MAY 2003

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

2 **COMMENT**
The bounds of reason

3 **NEWS**
Altered images
VB2003

3 **VIRUS PREVALENCE TABLE**

4 **LETTERS**

VIRUS ANALYSES
5 Delinquent Deloder: W32/Deloder.A
7 Casualties of war

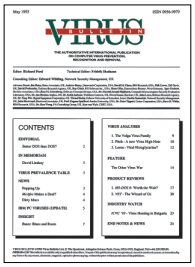
10 **TECHNICAL FEATURE**
Mission impossible: WebDAV update

FEATURES
13 Don't take Code Red lightly
15 Out of Africa …

18 **COMPARATIVE REVIEW**
RedHat Linux

24 **END NOTES AND NEWS**

IN THIS ISSUE



RINGING THE CHANGES

Spotted the difference? All spruced up and ready to face the 21st century, this issue marks *VB*'s first change in appearance for a decade.

page 3

CAUGHT IN THE ACT

The author of W32/Ganda was found and arrested by Swedish police within days of the worm's release. Gábor Szappanos and Gábor Molnár look at this poorly-coded email worm and declare that the author deserves at least half the prison sentence he is likely to receive purely for his sad attempts at coding.

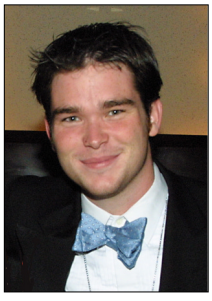
page 7

A LOOK IN THE REAR VIEW MIRROR

There is nothing innovative about recent strains of Klez, Yaha, Sircam and Code Red. Yet these have demonstrated unprecedented staying power on the Internet. Larz Sherer presents a strategy to assist network and security administrators in addressing 'new', yet old, threats.

page 13

ISSN 0956-9979



'The pot and the kettle have never had it so good.'

Peter Sergeant
Virus Bulletin

THE BOUNDS OF REASON

A recent thread on the *SecurityFocus focus-virus* mailing list began: 'Is there anything about Dr Hruska or his background that gives him the skills necessary to perform profiling [of virus writers]?' The question had been raised in response to an interview with Dr Hruska published on the *Reuters* website.

The question seems a little unnecessary. Not only is the man in question the co-founder of a successful anti-virus company, but he has written several publications on computer security and spoken many times at computer security conferences – he is a *bona fide* anti-virus expert by anyone's standards.

'The inherent problem with this profile (and with so many other popularized profiles) is that it's so sweeping and general as to be easily proven correct', pointed out one contributor to the *focus-virus* thread. The interview read '[virus writers] have a chronic lack of girlfriends [and] are usually socially inadequate.' But, of four virus writers profiled by researcher Sarah Gordon, three had girlfriends. Furthermore, the same *focus-virus* contributor notes, 'the computer industry is highly demanding … time to expand one's knowledge isn't always available during the hours of 9-5 … this places a strain on one's social life; exactly what happens in any other demanding career.'

But there is a more serious side to the repeated use of these broad stereotypes: 'The problem of dealing with the danger posed by the distribution of malicious software is not simplified by failing to recognise that the people who write viruses do not form a homogeneous group,' says Sarah Gordon, who has authored research papers about ethical development in virus writers.

To be fair, the anti-virus company in question is one of the more successful when it comes to the avoidance of hype, reiterating time and again that it's wrong to intimidate users. Despite regular attacks on somewhat over-stereotyped virus writers, most of this company's press releases stick to the facts. And the tendency to vilify virus writers is perhaps natural for those whose job is, effectively, to sabotage their work.

Other AV companies seem to experience substantially more difficulty in staying within the bounds of reason, however. One vendor's capitalization on 'VBS/Antrax' was described as 'cynical and tasteless marketing' (see *VB*, November 2001, p.3), only to be outdone by a rival company which actually *forged* screenshots of the virus. Another vendor's hyping of the non-eventful JPEG virus was similarly shot down (see *VB*, July 2002, p.3), when readers were reminded of a letter, sent only two years previously, in which a spokesperson for the company had decried the spurious press releases issued by one of its competitors. The pot and the kettle have never had it so good.

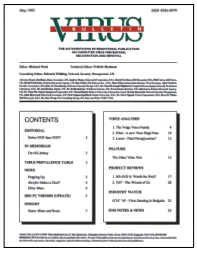
Whereas Ms Gordon found that ethics tended to develop in virus writers, it would appear that, in anti-virus PR divisions, they cycle: companies are all too happy to protest over the marketing ploys of others, while merrily pumping out their own junk. As Helen Martin, editor of *VB* said: 'the temptation to churn out press releases at every conceivable opportunity is irresistible. Whether the result is a pile of groaningly tenuous PR 'stories' or less than helpful scare-mongering, seems to be the luck of the editor's draw.'

Some smaller companies don't command huge marketing budgets, and need to score all the free publicity they can, so what can anti-virus companies do to keep out the hype? I propose a set of guidelines:

- Keep in mind that you are trying to get users to take viruses seriously – which they will find harder to do if they have to wade through hype first.
- Use press releases to explain what the virus does, and how users can minimize damage, rather than just harping on about how great your company is.
- If you must engage in sensationalist marketing tactics, please keep your hypocrisy to a minimum when a competitor does the same.

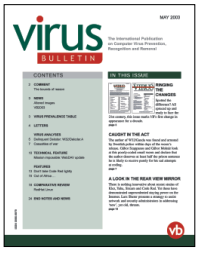
NEWS

ALTERED IMAGES



Readers with a keen eye for detail will have noticed a change on tearing open their *VB* envelopes this month … For the first time in almost a decade, *Virus Bulletin* has altered its appearance. Like the last time the magazine went for a new look, the intention is not to change the magagine beyond recognition, but to breathe a little fresh air into the publication and give it an image that we hope will endure long into the 21st century.

This is the most radical of the changes *VB* has seen over its near 14-year history and, like any changes to a long-established routine, may take a little getting used to. Beneath its new wardrobe, however, *VB* remains unchanged in its aims and objectives. In 1993 (see *VB* September 1993, p.2), then-editor Richard Ford wrote, 'It is all too easy to panic at the apocalyptic stories which can be plastered over the tabloid headlines: Jersualem, Datacrime, and Michelangelo have all been heralded as the "end of computing as we know it" … Throughout all this panic and hype, *VB* has provided a rock-solid platform upon which to build the fundamentals of a good computer security policy.'



Almost ten years on we have seen wide-reaching changes both within the virus and anti-virus arenas and in the world at large, yet substitute the names of the viruses with, say, Code Red, Nimda and Slammer (or take your pick from any of the hundreds that have been hyped by the media) and Richard's statement still holds true. As always, readers' comments are welcomed (comments@virusbtn.com).

VB2003

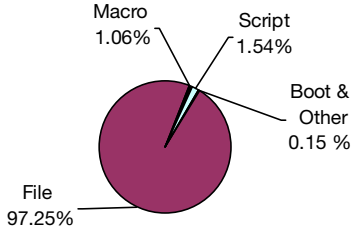
A temporary warning issued last month by the World Health Organization (WHO) advising against travel to Toronto prompted a number of enquiries as to the status of the Virus Bulletin conference, VB2003. On 29 April 2003 the WHO announced that it was lifting its warning against travel to Toronto. The organisers of VB2003 will continue to monitor all official advice closely, but currently have every reason to hope that the 25–26 September conference will go ahead in Toronto as planned. For details of the conference programme, exhibition and for online registration see http://www.virusbtn.com/.

Prevalence Table – March 2003

Virus	Type	Incidents	Reports
Win32/Klez	File	3179	42.07%
Win32/Opaserv	File	2011	26.61%
Win32/Dupator	File	379	5.02%
Win32/Bugbear	File	257	3.40%
Win32/Gibe	File	243	3.22%
Win32/Funlove	File	208	2.75%
Win32/Yaha	File	201	2.66%
Win32/Magistr	File	166	2.20%
Win32/Sobig	File	150	1.98%
Win32/Lirva	File	136	1.80%
Win32/Nimda	File	99	1.31%
Redlof	Script	90	1.19%
Win32/SirCam	File	51	0.67%
Win32/Hybris	File	41	0.54%
Laroux	Macro	36	0.48%
Win32/BadTrans	File	36	0.48%
Win95/ClH	File	32	0.42%
Win32/Lovgate	File	26	0.34%
Win95/Lorez	File	15	0.20%
Win32/Elkern	File	12	0.16%
Kak	Script	10	0.13%
Marker	Macro	10	0.13%
Win32/Ganda	File	10	0.13%
Others[1]		159	2.10%
Total		7557	100%

[1]The Prevalence Table includes a total of 159 reports across 75 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

A TRIBUTE TO AN ICONOCLAST

Sadly, the anti-virus community lost one of its most individual and unique members in March 2003.

Simon Widlake was a fiercely independent human being and virus researcher. He had an almost equal dislike of the AV and vX worlds – a position which led to many disputes with those who encountered him. However, he was unquestionably talented, with a huge knowledge of the subject of viruses, and an almost autistic obsession with detail.

His greatest contribution to the anti-virus field, while only four words, was one of which he was justifiably proud. The statement 'Viruses infect, worms infest' has since been referred to by many as a useful method of distinguishing these often troublesome groups of malware. The statement even found its way into *Viruses Revealed* (David Harley *et al*). Simon was very proud of his contribution to that book.

He was also proud to have discovered a problem in a specific scanner's detection of boot sector viruses – a particular obsession of Simon's.

David Harley remembers him thus:

> 'Simon and I shared many a lunch together discussing viruses, worms, octopii, hoax management, malware management, email management, my books, and virus writers, and hardly ever agreed on anything. However, I shall miss his undoubted technical insight, his ability to follow a subject down to its finest detail, and even his idiosyncratic style of email.
>
> Cheers, Simon. I'll raise my glass to you next time I eat lobster.'

This perhaps sums up what was best (and worst) about Simon. He was rigidly uncompromising when he thought he was right, and his arguments were forceful, cogent and often biting, but he was by no means one-sided.

On the many occasions when we disagreed, he always made the effort to ensure I understood that, while we might disagree about things, he still counted me as a friend. I valued that honesty.

Simon will probably be best remembered by the inhabitants of alt.comp.virus, where latterly (posting under the nym S.CHnappers) he seemed to delight in posting obscure and self-referential messages in a style which can only be described as bizarre.

His odd style gained him as many friends as it did enemies – he first came to my notice as I struggled to make sense of a particularly convoluted (but quite witty) posting. Simon was not someone who played well in groups, he was a true individual, but for all that, he was a good friend to those who could get past such barriers, and I, for one, am glad to have known him.

