ISSN 0956-9979 APRIL 2000

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **ZIP up tight!** For the first time, this month's Comparative Review includes archive file detection rates. Fifteen products line up for testing on p.14.

• **Three for the price of one:** the virus analysis column provides a close look at something old, something new and something borrowed. Email worms and an updated virus variant come under the microscope on p.6.

• **Collared by the watchdog:** matching the tone of this month's Comment page, David Harley asks some searching questions about the anti-virus industry's bad reputation. Find out if it is justified on p.10.

CONTENTS

COMMENT

Is the WildList too Tame?

[*Chris Scally, a* VB *subscriber since December 1990, is the Network Access Security Manager for a major financial institution in Dublin, with particular responsibility for its anti-virus strategy. This comment reflects his personal views and not those of his employer. Ed.*]

“ What is the benefit of the WildList to corporate users… ?”

I write in response to Shane Coursen’s thought-provoking article starting on p.9 of the January 2000 issue. Like many, I was under the ‘mythconception’ that the WildList was, in broad terms, a list of viruses known to be ‘In the Wild’. However, I now understand that ‘In the Wild’ and ‘In the Field’ are two totally different concepts (no pun intended!). Having read the article a few times, however, I am prompted to ask ‘What is the benefit of the WildList to corporate users today?’

According to Mr Coursen, the WildList ‘is a list the anti-virus vendors agree as important for *all* anti-virus software programs to detect and repair’, and lists those viruses ‘verified to be found spreading throughout diverse user populations worldwide’. He notes further that ‘if a virus is detected and repaired without a problem, it is not spreading’, and adds that ‘a virus that was once considered to be problematic is no longer so because sufficient time has passed to allow most scanners to detect and repair it without problems. Thus, the virus is usually no longer reported.’ Then Mr Coursen states that ‘The WildList rarely lists those viruses that are old and ‘known’ – e.g. … those already easily detected and repaired by most scanners.’

Set against these clear statements of purpose, it would seem to me that the WildList has, sadly, become a self-serving creation of the anti-virus industry, designed to enable them to claim that their products detect 100% of ‘In the Wild’ viruses (which I doubt is what Joe Wells had in mind when he first started the WildList in 1993). It has become a questionable benchmark against which anti-virus software can be measured for its detection capability, rather than a list of viruses which are currently causing problems to users and organizations.

In light of what the WildList now purports to represent, how can the anti-virus industry explain why viruses such as AntiEXE.A (first reported on the WildList in September 1994), Form.A (first reported in July 1994, and previously reported simply as Form since at least July 1993) remain on the WildList today? Surely, every anti-virus product can detect and repair the Form virus?

I also believe that the WildList is in danger of losing credibility in respect of macro viruses, which I think we all agree are the single biggest cause of corporate headaches today. Through the *WildList Organization International’s* (*WLO*) insistence on seeing two working samples of a virus prior to the virus being added to the WildList, and through the use of generic detection of macro virus families by some anti-virus products, the expediency of the WildList is significantly threatened. Despite assurances that the text can be ‘cleaned’, leaving only the macro intact, many users are reluctant to provide macro virus samples for fear of loss of data confidentiality. The *WLO’s* insistence on two independent working samples could therefore unduly delay the appearance of a new macro virus on the WildList. The practice of some anti-virus companies, whereby all variants of a macro virus family are identified only by a generic name (W97M.Ethan.gen, W97M.Groov.gen, and W97M.Marker.gen etc) is counter-productive, since it gives no indication of which variant has been detected. Therefore, unless all samples are submitted to *WLO* Participants, and they correctly identify the variant in question, the WildList will be seriously out of line with the ‘real world’ in a very short period of time.

While development and maintenance of the WildList since 1993 has been a Herculean and worthwhile exercise, is there now a case to be made for the monthly production of an ‘In the Field’ virus report, which would keep those of us responsible for safeguarding our corporate systems up to date with what is actually a threat? Let the anti-virus industry retain the WildList, but a combination of the WildList, the *VB* Prevalence Table, and a greater sharing of information among the user community is really the best guide to what is ‘in the wild’.

NEWS

Relinquishing Command?

Russian AV company *Kaspersky Lab* claims it has terminated its distribution agreement with US-based resellers *Central Command Inc*. Managers at *Kaspersky Lab* were more than ready to talk to *Virus Bulletin* about the whys and wherefores of this situation, and keen to reassure exisiting US customers that this would in no way affect the service they had come to expect from *Kaspersky Lab's AVP*. Indeed, we were told that there are plans to announce 'a special program' for US users very shortly.

On the other hand, representatives from *Central Command* were less communicative, advising *VB* that while they considered it to be 'business as usual' stateside, the situation had 'gone legal' and was out of their hands ▮

Meltdown

W32/Melting, a newly-discovered Internet worm, has been reported in the wild in Eastern Europe. The Win32 PE EXE file, about 18 KB in length and written in Visual Basic, is transferred via the Internet in an email message with the infected file named MELTINGSCREEN.EXE attached. Win32/Melting sends messages, each containing a copy of the worm, with the subject header 'Fantastic Screensaver', to addresses it finds in the *MS Outlook* address book.

The worm changes all .EXE file extensions in the *Windows* directory to .BIN extensions. It then 'melts down' the screen as promised. Win32/Melting does have bugs and will often freeze up the machine when active. Most of the major AV companies have updated their products to detect this latest worm ▮

Go Figure!

