2011 in Barcelona, Spain. *VB* is currently seeking submissions from those wishing to present at the conference. Full details of the call for papers are available at http://www.virusbtn.com/conference/vb2011. For details of sponsorship opportunities and any other queries relating to VB2011, please contact conference@virusbtn.com.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

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

Corporate rates include a licence for intranet publication.

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

Editorial enquiries, subscription enquiries, orders and payments:
Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England
Tel: +44 (0)1235 555139 Fax: +44 (0)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.