I'll let David Perry sum up:

> '"So then, what makes a Trojan horse?" I asked him. Simon took a sip of his beer, bit his lip and replied, "If it ain't what it says on the can, then it's a Trojan."
>
> I met Simon only three times, but he was one of the best people to know, to have known.
>
> To the little Joe Gould of the Internet underground, I raise my glass (although my glass contains only water – you know what was in his). Down the road, Simon, we'll catch up later.'

Andrew Lee
Independent AV researcher

VIRUS ANALYSIS 1

DELINQUENT DELODER: W32/DELODER.A

Ronald C. Bautista
TrendLabs, Trend Micro Inc., Philippines

In the early hours of 9 March 2003, a significant number of virus infection reports were received from China. A new Internet worm was found to have infected several establishments in the country: W32/Deloder.

Deloder is an Internet worm that does not use email for propagation. Instead, it propagates across the Internet by connecting through TCP port 445. It targets systems that are running *Windows 2000* and *XP*.

The worm carries a backdoor component that allows remote administration of compromised systems. What makes this worm different from others is that it uses two legitimate programs, one for its propagation and one for its backdoor routine.

ANATOMY OF THE WORM

Deloder is written in Microsoft Visual C++ and compressed with Aspack. The worm carries a backdoor Trojan and two legitimate network utilities: PsExec from SysInternals and VNC, developed at AT&T Laboratories Cambridge (see http://www.realvnc.com/).

PsExec is a remote process launcher which is used by the worm for uploading and executing both itself and its backdoor component on remote machines. VNC, which stands for Virtual Networking Computing, is a remote administration tool which the worm uses to access the infected machine remotely.

THE USUAL SUSPECT

Like a lot of other malware, this worm creates an autorun entry in the registry to enable it to be executed automatically during Windows startup:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
messnger = <worm's path>\Dvldr32.exe
```

When executed, the first thing the worm does is to check the version of the *Windows* operating system. If the system is running *Windows 2000*, *XP* or the *2003 Server* family, the worm proceeds. If the computer is running another operating system, the worm terminates.

In order to prevent multiple instances of the worm in memory, the worm creates a unique mutex named 'testXserv'. The worm checks for this mutex before resuming its execution process. If the mutex exists already, the worm terminates.

GUESSING GAME

In order to propagate via the Internet, the worm connects to random IP addresses using TCP port 445. This port, also known as the Microsoft-DS, is the default file-sharing port used by *Windows 2000* and *XP*.

The worm targets *Windows 2000* and *XP* systems with passwords that are 'weak', or easy to guess. The worm attempts to log on to these machines as an administrator by trying any of the passwords listed in its small dictionary of 'guessable' passwords:

<no password>	123asd	foobar	pw
0	123qwe	god	pw123
000000	2002	godblessyou	pwd
00000000	2003	home	qwer
007	2600	ihavenopass	root
1	54321	Internet	secret
110	654321	Login	server
111	88888888	login	sex
111111	a	love	super
11111111	aaa	mypass	sybase
12	abc	mypass123	temp
121212	abc123	mypc	temp123
123	abcd	mypc123	test
123123	Admin	oracle	test123
1234	admin	owner	win
12345	admin123	pass	xp
123456	administrator	passwd	xxx
1234567	alpha	Password	yxcv
12345678	asdf	password	zxcv
123456789	computer	pat	
1234qwer	database	patrick	
123abc	enable	pc	

When the log-on attempt is successful, the worm uses the PsExec utility to copy itself as Dvldr32.exe and the backdoor installer as inst.exe on the Windows system directory.

Using the PsExec utility again, the worm executes those dropped files remotely. In addition, the worm may drop the backdoor installer, inst.exe, in the following startup folders:

\<ip address>\C$\WINNT\All Users\Start Menu\Programs\Startup

\<ip address>\C\WINDOWS\Start Menu\Programs\Startup

\<ip address>\C$\Documents and Settings\All Users\Start Menu\Programs\Startup

Based on its random generator, the worm may disable the following hidden default shares: ADMIN$, D$, IPC$, E$, C$ and F$.

OPENING THE BACK DOOR

In addition to its propagation activities, the worm compromises the infected system by planting a backdoor Trojan. The backdoor arrives as an installer named 'inst.exe'. It is dropped and executed remotely by the worm during propagation.

When inst.exe executes, it extracts four files to the Windows Fonts folder: rundll32.exe, explorer.exe, omnithread_rt.dll and VNCHooks.dll. A further file, cygwin1.dll, is dropped in the Windows System folder.

With the exception of rundll32.exe, which is the main backdoor program, all the files that are dropped by the worm are non-malicious, but they are used by the backdoor program for its malicious routines.

The file explorer.exe, whose original filename is winvnc.exe, is the legitimate remote administration tool VNC. This tool allows the user to view the Desktop and manipulate the machine remotely.

The dll files omnithread_rt.dll and VNCHooks.dll are both dynamic libraries used by the VNC application. The other dll file, cygwin1.dll, known as the Cygwin Posix Emulation DLL, is a Unix emulator which provides Unix API functions for the backdoor.

To ensure that the backdoor is executed on every Windows startup, the following entries are added in the autorun registry key:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\
TaskMan = %Windows%\Fonts\rundll32.exe
Explorer = %Windows%\Fonts\explorer.exe
```

INFORM THE MASTERMIND

The worm's backdoor component, rundll32.exe, is responsible for notifying the malware creator that the infected machine is online and is ready for access.

The backdoor connects to any of the following IRC servers via port 6667:

cocket.nailed.org	cocket.minidns.net
cocket.mooo.com	cocket.dyn.nicolas.cx
cocket.bounceme.net	cocket.dynup.net
cocket.phathookups.com	cocket.pokemonfan.org
cocket.gotdns.com	cocket.staticcling.org
cocket.ma.cx	cocket.getmyip.com
cocket.orgdns.org	

Finally, it sends a notification to the following IRC nicknames:

garc	titi	boyzz
rock	kiwi	nikis
step	poer	south
wolf	fuck	penis
radi	turu	rahim
mike	coked	monic
moon	micha	uglyc
rosi	girli	serve
schen	trick	

LOWDOWN DELODER

While most other malware makes use of maliciously crafted files to complete its routine and add to its functionalities, Deloder goes for the easier route – using valid utilities for its malicious activities.

Making use of legitimate files makes it much easier for the author to create a program for malicious ends. The author did not need to bother himself with writing sophisticated code in order to achieve his malicious intentions.

This technique of exploiting third-party utilities to accomplish certain tasks is becoming a trend in the virus world. We have seen a lot of backdoor Trojans recently that made use of legitimate programs such as mIRC client program, HideWindow utility, ftp server applications, and so on.

This time, even a worm has made use of a legitimate program. Should this trend continue, future malware may become more powerful than ever. Detection for worms and backdoors will become more difficult, since we will have to identify first which of the components are the legitimate programs (should not be detected) and which ones are malicious (should be detected).

W32/Deloder.A

Type:	Internet worm.
Removal:	Delete detected files. Registry run entries created by the worm should be deleted.
Aliases:	WORM_DELODER.A, W32.HLLW.Deloder, W32/Deloder.worm, Worm.Win32.Deloder.

VIRUS ANALYSIS 2

CASUALTIES OF WAR: W32/GANDA

Gábor Molnár and Gábor Szappanos
VirusBuster, Hungary

There could have been no doubt that the theme of the Iraqi war would be picked up by virus writers and, sure enough, it was no time at all before W32/Ganda appeared.

This virus is a below-average email worm, which is also a parasitic Win32 PE EXE infector, and is written poorly in assembly language. It provides an excellent example of redundant programming: identical or similar procedures are used several times in the code, for no obvious reason. Still, it could make it to the WildList and, thanks to an unusual infection method, its disinfection proves rather problematic. In fact, complete disinfection is impossible, but infected executables can be recovered to functionally equivalent forms of the original programs.

THE I-WORM

The worm arrives in an email message as a 45056-byte attachment. Some of the messages in which the worm is sent make use of a known *Internet Explorer* vulnerability, described in the security bulletin MS01-20.

Most of the string constants in the virus are encrypted using an extremely primitive bit-wise negation. Ganda was written in Sweden, judging by the encrypted (and unused) text:

```
[WORM.SWEDENSUX] Coded by Uncle Roger in
Härnösand, Sweden, 03.03. I am being
discriminated by the swedish schoolsystem.
This is a response to eight long years of
discrimination.

  I support animal-liberators worldwide
```

While this message is not used, it provides clues as to the age, occupation and location of the virus author.

When the virus is executed, it collects the addresses of the Windows API functions that it will need in the code. The API names are also 'encrypted' – their first four character bytes are incremented by one. Then the worm checks whether there is already an instance running, by using the SWEDENSUX mutex. If another instance of the worm is running, the worm exits.

Next, the worm copies itself into the Windows folder as SCANDISK.EXE. An additional copy is created in the same folder, with an eight-letter random name and .EXE extension. SCANDISK.EXE is registered for startup under the registry key HKLM\Software\Microsoft\Windows\CurrentVersion\Run\Scandisk.

A third copy of the worm, named TMPWORM.EXE, is created in the Windows folder when the worm builds the outgoing mail message and encodes itself using the BASE64 algorithm.

Then the virus scans the desktop folder and the Start Menu folder recursively for .LNK, .EXE and .SCR files to infect. These folders contain links to commonly used executables, and Windows resolves API access to these links as direct access to the executables to which they point.

Like many contemporary worms, Ganda includes an anti-AV procedure, though it is somewhat more sophisticated than most. The worm enumerates the running processes, then kills those whose (lower-case) name includes one of the following strings:

virus	pc-cillin
firewall	trend micro
f-secure	kaspersky
symantec	sophos
mcafee	norton

Moreover, the virus enumerates the registry keys under HKLM\Software\Microsoft\Windows\CurrentVersion\Run and HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices.

If any of the keys contain one of the strings listed above, the virus will modify the file to which the key points by inserting a 0xC3 (RET) instruction at the entry point, causing the virus scanner to abort immediately upon execution.

On *Windows 9x*-type operating systems (i.e. where the PlatformID is not 2) Ganda also enumerates the keys under HKLM\System\CurrentControlSet\Services\VxD and checks whether any of the AV programs listed above is loaded as a VXD. If a matching entry is found, the virus deletes the registry key, but the programs are left intact.