A California-based, independent research firm – *Computer Economics Inc* – has released the disturbing results of its year-long internal survey of major corporations across the globe. It reports that the economic impact of virus attacks on business was in excess of $12 billion in 1999 alone.

According to research analyst Samir Bhavnani, the outlook is bleak, 'This form of economic terrorism is growing as viruses are no longer simply the minor annoyance that they were a few years ago.' Bhavnani's advice to businesses may sound familiar: 'Corporations cannot afford to play Russian roulette with professional virus writers.' ▮

Stand to Attention

We look forward to seeing you at the *Virus Bulletin* stand (number G226) at InfoSecurity 2000 at Olympia in London from 11–13 April ▮

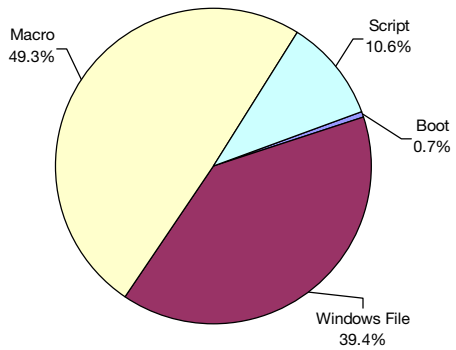
Prevalence Table – February 2000

Virus	Type	Incidents	Reports
Win32/Ska	File	268	21.90%
Win32/Pretty	File	140	11.44%
Marker	Macro	115	9.40%
Laroux	Macro	94	7.68%
Freelinks	Script	92	7.52%
Ethan	Macro	82	6.70%
Tristate	Macro	43	3.51%
Myna	Macro	40	3.27%
Class	Macro	38	3.10%
Win32/Fix	File	31	2.53%
Thus	Macro	28	2.29%
Melissa	Macro	27	2.21%
Kak	Script	26	2.12%
Pri	Macro	16	1.31%
ColdApe	Macro	14	1.14%
Fool	Script	12	0.98%
Win95/CIH	File	12	0.98%
Cobra	Macro	11	0.90%
Win32/NewApt	File	11	0.90%
Story	Macro	10	0.82%
VMPCK	Macro	9	0.74%
Cap	Macro	8	0.65%
Others [1]		97	7.92%
Total		1224	100%

[1] The Prevalence Table includes a total of 97 reports across 42 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

* In order to avoid a distortion of the figures, data for the 'self-reporting' W97M/ColdApe virus (totalling 943 reports in February) have been omitted from the table this month.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

The Scanner Situation

Some years back it was gospel that it was always a good idea to employ more than one scanner for better virus protection, the idea being that two scanners might cover up for each other's weakness.

However, what's the situation today? Have ideas used in present-day scanning engine design merged to such a degree that is doesn't improve security much to use scanners from more than one vendor? Isn't it a fact that differences in the most high-profile scanners now lie more in culture than in efficiency?

Jens Lynge
Danske Data
Denmark

Tackling Trojans

I have worked for many years as a Systems Administrator in Spain's largest university, witnessing how hundreds of our machines have been infected with Trojans of all kinds.

There is a strong lack of methodology with regards to studying if a computer is infected by a Trojan or not. With well-known Trojans like BackOrifice, NetBus etc, this is not a big problem because a competent Systems Administrator will know them well and perform regular scans on the network searching for familiar default Trojan ports. There is also specific software which detects and removes these 'Trojans for the masses', and even good anti-virus programs perform well on them.

The real problem is when we face new, unfamiliar Trojans. They are not documented at all and, unfortunately, they are spreading over the 'Net quickly and dangerously. There are hundreds of these little-known Trojans and no specific software to fight against them – no dedicated anti-Trojan programs, no anti-virus software (even using heuristics).

We have found one of these Trojans, called WinSATAN, and developed a general methodology to study untrusted software. We call it *MAUS* and we will probably present it for the first time at *ACSAC 2000*. We are currently working on a project to develop software based on our methodology.

If any *Virus Bulletin* readers consider this subject interesting, please contact me or the editorial team at the *VB* offices in the UK.

Julio César Hernãndez (jcesar@inf.uc3m.es)
Carlos III University
Spain

Reserving Judgement

I am writing in response to a letter from Paul Robinson, Editor of *Secure Computing*, in last month's issue. He asks an interesting question – how much better off are we as a result of the prosecuting and imminent sentencing of David Smith?

I agree with Paul that the potential for notoriety (for Smith) is great; that sort of notoriety certainly would not make us 'better off'. Also, the very real possibility this whole situation will turn into a positive thing for Smith (with other youths encouraged to follow suit) is one that we can't ignore. One need only consider another letter to *Virus Bulletin* (published in the same issue), wherein Jacky Cha documents the ways in which the media in his country portray the virus writer as a hero or genius, to observe the effect of this type of positive reinforcement. These are not new, nor isolated, phenomena.

However, while I appreciate Paul's sentiments, I think it is still too early to conclude (as he seems to have done) that this prosecution and sentencing of Smith will have little, if any, effect on future virus authors/distributors. After all, we have not yet had the opportunity to objectively measure what, if any, impact these recent law-enforcement and judiciary interactions have actually had.

Let's not throw out the idea of legal remedy, nor that of the social sanction which may be provided by a fair and just sentencing in this case, as having no correlation with viral impact. In this case, I think the opinion given by Paul Robinson is likely to be 100% right on – but let's wait until we've had the opportunity to observe and measure the impact of these legal realities, and to consider anecdotal evidence.

Sarah Gordon
IBM Thomas J Watson Research Centre
USA

Over a Million Served – A Bunch of Kak

Functionality, automation, scalability – all buzz-words which equate to productivity. In some aspects, they are heralded components of the way we work, operate and function. With each enhancement to a product, we are quick to grab and download the updates to enrich our lives and give us that added feature that makes our experience better in some small way. But are we too quick to implement that which may be harmful or benevolent to our own function or impede our progress?

Take, for instance, *Windows Scripting Host* (*WSH*), a fine tool for someone who may use automation or some other facet of development where the need for a low overhead

and yet functional task or process is required. But what about the rest of the computer world, do we really need this special feature add-on?

If you are like me, you often run out of space on your hard drive and wonder how you got into such a situation. Consider the bells and whistles of all the applications which may reside on your system and it doesn't take long to find out that a suite of products can consume over 200 MB of your drive easily. With this in mind, I'm often looking for things to trim out of my system, things that I don't necessarily need or use. What's this 'Windows Scripting Host'? Hmm, do I really need this? No thanks, I'm doing just fine without it. Powerful stuff yes, but sometimes too big for its own pants.

To see an example of this, one need not look far – visit the nearest newsgroup near you and get a glimpse of the number of posts made which contain VBS/Kakworm, an exploit of *WSH*. Not only do you not have to open the message to be a victim of this nuisance, if you are running the preview pane and have *WSH* installed, this little menace will simply install itself without batting an eye. On the next *Windows* restart, you are now the proud owner of an Internet worm that travels by (hidden) signature to HTML email messages – congratulations.

Perhaps I am being too harsh with regard to the full aspects of *Windows Scripting Host* and its pros and cons, and perhaps not. A solution to the actual exploit has been available for quite some time, yet how is it that the solutions are less visible than the upgrade to a product? It is my opinion that security patches and corrections to a product should be *more* visible than the add-on enhancement page. That's my story and I'm sticking to it.

Patrick Nolan
NAI
USA

Kak Revisited

Several points in Vanja Svajcer's 'Kak-astrophic' virus analysis last month (p.7) caught my eye. The vagaries of publication deadlines means the 'reputed to be in the wild' claim had been overtaken by events by the time *VB's* March issue rolled off the presses. As Kak is now quite widespread, I think some of these points should be clarified.

Perhaps the first is that the virus is written not in VBS but in JavaScript. Also, apart from the English language *Windows* Startup folder, the original version of Kak tries to drop KAK.HTA in C:\Windows\MENUDÉ~1\PROGRA~1\DÉMARR~1. This should match the 'startup' folder of French langauge versions of *Windows*, C:\Windows\Menu Démarrer\Programmes\Démarrage, and is a hint to Kak's likely country of origin.

The payload trigger condition test is for the hour being greater than 17, so the payload can only trigger on or after 6pm on the first of any month, not 'after 5pm' as stated.

This analysis error is quite widely repeated on several vendor Web sites. I could quibble about the phrase 'the ActiveX embedded code launches itself', because technically the HTML and JS parsers do the 'launching' – the point is that the code itself is not 'active' as this wording implies, but depends on the email client who is 'reading' the message.

I realize Svajcer did not have the luxury of space to comment on this issue, but Kak (and BubbleBoy) would be sterile, despite the Scriptlet.TypeLib security flaw, if *Microsoft's* HTML parsers were not so keen to find and interpret HTML code. Kak's HTA files have 1,018 bytes of binary 'junk' (from an HTML parser's point of view) preceding the Kak HTML code – a side-effect of using the Scriptlet.TypeLib control itself. If the parser choked on such 'junk' prior to a valid HTML header – not an unreasonable thing for an HTML parser to do – and thus failed to process the rest of the file, Kak would have been stillborn.

Another commonly mis-described effect seen in Kak's code is in the additions it makes to AUTOEXEC.BAT. The first line that Kak adds to that file does not call or execute KAK.HTA from the Startup directory. How could it? Doing so depends on MSHTA.EXE and the associated *Internet Explorer* HTML and scripting interpreter engines – all 32-bit sub-systems that cannot run until *Windows* is running. True, the first line of batch code is somewhat odd

```
@echo off>C:\<startup-path>\kak.hta
```

hanging a standard DOS prompt. However, in a batch file it causes the target of the redirection to be overwritten with a zero-length file. My assumption is that this is an attempt to make recovery of the contents of KAK.HTA more difficult.

Although only email is mentioned in the analysis, Kak also spreads via *OE5* news postings if the infected user has enabled HTML messages for news. Finally, the most important point about cleaning up a Kak infection is that you *must* close the Scriptlet.TypeLib security hole before doing so. If this is not done, you are easily re-infected from reading messages you have saved – even from your own messages in *OE's* Sent Items folder. If your virus scanner does not clean messages inside *OE* folders you also have to do something to prevent forwarding infected messages or replying to infected messages and sending Kak on with the reply. To achieve that you must disable the use of HTML format for email and news (good taste dictates that anyway!) *and* disable *OE's* 'Reply to messages in the format in which they were sent' option as well.

If you have not been infected yet and do use *IE 4.0* through *IE 5.0* inclusive, please check whether you need the Scriptlet.TypeLib security patch. A description of the problem and patch, and a link to download it, is available from http://www.microsoft.com/technet/security/bulletin/ms99-032.asp.

Nick FitzGerald
Computer Virus Consulting Ltd
New Zealand

VIRUS ANALYSES

Poetry in Motion

Péter Ször
Symantec, USA

This month Péter Ször takes a look at three *Windows* viruses, the first two of which are recently released mass-mailing worms. The third analysis is concerned with a potentially dangerous variant of WinNT/Infis.4608, now updated for *Windows 2000*.