Next the virus waits in an infinite loop for an Internet connection to become active. If the connection is active, Ganda searches all local drives plus the Internet browser cache path (the location stored under HKLM\Software\Microsoft\Windows\CurrentVersion\InternetSettings\Cache\Path) for email addresses. As the cache is normally on one of the fixed drives, the second search seems a little superfluous.

The virus scans all .HTM, .DBX and .EML files for email addresses. In each file the virus searches for two possible strings, *To:* and *lto:*, anything to the right of which would be an email address. The virus collects at most 1000 addresses during this search. The virus will abort the current file if it contains an address-like string that is longer than 29 bytes, because for each address a 32-byte area is used internally,

where the address is stored between '<' and '>' terminators, left-aligned, right-padded with zero bytes.

Although the address grep is very similar for the two substrings, the virus uses two separate procedures.

The virus uses a third method – the interface functions in WAB.DLL – for gathering addresses, as well. By calling the WABOpen API function, the virus can call the IAddrBook and related interface member functions – a rather unusual high-level language technique (seen also in W32.Fusic), used rarely in viruses.

SENDING MESSAGES

First the virus sends a message in Swedish to a list of Swedish email addresses (mostly journalists). This message has the sender skrattahaha@hotmail.com and the recipients are:

qruvabzabr@hotmail.com
red@fna.se
debatt@svt.se
susanne.sjostedt@tidningen.to
skolverket@skolverket.se
mary.martensson@aftonbladet.se
katarina.sternudd@aftonbladet.se
cecilia.gustavsson@aftonbladet.se
jessica.ritzen@aftonbladet.se
margareta.cronquist@tidningen.to
annika.sohlander@aftonbladet.se
kerstin.danielson@aftonbladet.se
insandare@tidningen.to
insandare@aftonbladet.se

Then the infected message is sent out to all the email addresses that have been collected.

Sometimes two copies of the worm are sent out, the first with subject and message bodies selected at random. If the language ID of the operating system is set to 1053 (Swedish), the body and the subject are selected from ten predefined Swedish text pairs, and in any other language setting they are selected from ten English body-subject pairs.

The English pairs are as follows:

Variant 1

```
Title: Screensaver advice.

Message body:
Do you think this screensaver could be considered
illegal? Would =
appreciate if you or any one of your friends could
check it out and =
answer as soon as humanly possible. Thanx !
```

Variant 2

```
Title: Spy pics.

Message body:
Here's the screensaver i told you about. It contains
pictures taken by =
one of the US spy satellites during one of it's
missions over iraq. If =
you want more of these pic's you know where you can
find me. Bye!
```

Variant 3

```
Title: GO USA !!!!

Message body:
This screensaver animates the star spangled banner.
Please support the =
US administration in their fight against terror.
Thanx a lot!
```

Variant 4

```
Title: G.W Bush animation.

Message body:
Here's the animation that the FBI wants to stop.
Seems like the feds are =
trying to put an end to peoples right to say what
they think of the US =
administration. Have fun!
```

Variant 5

```
Title: Is USA a UFO?

Message body:
Have a look at this screensaver, and then tell me
that George.W Bush is =
not an alien. ;-)
```

Variant 6

```
Title: Is USA always number one?

Message body:
Some misguided people actually believe that an
american life has a =
greater value than those of other nationalities.
Just have a look at =
this pathetic screensaver and then you'll know what
i'm talking about. =
All the best.
```

Variant 7

```
Title: LINUX.

Message body:
Are you a windows user who is curious about the
linux environment? This =
screensaver gives you a preview of the KDE and GNOME
desktops. What's =
more, LINUX is a free system, meaning anyone can
download it.
```

Variant 8

```
Title: Nazi propaganda?

Message body:
This screensaver has been banned in Germany. It
contains a number of =
```

```
animated symbols that can be related to the nazi
culture. What do you =
think, is it a legitimate ban or not? Please answer
asap. Thanx!
```

Variant 9

```
Title: Catlover.

Message body:
If you like cats you'll love this screensaver. It's
four animated =
kittens running around on the screen. Contact me for
more clipart. Have =
fun! ;-)
```

Variant 10

```
Title: Disgusting propaganda.

Message body:
Hello! My 12 year old doughter received this
screensaver on a CDROM that =
was sent to her through advertising. I find it
disturbing that children =
are now being targets of nazi organizations. I would
appreciate to hear =
from you on this matter, as soon as possible.
Thank you.
```

The sender and the recipient addresses are picked from the list of email addresses that have been collected. The worm creates two registry keys, HKLM\Software\SS\Sent and HKLM\Software\SS\Sent2, in order to register the addresses to which it has been mailed.

When picking an email address to use as either the recipient or the sender, the virus checks against the subkeys under this location. However, it may fail to save the addresses properly after it has sent the messages and, as a result, the virus may send itself to the same address repeatedly.

The message is sent with random subject, body and attachment name, the attachment name consists of two characters selected at random and a .SCR extension.

Occasionally a second message may be sent to the recipient. In this case the subject is empty, the body reads simply 'Myzli!', and the email contains a short HTML loader that runs the attachment using a known iframe vulnerability in *Internet Explorer*. The virus is attached to the message as XX.SCR.

Needless to say, the three messages are sent using three different procedures.

Messages are sent using the worm's own SMTP engine. It collects the SMTP server addresses defined in the CU\Software\Microsoft\InternetAccount Manager\Accounts section. At most, ten SMTP servers are collected.

In case no SMTP server is defined in any of these accounts, the virus adds 'm1.611.telia.com' to the end of this list. The virus will attempt to connect to the first server, then in case the communication fails during any phase, it closes the socket and switches to the next server.

FILE INFECTION

The worm searches for Win32 PE executable files including screen savers (*.LNK – whatever it points to, *.EXE and *.SCR). It creates a memory map to infect. The worm inserts zero bytes at the end of the last section then inserts the 567-byte worm loader component. In order to avoid multiple infections the worm writes 0x7219 to the checksum field of the infected file's header as an infection marker.

This worm loader uses the import addresses of the original program; DWORD pointers refer to these functions:

ExitProcess

GetProcAddress

GetModuleHandle

It rewrites the instruction 'call ExitProcess' with the instruction 'jump WormLoader'. The worm can determine and rewrite the following methods of calling this function:

```
FF 15 ?? ?? ?? ?? call ExitProcess; absolute
indirect call

E8 ?? ?? ?? ?? call j_ExitProcess; relative call
...
FF 25 ?? ?? ?? ?? jmp ExitProcess;

A1 ?? ?? ?? ??    mov  eax,ExitProcess;register
absolute address and call later via eax

1D 8B ?? ?? ?? ?? mov  ebx,ExitProcess;register
absolute address and call later via ebx

0D 8B ?? ?? ?? ?? mov  ecx,ExitProcess;register
absolute address and call later via ecx

15 8B ?? ?? ?? ?? mov  edx,ExitProcess;register
absolute address and call later via edx

35 8B ?? ?? ?? ?? mov  esi,ExitProcess;register
absolute address and call later via esi

3D 8B ?? ?? ?? ?? mov  edi,ExitProcess;register
absolute address and call later via edi

2D 8B ?? ?? ?? ?? mov  ebp,ExitProcess;register
absolute address and call later via ebp
```

In most cases applications use the first (absolute indirect call) method. This example shows the original along with the infected program code:

```
original                        infected
...                             ...
call ExitProcess (6 bytes)      jmp TheWorm (5 bytes)
...                             nop (1 byte)

                    TheWorm:

                                ...  ;Worm - component
                                call ExitProcess (6 bytes)
```

Before the application quits, the worm is activated. After the worm has been executed, it sets the function addresses from the pointers, and queries the address of the CreateProcess function. Then it changes to the Windows directory and executes the worm copy stored in the Windows folder with the CreateProcess function. The

filename of the worm is stored directly in this component during the infection.

This is not a 100% guaranteed method for worm activation, as the infected application may have several exit points, and Ganda will patch only the first one it finds.

DISINFECTION

Disinfection of an infected file is not simple. The perfect disinfection method would be to use the same method as the worm to find the modified ExitProcess call. But this is very difficult and slow, therefore most AV programs delete the infected files.

Anti-virus software users don't tend to like to lose any data or applications, so we resolved to remove the worm component from infected files. This method does not restore the file byte-for-byte; it only restores the application functionally: first the worm loader code is removed by cutting the file at the start of the worm loader. Then the instruction call ExitProcess is created at the entry of the worm loader. Finally the necessary checks are performed to ensure that the ExitProcess call is correct.

CONCLUSION

It is very fortunate that this virus couldn't cause massive infections. If it were even a mediocre (as far as success is concerned) worm, and had infected thousands of computers, then AV vendors and system administrators would have been in trouble.

Several AV programs don't disinfect the infected executables (which, strictly speaking, is the correct handling of the situation); the only solution is to restore/reinstall infected files. That is a nuisance even on a single computer. Fortunately, however, this virus was not a big hit.

EPILOGUE

It is a very rare occasion that the author of a virus is found before the analysis of the same virus could appear in *Virus Bulletin*. The author of this worm left many clues which could be used to trace him. Not only his handle (Uncle Roger), but also his age and city location are hidden within the virus code. No wonder the Swedish police found the author so quickly. Of course, when they found him he used the same tired excuse that has been used by virus writers time and again: he never thought that his creation would cause any trouble …

The sentence for this man's crime could be as many as four years in prison – half of which he deserves just for the poor coding.

TECHNICAL FEATURE

MISSION IMPOSSIBLE: WEBDAV UPDATE

Aleksander Czarnowski
AVET Information and Network Security

'Our intelligence reports the possibility of a very dangerous security flaw in the default installation of *Microsoft* Internet Information Services 5.0. The exploit is not publicly available, however we have unconfirmed reports of system penetration using an unknown security hole in the wild …' While this might have been the introduction to a computer game or movie, it is actually a poor joke about the recent *Microsoft Security Bulletin* MS03-007 (see http://www.microsoft.com/technet/security/).

When I wrote the 'Mission Impossible' series of articles about hardening *IIS* (see *VB* August 2002, p.10 and *VB* September 2002, p.8) there were a number of areas that I did not touch upon. The reason for this was quite simple: *IIS* is a very complex product, relying on several security mechanisms like file system protection through DACLs (NTFS only!) or ActiveDirectory (*IIS 5* and newer). My goal was to demonstrate that it is possible to run a secure web server with *IIS,* despite all its drawbacks and security flaws.

Every other web server is vulnerable to some security problems or its functionality is so limited that it cannot be deployed in a corporate environment. In addition, *IIS* is required by a number of other applications, so sometimes there is no choice but to install it, even if we don't really want to.