1. Win95/Haiku

The number of mass-mailing email worms is rising very rapidly. At least one third of the 32-bit *Windows* virus variants written this year alone can be classified as mass-mailing worms. Win95/Haiku.16384 was created by a longtime 'retired' founder member of the 29A group who calls himself 'Mr Sandman'. His creations include the infamous Esperanto *Windows* virus released some years ago (see *VB*, December 1997, p.3). Since then he has been very quiet and nowadays he is no longer part of 29A. He is very interested in languages, claiming to speak several, and works as a professional translator.

Win95/Haiku is his first mass-mailing worm. The really interesting thing is not its mailing routine, but its functionality. Haiku is capable of creating small poems, so-called 'haiku'. The worm propagates itself by sending emails with an attachment called HAIKU.EXE.

The Story of Haiku

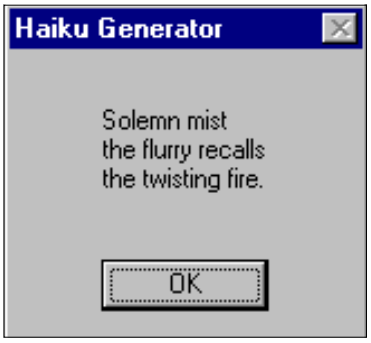
The subject of the email is 'Fw: Compose your own haikus!', so it looks like a forwarded message. The body contains a small introduction to the haiku form:

```
:))
—— Original Message ——
>"Old pond...
> a frog leaps in
> water's sound."
>- Matsuo Basho.
>
>DO YOU WANT TO COMPOSE YOUR OWN HAIKUS?
```

A haiku is a small, oriental-metric poem that first appeared in the sixteenth century. It is popular mainly in Japan and the USA. Apparently, its form transcends the limitations imposed by language structure and the scientific philosophy which treats nature and the human being as machines.

The poem usually consists of three lines and 17 syllables, distributed in five, seven and five format. It must register or indicate a movement, sensation, impression or drama of a specific fact of nature, rather like a photograph. More than inspiration, what you need in order to compose a real haiku is meditation, effort and perception.

Initialization



When the attached HAIKU.EXE is executed it installs itself on the system by copying itself to the *Windows* directory as HAIKUG.EXE (Haiku Generator). Then it modifies the RUN field in WIN.INI under the section in order to execute HAIKUG.EXE at each system start from then on. Then the worm displays a haiku message box. Win95/Haiku randomly selects words from a word table. Some words may have different endings using 's' and 'es', respectively. The first few words in the table are: 'bridge light sea fish butterfly foghorn day moon evening spring sunset boat petal blossom stone mist passage darkness dolphin ant shadow star frost… '.

Mail Propagation

The worm searches on the local hard disk for .DOC, .EML, .HTM, .RTF and .TXT files, opens them and checks if they contain any email addresses. Thus, Haiku is more like a spam generator – it does not determine emails on the fly.

Then the worm connects to IP address 194.106.68.104 and uses port 25 (mail). This server appears to be opened for anonymous usage. Anybody can log in and instruct the mail server to send emails. This is a very common security problem that is used by spam authors often. This will, of course, limit the worm's lifetime to the period when the mail server is open for anybody. The worm's mail engine uses the SMTP protocol to send emails.

First, it introduces itself to the server 'HELO haiku.com'. It then sends the email: 'MAIL FROM: haiku@haiku.com'. After sending the email the virus leaves the server with the 'QUIT' message. Haiku uses MIME encoding for the attachment. During propagation the worm may display a message box with the following encrypted text:

```
[ I-Worm.Haiku, by Mister Sandman ]
 The smallest box may hold
 The biggest treasure?
```

The Win95/Haiku worm also connects to 206.132.185.167 (http://www.xoom.com) and uses the GET command to download a *Windows* WAV file (…/HAIKU_WAV/HAIKU.WAV). It creates C:\HAIKU.WAV and plays the WAV. Finally, it deletes the WAV file. The header of the WAV file contains the copyright message: (c) Mister Sandman, 2-2000. The worm's propagation is speeded up because Win95/Haiku's code does not have to carry the 56 KB WAV file.

2. Win95/Fix2001

This is a relatively 'old' worm which was created during the autumn of 1999. At that time it was not particularly widespread in the wild. It took a few months for Fix2001 to get any real attention. Several companies in the US were hit by it in December and in January the number of submissions to *SARC* showed that Fix2001 is really out there, all around the globe.

Win95/Fix2001 is an Internet chain-letter worm that will secretly steal dial-up information (including the password) and send it out via email to the hacker. This capability makes it really dangerous, since a hacker can use the information to hack into previously infected networks unless the passwords are changed. For a few weeks the worm's mechanism was unknown to all major anti-virus vendors. This is because it uses a very sophisticated method to access the *Windows 9x* dial-up passwords. It gets this information from the active RASAPI32.DLL in memory.

The worm arrives via email as a MIME-encoded attachment named Fix2001.EXE. The subject of the email is 'Internet problem year 2000'. It is sent by a person named 'Administrator'. The body of the message contains a message written in Spanish and English encouraging users to use the email attachment to check for Y2K compatibility. Unfortunately, several corporate users believed it.

Initialization

When executed, the worm installs itself on the local PC's *Windows* system directory with the name Fix2001.EXE. It modifies the Registry's …\Currentversion\Run field to execute itself during subsequent reboots. When executed for the first time, it will display the following message:

```
Y2K Ready!!
Your Internet Connection is already Y2K, you
don't need to upgrade it.
```

The worm checks if a window procedure with the name 'AMORE_TE_AMO' exists. An already active worm creates this window procedure in order to send itself to other locations in the background. This way, there will be only one active copy of the worm in memory. Instead of modifying system DLL files on the hard disk, the worm hooks APIs to itself in memory by patching the process address spaces. Thus it will gain execution each time any Internet activity happens on the local machine. The technique and its implementation are unique to Fix2001.

When RNAAPP.EXE (Dial-up Network Application) is not running the worm executes it with the '-l' parameter. This will load RNAAPP.EXE silently. RNAAPP.EXE has import functions from RASAPI32.DLL and this is in the interest of the worm. Fix2001 patches a hook routine to RASAPI32.DLL's DialEngineRequest() API later on when RNAAPP.EXE is loaded. It puts a jump that points to its hook routine at the entry point of the DialEngineRequest() API, and patches its short code right after the import address table of RASAPI32.DLL. A string should appear right next to the empty area. Then the worm checks if a long enough area filled with 0 bytes is available and only patches the process if this is the case.

Fix2001 also hooks the 'send' and 'connect' APIs of WSOCK32.DLL loaded by Internet applications such as *Internet Explorer* or *Outlook Express*. This is a very similar technique to the one used by Win32/SKA.A, with the important difference that this patch is done in memory and not in the file. This provides the worm with the same potential to spread as SKA – a proven technique.

Once RNAAPP.EXE is patched, the worm hides it from the task list by registering it as a service process. The worm itself is registered as a service process too and therefore it does not appear on the task list. Since many utilities that list processes do not display service processes (that can be accessed only by specifying an additional bit for the process query function) it is not particularly easy to notice that the Fix2001 worm is loaded in memory.

The hook routine on the 'send' API looks for the 'RCPT' field of the mail header during postings. The worm sends its message with the Fix2001.EXE attachment to the very same place right after the original message. This is much the same idea as that used by several known email worms. The received email headers will always contain a header reading: 'X-Mailer: PUPI-MAIL v.0.1'.

Posting Dial-up Passwords

Via its hook function, Fix2001 is capable of searching for user information in the address space of RASAPI32.DLL. The function searches for a 'T' or 'P' character at specific locations – the locations of the user information data. This routine sets a flag when successful and only sends the information once to one of the hacker's three email addresses. Used email addresses are encrypted in the code of the worm. The phone line text message might start with 'T' or 'P'. (The first line is the machine name, the next is the dial-up number, then the user name comes and the last line is the password.)

Payload

The payload is activated after the worm has already posted itself to another location and an active connection exists. Then, a routine will perform a checksum on the last detected email address. If a particular email address encounters a checksum match, the worm will delete the C:\COMMAND.COM file and create another 16-bit COM program, named COMMAND.COM, that is 137 bytes long.

The Trojan will be executed next time the computer is booted. When the trojanized COMMAND.COM is executed, it will destroy the hard disk data (it overwrites it using I/O port commands) whenever the hard disk is an IDE drive. This can be a targeted attack against specific people, but the checksum can all too easily match someone else's email address by accident.

3. Win2K/Infis.4608

[*Readers are advised to refer to p.8 of November 1999's issue when reading this analysis. Ed.*]

A week after *Windows 2000* shipped, the WinNT/Infis.4608 virus was updated to support *Windows 2000*. Win2K/Infis, a 'memory resident', parasitic *Windows 2000* Kernel-mode driver virus, only operates under *Windows 2000* and is already likely to fail under the first service pack. It does not have a payload.

When the INF.SYS driver takes control the virus allocates a memory from the non-paged pool, reads its complete copy from the INF.SYS file for future use in its infection routine, and hooks INT 2Eh by patching the Interrupt Descriptor Table (IDT). This is all possible because drivers have the most powerful rights on a *Windows 2000* machine

INT 2Eh is the main *Windows 2000* service interrupt (just like in *NT*) and it is completely undocumented. A Win32 application normally calls an API from the Win32 subsystem. The subsystem translates the documented API calls to undocumented once exported from NTDLL.DLL. The NTDLL.DLL is the native *Windows 2000* API. It has hundreds of undocumented APIs. NTDLL.DLL is running in User mode, but it switches to Kernel-mode by using the INT 2Eh service interrupt with a function ID in the EAX register (on *Intel* platforms). Each function ID is created by a macro when *Microsoft* compiles *Windows 2000*. Therefore, the ID can be different between new releases of *W2K*.

Since Infis uses hard-coded IDs it will not be compatible with all *Windows 2000* releases. The most important modification in the virus is the new ID number usage. The parameters of the API calls are passed on stack. This way the appropriate *Windows 2000* kernel API will be called.

The INT 2Eh hook of the virus intercepts the file opening function only, checks the file name and extension, then opens the file, checks the format and runs the infection routine. (Infis only uses INT2Eh functions, even when an infected User mode application is executed and the virus User mode entry point is called. Thus, it completely bypasses *NT's* Win32 subsystem.)

Checking the loaded driver list can be tricky because Windows *2000* places the driver list under the Computer Management. First, you need to turn on the 'Display Administrative Tools' option for the taskbar. Then, click on the 'Computer Management' and select 'Device Manager'. The View has to be changed to 'Show hidden devices'. The 'inf' driver should appear on the list. With a right-click on the driver name you can disable the driver. The 'Properties/ Driver' tag also allows the driver to be stopped (this is because Win2K/Infis has a driver unload routine).