My point is that *IIS* can be a secure web platform – much like *Apache* or any other competing product. It all comes down to the knowledge of the system administrator.

WHY BOTHER?

Why should we bother with MS03-007? After all, the hot-fix is available already. The answer is not simple. First, this hot-fix is known to crash systems with a specific kernel version. Secondly, those who put a little effort into hardening their *IIS* installations were safe long before *Microsoft* published this bulletin. Finally, the WebDAV vulnerability has some interesting educational potential.

One of first rules of risk management for IT security is 'disable every non-essential service or functionality'. The rationale behind this is quite simple: there are fewer complex things to worry about and we are minimizing the risk of vulnerability exploitations.

Unfortunately, *IIS* comes with a lot of functionality enabled by default. While this can speed up the installation process

enormously it has also caused *IIS* to become recognized as one of the most insecure web servers.

Over the last few years *Microsoft* has tried to identify security problems and provide additional resources for solving them. While *IIS* is a commercial product, *Microsoft* has done well to provide a lot of additional documentation and free tools.

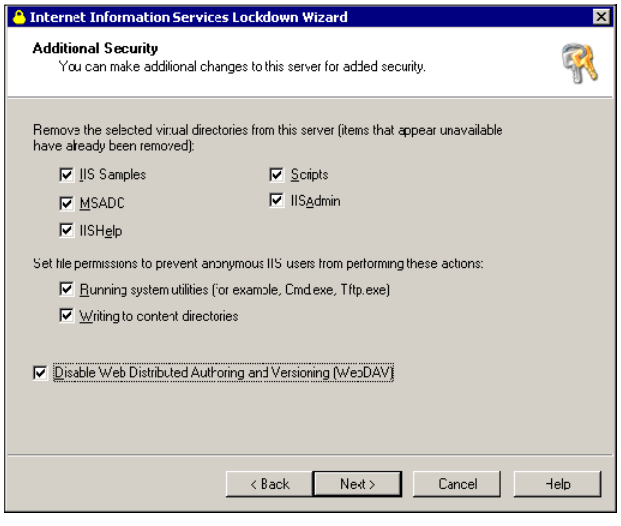
In fact it is only the ability to recompile *IIS* with a stack protection mechanism like the one provided through ProPolice or Stackguard compiler that is lacking. However, some *IIS* application developers are taking advantage of stack protection from Visual Studio .NET already. Also, there are some interesting security features in the *.NET* Framework and *IIS 6.x*.

WEBDAV, IISLOCKDOWN TOOL AND REGISTRY

One interesting feature of *IIS* (and probably one that is rarely used) is WebDAV.

WebDAV is of no practical use if we are using *IIS* as a simple web server providing static pages. Even if we wanted to use more advanced features, such as SSL, we still would not need WebDAV. The next logical step, therefore, should be to disable this feature. Unfortunately, however, many administrators leave it turned on.

Even if one uses the IISLockdown tool for *IIS* hardening there is still a chance that the WebDAV feature could be left enabled. This is a good example of why, when working with the IISLockdown tool, we should always take time to review the proposed security template. This takes just a few minutes and the gain in terms of security can be enormous.



IISLockdown tool: disabling WebDAV during template review.

Unfortunately, in the real world nothing is that simple. While users might neither use the feature nor even know what WebDAV is, an application such as *Exchange* might use it.

Again, this problem could be solved quite easily in some cases. You should not expose *IIS* FTP and web services directly if *IIS* is used as a foundation for another server application like *Exchange*. For example, Outlook Web Access (OWA) is a dangerous feature and should not be enabled on critical servers.

If we didn't disable WebDAV through the IISLockdown tool we could still do so through the appropriate registry settings. We should add a new DWORD type key, DisableWebDAV, to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters, and set its value to 1.

There is another setting that we cannot access through MMC snap-ins: MaxClientRequestBuffer value. This allows us to control the size of the URL buffer.

To limit the size of the URL buffer we need to modify the MaxClientRequestBuffer (DWORD) value in the registry under: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters.

Fortunately, *Microsoft* provides the 'Url Buffersize Registry Tool', which sets MaxClientRequestBuffer for us (see http://support.microsoft.com/).

Another thing to remember during *IIS* hardening is the installation of URLScan – it can protect *IIS* from some attacks. Note however, that just like any other tool, it cannot protect *IIS* from all kinds of attack. It is possible that a vulnerability could be triggered and exploited before URLScan has taken control (see http://www.blackhat.com/presentations/win-usa-03/bh-win-03-aitel/bh-win-03-aitel.pdf).

IP FILTERING

From the Internet Information Services MMC snap-in you have access to web server properties. Under the Directory Security tab you can limit access to web services for selected IP addresses through the 'IP Address and Domain Name restrictions' option.

While many *IIS* administrators seem to know and even use this feature, I believe there are very few who understand how it works. To demonstrate its behaviour we need to perform a small experiment.

First, with the help of netcat, we initiate a connection to the web server from an unrestricted IP:

```
# nc -v 10.0.0.100 80
Connection to 10.0.0.100 80 port [tcp/www]
succeeded!
GET /
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
<!—
            WARNING!
          Please do not alter this file. It may be
replaced if you upgrade your web server
     If you want to use it as a template, we
recommend renaming it, and modifying the new file.
          Thanks.
—>
[output has been modified]
```

As can be seen, we have access to TCP port 80 and we could perform a 'GET /' request. Now we will attempt to perform the same query from a restricted IP address:

```
# nc -v 10.0.0.100 80
Connection to 10.0.0.100 80 port [tcp/www]
succeeded!
GET /
HTTP/1.1 403 Access Forbidden
Server: Microsoft-IIS/5.0
<title>You are not authorized to view this page</
title>
[output has been modified]
```

Again, we could access the TCP port, but this time our GET request has been denied by *IIS*. It is clear now that the *IIS* 'IP Address and Domain Name Restriction' option does not filter packets before they reach *IIS* services – this makes a huge difference, as noted during the URLScan discussion.

In order to disable access to port 80 fully using only built-in *Windows* features we need to use a packet filter for network adapters. However, in *Windows NT* and *2000* limitations of the server configuration could make this quite tricky. *Windows XP* has a better (although still far from perfect) built-in packet filter.

From a simple cost/benefit analysis we can see that it is not possible to protect a server from all vulnerabilities. But even if we cannot predict vulnerabilities we still want to be able to detect such incidents.

Web and FTP server logs, together with *Windows* logs, can provide traces of incidents. Analysis of logs by hand is not a desirable method and when done this way a lot of information may be missed. One solution is to add a network intrusion detection system. Such functionality is built into *ISA Server 2000*, but *ISA* is neither the only solution nor the best solution to our problem as it is a complex product.

A much better solution is to use open-source Snort (http://www.snort.org/) for Win32 platforms. Snort can be deployed easily on *IIS* servers and run as a service. It requires winpcap library (see http://winpcap.polito.it/). Until recently this posed problems for multiprocessor machines and those with HyperThreading due to the limitations of winpcap. The 3.0 beta version of this library works correctly on such hosts. Snort has a set of rules to detect unknown buffer overflow attempts.

We can also enable the experimental preprocessor Fnord. Fnord is a polymorphic shellcode detector that is capable of detecting 'mutated' NOP opcodes for several CPU architectures including Intel and SPARC. For Intel architectures Fnord currently works against one-, two- and three-byte opcodes that could be used to replace NOP.

When *Microsoft* published security bulletin MS03-007 I received a call from one of our customers, asking whether the successful penetration of *IIS* server would result in the disappearance of system logs. This is a good example of a simple question which does not have an easy answer. Of course an attacker could have removed all logs. On the other hand, the logs could have been overwritten by the system itself if the EventViewer setup allowed such an operation. The moral of this story is that you need to watch your logs: not every attacker is clever enough to cover his tracks. By setting up additional lines of defence we might be able to both detect an incident and stop it immediately.

MISSION REPORT

It turns out that there are a few more ways to protect *IIS* servers besides the installation of hot-fixes and service packs. While writing this article I have received two different exploits for this vulnerability and one rootkit containing everything needed for successful and automatic penetration of *IIS*. Those tools await full analysis.

There are still many other aspects of *IIS* security that we haven't covered – for example *IIS* metabase ACLs (see http://msdn.microsoft.com/library/en-us/iisref/htm/FileLevelSecurity.asp). However, almost all the information required for successful *IIS* hardening is available on the *MSDN* website. It is also advisable to read the technical editions of *MS* Security Bulletins – and read them carefully as a lot of important information is hidden within.

Until the next vulnerability …

Author's note: Since this article was written, Microsoft *has published a revised MS03-007 bulletin, stating that* NT *is also vulnerable. I was using fnord preprocessor from Snort 1.9.1 which has couple of vulnerabilities and should no longer be used in a production environment. However, Snort 2.0.0 does not contain the Fnord preprocessor.*

FEATURE 1

DON'T TAKE CODE RED LIGHTLY

Larz Sherer
Independent researcher, USA

There is nothing innovative about recent strains of Klez, Yaha, Sircam and Code Red. Yet all of these worms have demonstrated unprecedented staying power on the Internet –despite the existence of patches, anti-virus signatures, personal firewall protection and Intrusion Detection technology.

Why are these threats so prolific and why do threats gain traction so quickly if all they amount to are recycled malicious code?

This article analyses the patterns of emerging malware and presents a strategy to assist network and security administrators in addressing 'new', yet old, threats.

A LOOK IN THE REAR VIEW MIRROR

It is easy to dismiss old news and, as in the case of Code Red, nobody likes to look back in the rear view mirror. Many would prefer to forget about a malware invasion that required IT staff to work overtime in order to rebuild and patch machines, and audit networks to make sure our respective environments were clear of the virus.

However, there is a great deal to be learned by examining the history and effects of the Code Red outbreak, from its inception in the summer of 2001 through to the present day.

First and foremost, the experience should remind us not to downplay or give up for dead any malicious code in the wild. Code Red was estimated by the United States Government Accounting Office to have caused upwards of 2.4 billion dollars worth of damage, with hundreds of thousands of *MS Internet Information Servers* having been infected.

WAKE UP AND PROTECT YOUR ASSETS

Who even remembers that Code Red was considered at the time to be so severe a threat that it brought *Microsoft* and the FBI together to brainstorm solutions?

Unfortunately, wake up calls seem to have a very short shelf-life. We are all driven by new priorities every day and if there is perceived to be no immediate danger, it is natural to forge ahead with those tasks that require more immediate attention.

Still, this should not preclude you from maintaining a diligent asset protection program with ongoing patch and change management processes.

There is tremendous value in keeping an eye on early warning reports of new malware threats (no matter how 'old hat' they may seem), testing these new threats and exploits whenever feasible, and ensuring that your environment is as protected from attacks as you can make it.

This involves more than merely sending an advisory email to your user-base regarding new threats and information on where to download a patch. Ongoing preventive maintenance involves written procedures based on notes you have taken and information you have collected in preparation for the day we all hope never arrives, when the unforeseen happens and your network is ripped to shreds by a malware attack.

OUT OF THE BLUE …

Lightning does strike out of the blue and, contrary to popular belief, it can strike twice. Network and security technicians must never overlook seemingly innocuous details.

Perhaps you are already familiar with that sinking feeling when you discover a compromised box on your network. That alone should be motivation enough to maintain a preventive maintenance program – but if this notion reflected reality, we would not be discussing a further re-emergence of Code Red.

The first step in preventive maintenance is adapting a proactive rather than a reactive approach to combating Internet threats. We tend to think that the most important details involve retracing what we have already done to address the last outbreak – our machines are patched, we've upgraded our gateways and desktops and laptops with the latest anti-virus signatures. What can possibly go wrong?

CODE RED REVISITED

To find out what *could* go wrong, let's look at the pattern that occurred when Code Red first emerged (see *VB*, August 2001, p.5).

The original Code Red attacked an *IIS* buffer overflow vulnerability that was discovered in July 2001. It took at least one month for the worm's author(s) to develop their code, release the worm into the wild, and for it to gather steam. It did not make an immediate impact.

As we know, some worms have the ability to propagate very rapidly, but this is not always the case – we should not be fooled by so-called 'low risk' worms. All worms have the potential to become greater problems.

In the case of Code Red a patch materialized eventually, although by the time the patch was released, the worm had cascaded across the Internet and the damage had been done.

As part of the cleanup process there was an industry collective mindshare in discussing the Code Red problem and how best to prevent it from happening again. The IT industry became fixated on bracing itself for an even greater and more sophisticated malware threat 'in the future'.

'THE FUTURE'

Well, the future is here and it seems our preventive maintenance procedures haven't changed very much. Code Red is back in 2003, and following the same pattern as it did in 2001.

Maybe it won't repeat the same scale of menace but clearly, the concept is applicable to any new threat. For example, it seems that a new Yaha strain emerges every other month. Sircam won't go away and Klez retains a stranglehold as the most hardy malware the Internet has ever seen.

All this being said, are malware techniques becoming more sophisticated? Are the propagation methods any different? Not really. We're looking at the same patterns emerging and in many cases through the same malware.

There are a few differences here and there but, by and large, it's all old hat and we're just as vulnerable to a network shredding now as we were in 2001.

WHAT ARE WE DOING WRONG?

Aside from negligence in not keeping up with our best intentions for preventive maintenance, what are we doing wrong?

We're more sensitive to impressing security measures upon end-users. We have a stronger appreciation for taking network maintenance seriously. We have improved protection at the gateways and other vectors into a network. Within most companies security expenditures have increased from year to year.

The industry is more open than ever before when it comes to the disclosure of vulnerabilities as well as the development and distribution of patches.

Even *Microsoft* has made a commitment to greater security as it lumbers toward another platform release. Will *Windows 2003 Server* and *IIS 6* solve security issues or bring a new set of problems to be dealt with? It all remains to be seen.

MORE OF THE SAME

The age of polymorphic malware is upon us, and we can expect more of the same: intelligent algorithms to identify IP addresses, backdoors sending broadcasts to other servers with the same vulnerabilities as the infected host.

Even if the malware is not successful in locating suitable new hosts, the replication process is causing the most harm – in fact, this causes more bottlenecks on the Internet than spam.

Experts predicted that the worms of the future would leave us with no lead time to respond to new threats after a vulnerability is published. To an extent, that prediction has come true.

It is not uncommon for the speed of saturation to be extraordinarily rapid. For example, SQL Slammer (see *VB*, March 2003, p.6) sought targets by broadcasting connection requests to random IP addresses in a rapid manner. Although the worm itself was applicable only to *Microsoft SQL Server*, and *Microsoft* had released a patch for the vulnerabililty six months earlier, the rate of infection was very high.

NEW APPROACHES

Granted, it is not possible to stop every worm outbreak, but records over the last two years show clearly that new approaches are needed to deal with the proliferation of pattern malware attacks.

This is especially true with regard to repeat offenders who have no business cropping up every few months with a new variant. There may be only subtle differences among strains but malware is a sophisticated and intelligent menace.

The only way to understand the threat is to see it in action, and study its behaviour in a contained environment.

CHALLENGES FOR IT STAFF

As a network or security administrator, it is not in your best interests to shy away from testing suspicious programs to gauge their impact on your network.

Administrators should take note of patterns in file names associated with particular malware and utilize security software that makes use of MD5.

MD5 is an algorithm that produces 128-bit message digests that are unique to every application. Computationally, it is infeasible for applications to have the same MD5 signature. Therefore, MD5 can be used to verify data authenticity and to serve as the primary instrument of file comparison and file detection, as well as a determinant of file corruption and tampering.

The practice of replicating user experiences in a safe environment is invaluable to your own education and will come into play as you continue to flesh out the priorities of your defence strategies. Speed and accuracy are critical. Having a test environment ready may help you win the day.

If you really want to get serious about malware testing, build a lab, segregate it from your company's network and use it exclusively to test malware, spyware and adware.

MALWARE RESPONSE FRAMEWORK

Dealing with malware at an early stage will prevent a great number of problems and a great deal of frustration later. There is no substitute for adopting an ongoing preventive maintenance attitude. While there may never be an absolute 'magic bullet', nothing should be left to chance.

The following are some suggestions to be used in building or adding to your malware response framework:

- Devise rapid response checklists and workflows that *anyone* can follow. The hardest part of this is finding the time to keep them updated. Structured documentation goes a long way.
- Have a GHOST server or another image software server at the ready to warehouse the most recently updated operating systems, service packs and security fixes. Most importantly, make sure the builds are clean. If you suspect that an image build is compromised, it is advisable to err on the side of caution and build it again.
- Maintain a secure FTP server with backups of image builds, diagnostic tools, bookmarks, and so on. Make sure that everything you need is ready for rapid redeployment in case disaster strikes and your main repository is cut off.
- When you install a patch, test its effectiveness. This is an extra step that most technicians don't bother to take. It can be a little time-consuming, but it's all too easy to place our faith in a vendor to fix a problem simply by installing a patch.
- As evidenced by the strength of malware, sometimes patches don't fully solve a problem, they just cover it up. Second-wave vulnerability discoveries are common. It takes more than one layer of shielding to thwart some of the more resilient malware.
- End-users will be independent, but that should not stop you from training and educating them on effective desktop security usage.
- The more you impress handy tips upon your end-users, the less prone they will become to making mistakes that can impact your network. Familiarity with new security policies must be reinforced.

Finally, don't take compromises personally. You won't win every battle. Take careful notes and make the effort not to repeat the mistakes of the past. Become a stronger technician with each experience.

FEATURE 2

OUT OF AFRICA...

Martin Overton
Independent researcher, UK

Africa is often referred to as the cradle of the human race (see http://www.kenyalogy.com/eng/info/histo.html); it is also the birth place of the 'Advance Fee Fraud', aka '419 scams/frauds', aka the 'Nigerian Money scam/fraud'. The humble 419 scam, known as 'The Game' or 'The Plan' by those who practise it, has been around for many years in one form or another. In fact, some claim it should be considered an African 'cottage industry'. But the old tried-and-trusted formula has changed recently, as has the level of media interest in both victims and perpetrators of the scam.

WHAT'S IN A NAME?

419 frauds combine the threat of impersonation fraud with a variation of an advance fee scheme. A letter or email from Nigeria (originally, but from just about any country now), offers the recipient the 'opportunity' to share in a percentage of millions of dollars that the author – often a self-proclaimed government official, doctor, engineer, bank official, religious minister etc. – is trying to transfer out of the country illegally with a little help from their new-found friend and benefactor: the recipient.

The victim is encouraged to send information to the author of the fax/letter/email, such as blank letterhead stationery, bank name and account numbers and other identifying information.

The scheme depends on convincing a willing victim, who has demonstrated a 'propensity for larceny' by responding to the invitation, to send money to the author of the letter in several instalments of increasing value.

Often, the requirement to pay taxes, bribes to government officials, and legal fees are described in great detail, with the promise that all expenses will be reimbursed as soon as the funds are spirited out of Nigeria. In fact, the millions of dollars do not exist and the victim ends up with nothing.

In many cases the victim is encouraged to visit Nigeria or a neighbouring country, and is smuggled across the border. Even more money can then be extorted, supposedly to enable the victim to get out of the country they have entered illegally.

Should the victim stop sending money, the perpetrators have been known to use the personal information they were sent to impersonate the victim, draining bank accounts and credit card balances until the victim's assets are exhausted.

Most law-abiding citizens identify the 419 emails/letters as hoaxes/scams. However, millions of [insert your local currency here] are lost annually as a result of these schemes around the world.

The Nigerian government is not sympathetic to those who have been fooled by the schemes, since the victim has effectively conspired to remove funds from Nigeria in a manner that is contrary to Nigerian law (even though no such funds actually exist).

The scheme violates section 419 of the Nigerian criminal code, hence the label '419 fraud' although the fraud is now common outside of Nigeria too.

There are many reports, from both the UK and the USA, that a surprising number of unsuspecting victims have lost significant amounts of money, been lured to the originating country where they have been imprisoned, tortured and occasionally even lost their lives (so much so that the 419 fraud has been the subject of an FBI warning, as well as one from the US Secret Service).

ORIGINS

The scam is claimed to have been started in the 1980s, in Nigeria. Some investigators link the start of the scam to a downturn in Nigeria's oil industry in that decade (see http://www.the-ria.com/419.html). In the early days, the offers were sent in letters through the postal system, and later via fax machines. As the Internet became ubiquitous the scam started to transfer to the current email-based versions.

WELL TRAVELLED?

Although the Advance Fee Fraud was born and brought up in Nigeria it has begun to travel. I have seen versions of 'The Game' from the following countries (in addition to Nigeria):

United Kingdom	Canada
Ivory Coast	Dubai
South Africa	Yugoslavia
Netherlands	Sierra Leone
Zimbabwe	Philippines
Angola	Taiwan
Togo	Germany
DR Congo (Zaire)	Iraq

'You would [be] shock[ed] at how many wad want something more for nothing. Greed carry their head[s] and turn[s] them foolish.'