While Win2K/Infis still infects some files incorrectly, it is more stable than its predecessor. Unfortunately, such new driver viruses can use the CIH damage routine under *Windows 2000* since drivers can execute port commands.

FEATURE

A Nightmare on Researcher Street

Andy Nikishin & Mike Pavluschick
Kaspersky Lab, Russia

As is often written in *Virus Bulletin*, it is always a little daunting to predict the future – what if predictions come true? Some time ago we discussed polymorphism in macro viruses and in the last part of that article we talked about the future of polymorphic macro viruses (see *VB*, June 1999, p.14). Back then we said that it would be possible to create real, strong polymorphic viruses using VBA5. It looks like our predictions came true.

At the end of December 1999, a Russian virus-writing group released its magazine – *DVL*. The issue contains write-ups on different kinds of viruses and various other articles. One of them piqued our interest – it was a little essay called 'Polymorphism in Word 97'. To be honest, we have read a lot of this kind of thing and we must say that most of them are pretty dull, but this one really impressed us. The author of this particular piece approached polymorphism in a different way.

Most recommendations for polymorphism suggest adding either comments in random places or unusable variables in code to confuse heuristic analysers and complicate virus analysis. This method has one main disadvantage – in a few 'virus generations' the virus will grow, so the macro stops working. A good example of such a virus is W97M/Groov. The size of its original code is about 6 KB, but the third generation is about 10 KB and so on. In the *DVL* article a virus writer suggested using good old file virus technologies – encryption and a polymorphic decryptor containing a garbage instruction which looks like a useful one:

```
RKFe5 = 1 ' Decryptor's part
Do While RKFe5 <= Len(Y7) ' Decryptor's part
Do Until o0Bukn4 > 30
o0Bukn4 = o0Bukn4 + 2
Loop
LjPvXw8 = (UsRgNN5 + BgaB0) Mod 255 '
Decryptor's part
jEcmjs1AXhT5 = 78
DpOjLoB1QaZzu8 = 151
LNTL1oGAFc7 = 0
IsMb2io0 = 175
Do While LNTL1oGAFc7 < 52
LNTL1oGAFc7 = LNTL1oGAFc7 + 5
Loop
cxJJIVJ3 = Asc(Mid$(Y7, RKFe5, 1)) Xor
LjPvXw8 ' Decryptor's part
cZS7 = cZS7 + Chr$(cxJJIVJ3) ' Decryptor's
part
kqNCQI5XUE6 = 5 YOCk3FY6 = qekoP8 + qRdlho3
For evsGCm1iMOuB6 = 5 To 30 Step 3
nGyydTy0howSOOO7 = 2
```

```
Do
nGyydTy0howSOOO7 = nGyydTy0howSOOO7 + 4
Loop Until nGyydTy0howSOOO7 > 62
Next
Do
kqNCQI5XUE6 = kqNCQI5XUE6 + 8
Loop Until kqNCQI5XUE6 > 89
RKFe5 = RKFe5 + 1 ‘ Decryptor’s part
Loop ‘ Decryptor’s part
```

So, only a few strings of actual code are used in the decryptor and the others are garbage. The garbage code is generated randomly, but it looks very realistic. This trick not only complicates the analysis of a virus, it really complicates detection.

Included in the magazine were two viruses which illustrated these principles – namely, W97M/PolyMac and W97M/PermutationPolyMac. Both of them use the same polymorphic engine – MCPRACE (Macro Crypted Polymorphic Realistic Antiheuristic Code Engine). Let us review these viruses in more detail.

The Polymorphic Engine

The engine is actually a standalone procedure and can be built into any virus quite simply. As a result of its work the engine generates the string that contains garbage code. It uses three main constructions as this code:

1. Operations with variables – assigning, multiplying, adding or subtracting variables or constants.
2. Five kinds of loop –

   do while: loop

   do until: loop

   do: loop until

   do: loop while

   for: next

   Inside the loops there are operations with the loop counter.
3. Condition statements with variables and constants. All constants and variable names are randomly generated.

This engine uses a recursive algorithm to generate garbage code. This means that inside any garbage construction there may be another one up to the level of recursion. In this implementation the recursion level has been set to five and there are only three main constructions used, but it is not hard to increase that number.

W97M/PolyMac

W97M/PolyMac infects *Word 97* documents and the normal template. It contains one macro – ‘Document_Open()’ – in the ‘ThisDocument’ module. The virus infects the global macro area on opening an infected document. Other documents are infected when they are closed. PolyMac is encrypted and contains a lot of garbage code that is not used either in decryption or in infection routines.

After the virus gets control it decrypts its code, creates a new document and places the decrypted code into it. It saves this infected document with a random name to the normal template, causing code recompilation. Then, it turns off *Word’s* macro content warning facility, opens the file saved previously, and closes it again. The infection routine needs this in order to take control and infect the normal template and all currently open documents. Finally, the virus deletes the temporary file that has been saved.

PolyMac uses the ‘ConfirmConversions’ global property to prevent repeated infection of the temporary file. During infection the virus encrypts its body using the exclusive OR logical operation (XOR) with randomly generated keys. It saves the result as a string, splits it into sizeable strings, adds garbage code and finally adds the decryptor (also containing garbage instructions). The virus inserts the resulting code (with the encrypted body and the decryptor) into target victim documents and NORMAL.DOT, if they still contain no macros. PolyMac has no payload.