'The Game' perpetrator
From www.wired.com

THE GAME

Let us look at the perpetrators' perspective. Why do they do it? For many 'The Game' is a way of life, a business, nothing more, nothing less. Like their victims they are driven by greed. Unlike their victims they justify the scam by claiming to be taking money from those who have too much already (known as 'wad' [rich people]) and who are seen as being so greedy that they deserve to be fleeced in this way.

In one of the very few known interviews with those that run 'The Game' (see http://www.wired.com/news/print/0,1294,53818,00.html) one of the perpetrators said: 'You would [be] shock[ed] at how many wad want something more for nothing. Greed carry their head[s] and turn[s] them foolish.'

Those who run the scam use the derogatory term 'mgbada' when talking about their victims.

VICTIMS

Unfortunately, it is not always only the victim who suffers the consequences of the scam; there are often innocent bystanders who end up being part of the 'collateral damage'. On 19 February 2003, a Nigerian diplomat was shot dead in Prague by a Czech pensioner who allegedly had been taken in by 'The Game' (see http://news.bbc.co.uk/1/hi/world/europe/2780259.stm).

Other documented cases demonstrate that some 'mgbada' have been quite happy to beg, borrow or steal to play 'The Game'.

A 59-year-old female employee of a Michigan law firm was taken for $2.1m by Nigerian 419 fraudsters promising a percentage ($4.5m) of $18m in return for her help in getting the money out of Africa. Allegedly, she funded the entire operation with the contents of her employers' bank account, and was only rumbled when a cheque for $36,000 bounced. The woman now faces up to three years in jail on 13 counts of wire fraud (see http://www.theregister.co.uk/content/archive/27243.html).

A businessman in the US was conned out of $750,000 by Nigerian 419 scam artists. The twist here is that $250,000 of the money he handed over did not belong to him. In this

case the fraud/scam victim had been scamming and defrauding seven of his own friends to raise some of the 'Advance Fees/Bribes' (see http://www.theregister.co.uk/content/28/29673.html).

THE WORLD WANTS TO BE DECEIVED

According to figures from the UK's National Criminal Intelligence Service (NCIS) at least 150 Britons were defrauded in 'The Game' in 2002. The money defrauded from them totalled a staggering £8.4m, which works out at around £56,675 each. One victim travelled to Africa to collect his percentage, and got more than he bargained for, when he was beaten and tortured instead (see http://www.theregister.co.uk/content/6/29536.html).

'If fifty thousand people do a foolish thing, it is still a foolish thing.'

Anatole France (modified)

The US Secret Service reported that, in June 1995, an American who had been trying to get his money was found murdered in Lagos. Numerous other people have been reported missing.

The FBI reported that Internet fraud in the US took $35 million from the many victims (48,252) that took the bait in 2002. In 2001 only $11 million was taken from victims. 'The Game' accounted for the largest monetary losses of all reported frauds (an average of $3,864 per victim). It seems that this is a problem that is still growing.

Indeed, a modified version of a famous quotation credited to Anatole France (1844–1924) seems to sum up those that fall for this fraud: 'If fifty thousand people do a foolish thing, it is still a foolish thing.' More stories can be found at http://www.crimes-of-persuasion.com/.

WHY DOES IT WORK?

The scam works because it relies on social engineering. In this case the scam focuses specifically on greed, altruism (sometimes) and a terminal lack of scepticism. Any shred of scepticism that does persist tends to be overridden by the large sums of money that are promised.

WHAT NEXT?

What new twists can we expect 'The Game' to take? We have already seen the following themes: Diamonds, Oil, Land, Illness, Cash, Gold, Lottery, Online Auctions and Religion.

The biggest change to 'The Game' appeared early last year with the advent of the 'Lottery' variant. This works in much the same way as the original fraud, but there are smaller amounts of money involved (one million Euros rather than 20–30 million US Dollars) and many other countries, (for example the Netherlands, Spain and Canada). So, if you receive an email claiming that you have won a lottery jackpot for a lottery that you don't remember taking part in, be very sceptical.

It may well be the case that we will see more of these fairly major re-workings of 'The Game'. I won't list my ideas as to possible new variants, as I do not want to be responsible for the birth of new variants of this fraud.

FIGHTING BACK

The advice from the UK's National Hi-Tech Crime Unit and from the Nigerian Government is that, when you receive one of those 419 scam emails, you should forward it to the 'abuse@' address of the ISP involved. Or delete it.

However, a number of people have ignored this advice and taken it upon themselves to 'fight back'. Some entertaining results can be read online at http://www.fattibastardo.com/fraud.html and http://www.savannahsays.com/kizombe.htm.

'If something seems too good to be true, it probably is.'

Should you wish to forward 419 messages to your country-specific law enforcement fraud section, a list of the appropriate email addresses can be found at: http://home.rica.net/alphae/419coal/.

CONCLUSIONS

The humble 419 Advance Fee Fraud is alive and well and seems to be producing offspring to perpetuate the species. Are we seeing the fraud equivalent of the transition from *Homo erectus* to *Homo sapiens* or is that evolutionary leap yet to come for the 419 genus?

Most days I receive at least two 419 emails, while some days I get well over a dozen. It always raises a smile when I receive them at my 419@arachnophiliac.com email address. Is it a case of irony, serendipity, coincidence, fate, karma or maybe a 419 conspiracy? A searchable database of 419 and its many variants can be found at: http://arachnophiliac.com/hoax/419_search.htm.

Finally, here's another mantra that you should encourage your staff, friends, loved ones and acquaintances to learn: *'If something seems too good to be true, it probably is.'*

COMPARATIVE REVIEW

REDHAT LINUX

Matt Ham

It is thirteen months since the first *Linux* comparative review graced the pages of *Virus Bulletin* (see *VB* April 2002, p.16). During those months the operating system has enjoyed a significant rise in popularity, so it came as something of a surprise to receive only 11 products for this review – the same number as last time. With the production of an on-access scanner for a *Linux* product being trickier than on more homogeneous operating systems, there were no VB 100% awards given in the previous *Linux* comparative review. (A cynical reviewer might link these two facts.)

A newcomer to the comparative reviews this month is *H+BEDV*, whose product *AntiVir* has been a feature of the anti-virus landscape since time immemorial. The only company, as far as I am aware, to give away branded beer as a marketing gimmick, I have good reason to wish them a long stay in the regular line-up for comparative review.

THE TEST SETS

The test sets compiled for this review were derived from the March 2003 test sets. With the deadline for product submission being only days after the release of a provisional WildList, this is probably one of the tougher tests for vendors – usually there are a couple of weeks' grace between the release of the WildList and the submission deadline.

Since the last comparative review was carried out before the WildList had stabilised to a new regular production schedule, there were a large number of changes to the In the Wild (ItW) test set. Some of the changes had been anticipated, while others seemed, initially at least, downright outlandish.

On-demand tests	ItW File		Macro		Polymorphic		Standard		Linux	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%	Number missed	%
Alwil avast!	0	100.00%	3	99.56%	160	91.22%	11	99.55%	40	59.33%
DialogueScience Dr.Web	5	99.51%	0	100.00%	0	100.00%	0	100.00%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	4	99.82%	4	99.73%	6	66.67%
F-Secure Anti-Virus	0	100.00%	0	100.00%	0	100.00%	3	99.86%	7	65.00%
GeCAD RAV	0	100.00%	0	100.00%	35	97.61%	0	100.00%	0	100.00%
H+BEDV AntiVir	7	99.23%	47	99.42%	753	83.28%	52	97.79%	44	0.00%
Kaspersky KAV	0	100.00%	0	100.00%	1	99.92%	0	100.00%	0	100.00%
Norman Virus Control	1	99.76%	56	98.95%	179	91.25%	12	99.53%	5	85.67%
Sophos SWEEP	0	100.00%	0	100.00%	60	95.79%	15	99.31%	14	46.67%
Trend Server Protect	0	100.00%	0	100.00%	214	95.81%	11	99.59%	7	60.00%
VirusBuster VirusBuster	0	100.00%	3	99.93%	160	89.13%	11	99.52%	40	6.67%

On the way out of the test sets were a motley collection of Win32 viruses and O97M viruses. The fact that the problematic W32/CTX has finally departed the test sets will be a reason to rejoice in some camps, though some may shed a tear over Junkie for nostalgia's sake. Replacing these were a rush of Win32 mailers and network-aware pests, including nine new W32/Opaserv variants since the last comparative review.

The surprise amongst the newcomers was the large number of W95 viruses making an appearance for the first time. Six W95 specimens were added to the test sets, including a further variant of an old stalwart, W95/CIH.1049. Quite what could have caused the resurgence of infected *Windows 95* machines? In fact, there is no such resurgence, since most of these viruses throw up errors by the ton if run on any *Windows 95* machine. *Windows 98* could tempt some to run, somewhat half-heartedly. However, the mass of additional DLLs required by some of these viruses leaves a question mark as to quite how they have entered the wild.

The *Linux* test set was much the same as that used in the last *Linux* review. Internal files from malware which arrives in large packages, e.g. Linux/Lion, were removed however, since they were giving the test sets an undeserved aura of importance as a result of their bulk. The *Linux* files in the test set are present for one reason: to determine whether products are even attempting to detect *Linux* malware. As such, the files can be divided into two main categories: the archive stored worms and the ELF format file viruses.

LINUX PECULIARITIES

The *Linux* platform is a difficult one for which to design an on-access scanner, on account of the flexibility of the operating system. The number of different *Linux* kernels is as grains of sand in a desert, and offers no solidity for those who require a firm and unchanging environment.

The flexibility of the operating system is one of the strongest features of *Linux*. As users wish to perform ever more cunning tricks on a *Linux* machine, however, the number of details required as to what exactly is or is not in the kernel increases significantly. Interrupting file access is just such a sneaky trick – and one required by on-access scanners.

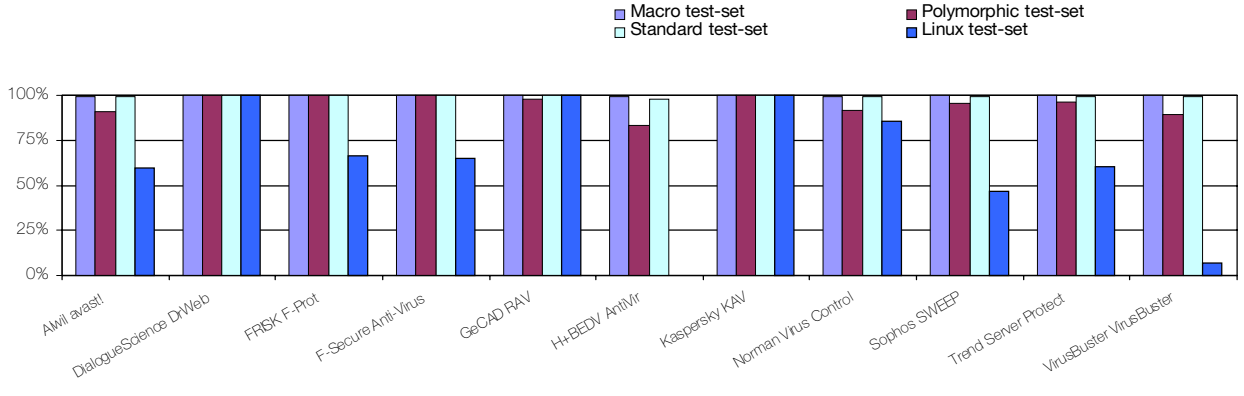
The developers of products in this review have used a number of techniques to overcome this kernel dependency. The method requiring least user interaction is that which uses a non-kernel component as a vector for filtering. The designated on-access test scenario in this review was file opens through *Samba*, therefore it was not surprising to see *DialogueScience* and *GeCAD* using *Samba* to pass files for on-access scanning. *F-Secure* offers a daemon which can intercept http GET requests in a similar fashion, though this was not tested in the review.

However, this method of scanning is somewhat limiting in that file access from other sources can occur with no checking, and such outside access can play havoc with the scanning cache, if vendor documentation is to be trusted. The use of a kernel driver can allow all file access to be filtered but is, as stressed earlier, kernel-dependent. *Trend Micro* has a sufficiently large user base that a selection of kernel modules for popular kernel constructions is offered.

This is not so much use to the inveterate tinkerer, however, who must compile his own source code for the kernel module. *H+BEDV* and *Alwil* use an open source basis named *Dazuko* for this process. The resources associated with this project were sufficient to allow easy and successful compilation of the source.

Kaspersky Lab also supplies source for its kernel module – although the documentation provided, and the peculiarities of *RedHat Linux,* made this a task which was not surmountable within the allocated timeframe. *Kaspersky*'s suggestion for obtaining sufficient information for guaranteed

Detection Rates for On-Demand Scanning



installation is to compile a kernel from scratch – which seems a rather high expectation for a user concerned with uptime and preserving a machine in a state of stabilty.

Alwil avast! 4.0 (beta1)

ItW Overall	100.00%	**Macro**	99.56%
ItW Overall (o/a)	100.00%	**Standard**	99.55%
Linux	59.33%	**Polymorphic**	91.22%

Alwil's *avast!* was submitted as a beta version of the software, which can be daunting when a review is to be performed. The beta status of the product may explain the slight awkwardness of the installation procedure, which required the execution of two shell scripts in different locations. A rather less avoidable part of the installation procedure was the need to compile the *Dazuko* source code – a relatively easy task once the appropriate website (http://www.dazuko.org/) had been paid a visit.

Once the program was up and running, scanning commenced, only to end speedily. The culprit was a segmentation fault caused by one of the Linux/Bliss samples in the test set. This caused the scan process to crash on demand repeatedly and the offending sample was removed from the set for this scan and recorded as a miss. The same file caused problems on access. In this case, however, there were no outward signs of the scanning failure – the engine simply ceased operating after this file had been scanned. Again the sample was noted as a miss, and once the scanning daemon had been restarted, no further problems arose.

Other than this issue, scanning results were good. Large differences in performance on *Linux* samples on access and on demand can be attributed to different treatment of archives under these two scenarios. With full detection both on access and on demand, and no false positives, *avast!* is the first product in this review to receive a VB 100% award.

DialogueScience Dr.Web for Linux 4.29.7

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

Dr.Web arrived as two packages, one for the main on-demand scanner and another for the *Samba* daemon-based scanner, both of which were in RPM format and installed with no problems. It was notable in this review that the products were split roughly between those which installed a path or link to their executables and those which leave this task to the person installing the software. Both methods will have their advocates – *Dr.Web* is one of those in which the onus is on the user to perform the task.

Installation and activation of the *Samba* scanner was simple enough, requiring only a single-line addition to the smb.conf file for each share to be protected. What was noticeable, however, was that access to files on the *Samba* share slowed noticeably when the scanning daemon was in place. Despite this sluggishness on access, scanning efficiency was close to the usual *Dr.Web* levels – but fell short of full detection. The files that were missed were the five samples of W95/Bodgy in the ItW test set, denying *DialogueScience* a VB 100% award. Less of a surprise were the presence of the now somewhat traditional 15 suspicious files in the clean test set.

FRISK F-Prot Antivirus for UNIX 3.13a 3.13.2

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	n/a	**Standard**	99.73%
Linux	66.67%	**Polymorphic**	99.82%

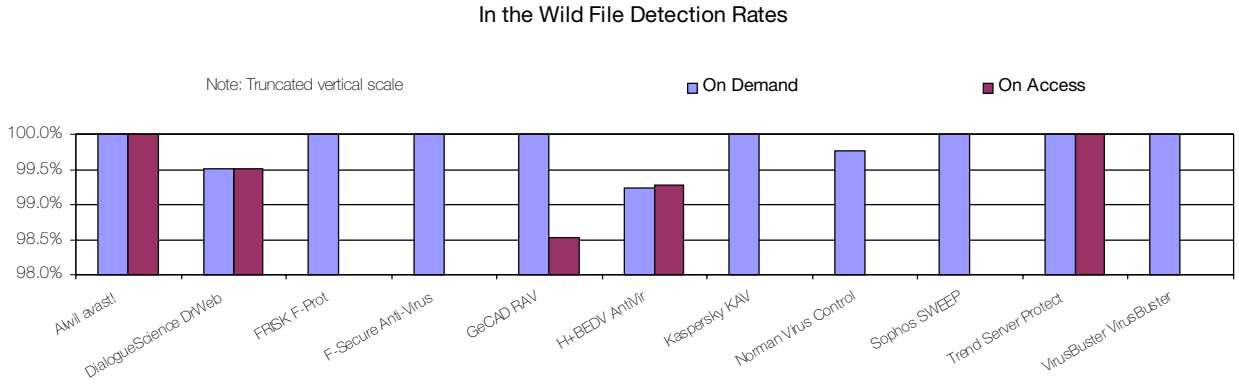
F-Prot Antivirus was another product to offer the package in RPM format, and as a result was simple to install. An on-access component is supplied with the product, though this was not tested since it filters only http GET requests, rather than the fopen/fclose accesses which are tested in *VB* protocols. Such a method of access filtering thus lies outside the scope of comparative testing. This caveat also applies to other products in this review. Several have on-access features which lie outside the scope of the review, and the lack of a VB 100% award in this test is relevant only within the limitations set by the need to keep the test procedures practical.

The *FRISK* product showed very good detection rates across all test sets. A sizeable proportion of the small number of misses seen was attributable directly to the fact that *F-Prot Antivirus* has archive scanning disabled in its default setting. This explains misses of the W32/Heidi virus and also for the *Linux* worms which distribute themselves as archives. In contrast, *Linux* ELF infectors were detected perfectly.

F-Secure Anti-Virus for Linux Server 4.50.2111

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	n/a	**Standard**	99.86%
Linux	65.00%	**Polymorphic**	100.00%

The package supplied for the installation of *F-Secure* was in a proprietary encrypted format, requiring the registration



key for installation. This format allowed a more interactive installation procedure than that seen for the RPM-based installers. The installation procedures can be divided into three camps. The first is the bare-bones approach, where scattered shell scripts, manually edited configuration files, and a healthy attention to man pages are the order of the day. A second camp opts for RPM packages – which, although very easy to use, tend to leave the user rooting around in the background when fine-tuning of the configuration is required. The approach chosen by *F-Secure* may not adhere to any industry standards, but for simplicity of both installation and configuration it certainly has its advantages.

As expected from a product using two engines, the detection rates for *F-Secure*'s product were at their usual high level. Files were missed either as the result of not scanning archives by default, or of choosing not to scan file extensions which are only rarely host to dangerous code.

GeCAD RAV for Linux 8.4.2

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	98.54%	**Standard**	100.00%
Linux	100.00%	**Polymorphic**	97.61%

The installation method of *GeCAD*'s *RAV* was the RPM format, with an on-access scanner being supplied for *Samba*. This gave, in total, four RPMs to be installed, with the requirement that these be installed in order of their dependencies. Although this order was fairly easy to guess, this was a minor irritation.

When scanning on demand, the *RAV* engine had no problems whatsoever in the test sets, missing samples of W32/Fosforo, with the remainder of misses being a few other incompletely detected viruses in the polymorphic set. Matters were trickier in the on-access tests. *GeCAD*'s documentation states that access to the shared drive performed by methods outside the *Samba* functionality could cause problems for the scanner and this seemed to be the case even when only one or two files were concerned. Being more conscientious about methods of access to the shared resource gave several on-access scans which performed oddly and it took some patience to reach a final test result.

The final result was identical to that seen on demand, with the exception of misses on X97M/Jini.A1, W32/Gibe.B and W32/Lovgate.C. Several more tests repeated under the same conditions demonstrated that this was a reproducible set of misses. Since these are all in the ItW test set, *RAV* was denied a VB 100% award on this occasion.

H+BEDV AntiVir Workstation 2.0.7

ItW Overall	99.23%	Macro	99.42%
ItW Overall (o/a)	99.27%	Standard	97.79%
Linux	0.00%	Polymorphic	83.28%

This is another product that uses *Dazuko* – which is not a surprise, since *H+BEDV* has played a significant part in the production of this resource. With the practice obtained from installing *Dazuko* for *avast!* this part of the installation procedure proved the easiest aspect. The program installation itself was slightly complicated by the fact that the archives supplied had been produced on a *Windows* machine, this causing changes to the case of several file names.