W97M/PermutationPolyMac

This virus uses exactly the same infection routine as W97M/PolyMac but it has a much more comprehensive polymorphic engine. While the polymorphic code is being generated the engine makes additional commands to manipulate flow control. It inserts a randomly generated label at the beginning of every split line and a GOTO command that passes control to the next split line.

This method was previously implemented in the W97M/Walker.B virus, but PermutationPolyMac uses one more feature to ‘permutate’ its body. During the processing of each split line the virus, depending on a randomly generated number, leaves that line as it is or creates a new subroutine with a random name and moves this line into it. The virus places a CALL statement to the subroutine at the original location of the moved line. Finally, it changes the order of all split lines, but the code preserves its functionality by using GOTO statements. As a result, the code becomes very mixed, with lots of subroutines that really complicate analysis and detection of this virus.

Conclusion

These viruses will not become widespread for several reasons. The polymorphic engine uses many loop statements (most of them recursive) which slow them down significantly. The time required to open documents is increased tens if not hundreds of times. Also, the VBA engine restricts the size of code that can be placed in one procedure. The polymorphic engine can generate code that exceeds that limitation and such a procedure will cause an internal VBA error. Finally, both viruses contain bugs that cause a malfunction. PolyMac and PermutationPolyMac are the very first macro viruses to use such comprehensive polymorphic algorithms. These two may have no future, but their ideas and code may be modified and debugged. Then we really will have a nightmare on our hands.

OPINION

Childhood's End – Demythologising Anti-Virus

David Harley
Imperial Cancer Research Fund, UK

It is not what people know, it's what they know ain't so. The world is full of self-perceived virus experts, and misinformation wants to be free – especially virus-related misinformation.

The Two Faces of Dave

You don't have to be crazy to work in security, but it helps. Paranoia may be a survival characteristic, but I sometimes wonder about my multiple personalities. Time and time again I catch myself talking about 'security and anti-virus' as if they were separate issues. I don't believe that, but it sometimes seems the security industry does (both the industry in general, and the anti-virus subset).

Why does the anti-virus sector have such a bad reputation? Anti-virus vendors are seen by other sectors of the industry and an increasing proportion of their customers as 'poachers turned gamekeepers' (or worse), actively contributing to the problem they claim to solve. AV personnel, in their turn, constantly present themselves, self-protectively, as a special case: self-perceived, super-ethical white magicians possessing special knowledge which is unsafe to share with the 'Great Unwashed'. Where does this extraordinary divergence in perceptions originate, and how far do the stereotypes reflect reality?

Playing the Numbers

The anti-virus industry is frequently perceived as having a vested interest in the creation of new viruses. Indeed, even as an outsider to the industry, when I conduct virus-related training sessions and talks away from the regular conference circuit, I'm still often asked 'Have you ever written a virus yourself?'

The industry doesn't do itself many favours here. Firms continue to play 'follow the leader', however reluctantly, through the numbers game, counting multiple progeny of the same construction kit as single viruses where a generic driver will catch them all. The dubious use by marketing departments of the term 'in the wild' can also backfire. People wonder how there can be so many if they aren't being mass-produced, and ask who benefits most from virus mass-production.

This argument can be countered to some extent by pointing to statistics which demonstrate the continuing presence in the WildList of viruses which might be expected to have burnt out years ago. However, that may indicate only that there are plenty of totally unprotected machines out there. It doesn't address the fact that anti-virus companies continue to cling to an arguably self-defeating, known virus-specific approach to detection. After all, if the medical industry operated in the same way as the anti-virus industry, everyone would be immunised against all viral diseases as and when they were discovered, with a likely negative long-term impact on species viability as natural immune response systems atrophied.

Still, detection of known viruses works, up to a point – perhaps better than it does with biological viruses. It's conceptually easier to continue on that course, and it's much better at handling known virus incidents transparently than generic semi-solutions are. (I'm talking about reactive, not proactive solutions here, of course.)

Furthermore, most customers are resigned to being locked into a profitable upgrade/update cycle. However, it all adds weight to the stories of virus production lines and bounties paid to virus writers, despite the lengths companies have gone to in the past to disassociate themselves from suspicions of employing current or former virus writers.

DDoS, Done and Dusted

What about the accusations that AV researchers are more interested in debating absolute definitions of 'malware' and the precise number of polymorphs that can dance on a pinhead than in implementing a holistic enterprise security solution? It has to be admitted that vendors usually have a fairly narrow focus.

At the recent *EICAR* meeting, a panel of experts from the networking industry discussed current and future strategies for reducing the impact of DDoS (Distributed Denial of Service) attacks (not eliminating them, you'll note). The next day, I sat at a product launch where it was indicated that DDoS tools were a 'kind of virus' that the anti-virus industry had already got the measure of. Are both these groups looking at the same problem?

It's all too common in IT, when faced with a problem we can't solve, to address it as if it were a different problem, one that we might be able to solve. It's not feasible for the average anti-virus company to solve internetworking problems, though some have staked a claim of sorts in that marketplace by cooperating with enterprise firewall manufacturers, or buying in firewall, IDS (Intrusion Detection System) or personal firewall technology.

However, DDoS attacks can't be eliminated by single-layer solutions, even by the backbone ISPs and other network specialists who have the most experience with this type of attack. Signature scanning detects some of these attacks as

they begin, but detection isn't elimination. The impact of a flooding attack can be reduced by rate limiting – capping the bandwidth available to particular types of packet associated with DoS (Denial of Service, distributed or otherwise) attacks. However, that approach barely qualifies as detection, let alone elimination, since it doesn't necessarily discriminate between 'rogue' packets and 'legitimate' packets (not necessarily a feasible distinction).