H+BEDV does, however, offer one of the more interactive shell scripts for product installation, which allowed easy detection of which files should be called and their locations, since it declared the source of any installation errors. This was, of course, very useful for configuring the program after installation as well as this early negotiation of problems. One problem which proved insurmountable was the issue of a licence key, since none of those supplied could be persuaded to work. However, an unlicenced copy of the

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files		Linux Files	
	Time (s)	Throughput (MB/s)	FPs [susp]	Time (s)	Throughput (MB/s)	FPs [susp]	Time (s)	Throughput (MB/s)	(s)	Throughput (MB/s)	(s)	Throughput (MB/s)
Alwil avast!	107.0	5111.5		12.4	6397.9		56.0	2846.7	14.8	5041.0	12.8	3269.4
DialogueScience Dr.Web	149.0	3670.7	[15]	9.3	8530.5		72.0	2214.1	11.6	6431.7	14.3	2926.5
FRISK F-Prot	77.0	7103.0		3.5	22666.8		39.0	4087.6	4.8	15543.2	6.7	6246.0
F-Secure Anti-Virus	181.0	3021.7	[1]	11.2	7083.4		185.0	861.7	34.0	2194.3	5.3	7895.9
GeCAD RAV	287.0	1905.7		4.6	17246.5		132.0	1207.7	4.5	16579.4	7.5	5579.8
H+BEDV AntiVir	101.0	5415.2	1	48.0	1652.8		83.0	1920.7	8.9	8382.9	10.3	4063.0
Kaspersky KAV	148.0	3695.5		11.3	7020.7		80.0	1992.7	19.1	3906.2	25.9	1615.8
Norman Virus Control	129.0	4239.8		9.0	8814.9		75.0	2125.6	17.0	4388.7	26.0	1609.6
Sophos SWEEP	59.0	9270.0		9.5	8350.9		37.0	4308.6	10.2	7314.5	4.9	8540.5
Trend Server Protect	93.0	5881.0		8.6	9224.9		45.0	3542.6	15.3	4876.3	18.4	2274.4
VirusBuster VirusBuster	163.0	3355.4		6.1	13005.5		93.0	1714.2	10.7	6972.7	3.7	11310.4

software lacks only logging to file and the ability to perform actions on detected viruses. Since logging of infections to syslog is supported, this was used for detection analysis.

As a product that is new to the testing process, certain misses were more or less expected, such as ACG.A and ACG.B. More concerning was the miss of W95/Bodgy In the Wild, which was sufficient to deny *H+BEDV* a VB 100% award.

Kaspersky KAV for Linux 4.0.30

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	n/a	**Standard**	100.00%
Linux	100.00%	**Polymorphic**	99.92%

KAV for Linux arrived as a set of files, one of which is launched as a proprietary installer and searches for the others. This mechanism did not seem to be implemented perfectly, though after two or three tries of various command line options it became apparent that stating the target file explicitly was a much more reliable method of initiating installation.

On-demand detection was very good indeed, with only a single sample of W32/Etap being missed over the entire test set. An on-access scanning module was also supplied, though this was distinctly more problematic. With the installation of *Dazuko* having provided practice in the complexities of kernel modules, it was expected that *Kaspersky*'s module would prove just as easy to produce. Unfortunately this was not the case, with numerous attempts to compile the module ending in failure. The documentation supplied accepted that this was a likely outcome, given the nature of some *Linux* distributions and their kernel config files. The suggested remedy was to recompile the kernel so as to have a known version to work with. However, given the time constraints in testing, and the specific kernel stipulated for the test protocol, this remained untested.

Norman Virus Control Version 5.53.02

ItW Overall	99.76%	**Macro**	98.95%
ItW Overall (o/a)	n/a	**Standard**	99.53%
Linux	85.67%	**Polymorphic**	91.25%

Norman's product uses the RPM method of installation, resulting in an uneventful process. In fact, 'uneventful' sums up the performance of *Norman Virus Control* in the testing process, with no problems being encountered. Misses for the product were well spread among the test sets, with the In the Wild miss of W32/Zoek.D being the only surprise.

Sophos SWEEP 3.68

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	n/a	**Standard**	99.31%
Linux	46.67%	**Polymorphic**	95.79%

The *Sophos* product is installed by means of a shell script, which is not quite as intelligently constructed as it might be. The documentation supplied states that, in order to run the on-demand scanner alone, no users need to be added, though if the product is to be used with clients on other machines, a *SWEEP* user must be installed. However, the installation script will not run unless this user is created manually, despite there being no need for the user other than to satisfy the script's demands.

Once past this niggle, installation and scanning went smoothly, and detection was as expected with one exception: clearly some engine tweaking has been going on at *Sophos*, since the detection of polymorphic viruses has improved noticeably since the last test.

Trend Server Protect Linux 1.1

ItW Overall	100.00%	**Macro**	100.00%
ItW Overall (o/a)	100.00%	**Standard**	99.59%
Linux	60.00%	**Polymorphic**	95.81%

Trend's *ServerProtect* is the only product to have been supplied as a graphical application in this test. *GeCAD* and *FRISK* offer graphical front-ends for their home-user *Linux* software, though these were not submitted (there may be others of which I am unaware).

The use of a graphical interface requires a little preparation on the part of the user. The interface uses the http protocol to communicate with the *ServerProtect* engine, and requires Java functionality which is not a standard installed package for *Mozilla*. After installation of the appropriate Java RPM a few symbolic links must be created.

The installation packages provided by *Trend* can accept a variety of pre-made kernel modules, the method here being a forced install with standard modules and then replacing these modules with those appropriate for the kernel present on the machine in question.

After this set of procedures is completed, however, the GUI offered through *Mozilla* was one which has all the features standard on any of the other *Trend* GUIs seen on other platforms. Although not used as such in this test, the interface can, of course, be used by a browser from any machine which is allowed access to do so – which would be a more usual method of using this functionality.

Such a pretty face, though, is pointless if there are no brains behind it, and *ServerProtect* did not disappoint on this front. With no false positives and full detection In the Wild, *ServerProtect* gains a VB 100% award. One problem which was noted, however, was that on one scan of the whole test set the server protect chain of command was broken at some point, and the *ServerProtect* GUI had to reconnect in order to regain control of the application.

VirusBuster VirusBuster LINUX 7.647

ItW Overall	100.00%	**Macro**	99.93%
ItW Overall (o/a)	n/a	**Standard**	99.52%
Linux	6.67%	**Polymorphic**	89.13%

VirusBuster uses the install script method of installation, which produced errors when run. The error messages were perhaps not designed to be read in a default KDE terminal window however, as cyan-on-white made the messages all but invisible to the naked eye. Some repositioning of the files solved this problem, and thereafter *VirusBuster* performed without a hitch. Scanning results were good in all but the *Linux* test set, in which only the cross-platform W32/Lindose virus was detected. With such a result it might be suspected that the detection of *Linux* native malware is not a high priority for *VirusBuster*.

CONCLUSIONS

The last *Linux* comparative was a sorry tale indeed, with all of those products that offered an on-access scanner proving to be untestable for one reason or another. The change of review platform from *SuSE* to *RedHat* has probably helped the developers somewhat, *RedHat* having a larger user-base to discover potential pitfalls. However it is the ever-increasing popularity of *Linux,* both in businesses and amongst home users, that is a more significant factor. The situation is eerily similar to the early days of *Windows* scanners – perhaps next year the full line-up will offer on-access scanning functionality.

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 *RedHat Linux 8*, kernel build 2.4.18-14 and *Samba* version 2.2.5. 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/2003/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 AND NEWS

EICAR 2003 takes place 10–13 May 2003 in Copenhagen, Denmark. Check http://conference.eicar.org/ for details.

Black Hat Europe 2003 will be held 12–15 May 2003 at the Grand Krasnapolsky, Amsterdam, the Netherlands. Trainings take place 12–13 May and Briefings 14–15 May. For more information see http://www.blackhat.com/.

The DallasCon Wireless Security Conference takes place 24–25 May 2003 in Plano, Texas. A two-day wireless security course precedes the conference, including hands-on lab experience. For full details see http://www.DallasCon.com/.

Infosecurity Canada Conference and Exhibition takes place 4–5 June 2003 in Toronto, Canada. For registration and exhibitor details see http://www.infosecuritycanada.ca/.

The 15th Annual Computer Security Incident Handling Conference takes place 22–27 June 2003 in Ottawa, Canada. For more information see http://www.first.org/conference/2003/.

NetSec 2003 Conference and Exhibition takes place at the Hyatt Regency, New Orleans 23–25 June 2003. For the conference programme, exhibitor list and registration information, see http://www.gocsi.com/.

The Third World Conference on Information Security Education takes place 26–28 June 2003 in Monterey, USA. For details see http://cisr.nps.navy.mil/wise3/.

The Black Hat Training and Briefings USA 2003 take place 28–31 July 2003 at the Caesar's Palace hotel, Las Vegas. For full details and registration see http://www.blackhat.com/. DEFCON 11 will take place 1–3 August 2003 in Las Vegas, following the Black Hat Training and Briefings. See http://www.defcon.org/.

COMDEX Canada 2003 will be held 16–18 September 2003 in Toronto, Canada. See http://www.comdex.com/.

The 13th *Virus Bulletin* International Conference and Exhibition (VB2003) takes place 25–26 September 2003 at the Fairmont Royal York hotel in Toronto, Canada. For exhibtion details, contact Bernadette Disborough on +44 1235 555139 or email vb2003@virusbtn.com. For more information and online registration see http://www.virusbtn.com/conference/.

The 5th NTBugtraq Retreat takes place in the days immediately following the Virus Bulletin conference in Ontario, Canada. A welcome event on the evening of 26 September will be followed by the Retreat from 27–29 September 2003. Full details can be found at http://www.ntbugtraq.com/party.asp.

Black Hat Federal 2003 takes place 29 September to 2 October 2003 in Washington D.C. For more information and online registration see http://www.blackhat.com/.

InfowarCon 2003 takes place 30 September to 1 October 2003 in Washington D.C. Military leaders, political forces, academics, and industry members will discuss the concepts of the latest on-going initiatives in the Homeland Security and Critical Infrastructure Protection communities. For details see http://www.infowarcon.com/.

The Workshop on Rapid Malcode (WORM) will be held 27 October 2003 in Washington D.C. The workshop aims to bring together ideas understanding and experience relating to the worm problem from academia, industry and government. See http://pisa.ucsd.edu/worm03/.

COMDEX Fall 2003 takes place 15–20 November 2003 In Las Vegas, USA. See http://www.comdex.com/.

***ICSA Labs* reports that businesses recovery costs from virus attacks are increasing**. The *Labs*' annual Virus Prevalence Survey found that in 2002 the average estimated cost of recovery increased from £45,000 in 2001 to £52,000 in 2002. To read the survey see http://www.icsalabs.com/.

***Eset Software* and *Canon System Solutions Inc.* have announced an alliance** aimed at developing virus detection solutions for the Japanese market – the first of which is a Japanese version of *NOD32*. See http://www.nod32.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:
UK £195; Europe £225; International £245 (US$395)

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 www.virusbtn.com