Some anti-virus companies seem to be arguing that since they can detect known DDoS slave software in the same way that they can detect RATs (Remote Access Tools), worms etc, they can deal with the problem by scanning for such tools at the gateway and on vulnerable servers and workstations. This can help, but it addresses rather a small part of the problem.

Besides, if known virus detection is so effective how is it that Cap, Concept and even Form still feature strongly in the WildList and the *VB* Prevalence Table? Can we, in any case, safely assume that anti-virus vendors can keep up with DDoS identification as promptly and effectively as they do with virus identification?

K001 Hand Luke and the Oscar Wilde List

What we have here is not just a failure in communication, but two cultures separated by a common terminology. When anti-virus experts talk about signatures, they're usually talking about a largely outmoded virus detection technology, simple pattern detection in a system area or file stream. When intrusion detection experts use the same term, they usually mean a characteristic pattern in a packet stream – same terminology: different media, different perceived functionality.

When is a virus not a virus? When it's a worm, or not, depending on whether you're listening to a virus specialist or a network security specialist. Maybe it doesn't matter. Certainly most practitioners have very little interest in the distinguishing characteristics of viruses, worms, rabbits, bacteria, octopii and gerbils. Outside the conference circuit, maybe the industry doesn't either. How else (apart from sloppy programming and an inability to think ergonomically about interface design) do we explain the countless warnings from AV software that a file is infected with the xxx/Trojan virus, or the Joke/xyz virus?

Great Hoaxes from Little Acorns

Take hoaxes. In this industry, 1997 seems to have been the year of the hoax, and several papers at VB'97 addressed the issue. Then the industry, having done its duty, went on to more interesting issues, like upconversion and pseudo-biological auto-immune response systems.

Yet hoaxes continue to proliferate, and computer users continue to react inappropriately, as has been pointed out in *VB's* Comment column for two months in succession. It obviously isn't enough for the industry to point people to www.kumite.com/myths or murmur politely 'tighten your policies', or for practitioners to rely on descriptions of known hoaxes and multiple exclamation marks. Surely I'm not the only person to have observed the number of hoaxes which seem to have their origins in anti-virus software false alarms? (Not only false positives, but also joke programs categorised as viruses or Trojans.)

Gods and Ants

Is AV really a special case or the last refuge of the 'gods and ants' mindset? Certainly, there's room for some home improvement up on Olympus. When I talk to other independents, some complaints are raised time and time again:

- Poor and inconsistent handling of non-viral malware, semi-malware, and non-malware. On the sites I administer, CokeGift has been infinitely more troublesome than CIH: not because it's intrinsically dangerous, but because anti-virus software identifies it as a virus.
- Inconsistent terminology and virus nomenclature is another favourite. What could be more enjoyable than spending late Friday afternoon browsing vendor Web sites and playing with VGrep in the hope of finding out what some distant customer has really found on their PC?
- Poor on-line help, documentation and help-lines. Inaccurate on-line and Web-based specific virus information. Problems with misidentification and disinfection, and an inability to admit to the existence of these problems (and others such as incompatibilities with specific software or OS versions) until forced to by third party publicity, or until a fix is available.
- Inconsistency between marketing claims and real-world malware management, and the announcement of vapourware as if it were available now, as fully mature technology.
- Virus alerts/advisories and quasi-independent training used as a marketing tool. It takes more than product training to make a virus expert. For product-independent training go to an organisation outside the industry, possibly one so far removed that it includes virus code in its Intrusion Detection FAQ.
- Inability to share critical information outside the industry. Giving the customer what they think they want or the vendor tells them they want, instead of giving them what they really need.

Some of these criticisms are truer of some vendors than others. However, mistrust of the industry is so deep-rooted that it's unlikely that individual vendors can overcome these problems piecemeal, or without the support of independent bodies and experts outside the vendor community. This is likely to be a painful evolutionary process, though, and getting it to work is an article in itself.

A DAY IN THE LIFE

Get the Message?

Alex Shipp
MessageLabs, UK

I work for a company called *MessageLabs* (which, until very recently, was known as *StarLabs*). My main responsibility is running our managed service, scanning email for viruses. We have a number of ISPs reselling our service, including UUNet, and over 100,000 users across the UK.

Although we are based in the UK, we scan email worldwide using our scanning towers. They are hosted at locations around the world and each one is capable of scanning 3,000,000 emails a day. Not surprisingly, all this takes some looking after which is the job of our operations team based in Cirencester in the UK. In the following article I will be taking a look at a typical day in the life of our operations team.

Aside from a multitude of less vital jobs, the operations team has three major tasks assigned to it:

- keep the virus scanners up to date and running smoothly
- keep the mail flowing smoothly
- answer support calls that first-line support cannot

Keeping the Scanners Up to Date

The first task of the day is to check email for any breaking news, typically looking for information on new viruses in the wild. If a new virus breaks then we have several methods available to stop it. Firstly, if a public signature is available, then our automatic web patrol robots will have already detected and applied it.

Secondly, we can contact our anti-virus vendors to see if they have a signature available. We use three virus scanners, chosen after extensive trials for their overlapping coverage. They all come from major international developers familiar to the readers of *Virus Bulletin*. We have a very good relationship with all of our suppliers.

Every virus that is caught is sent to our 'virus pen' where it is passed through all of the scanners. If a scanner misses it we investigate and send a sample to the appropriate anti-virus vendor. Lastly, we can update our own rule-based scanner – *Skeptic* – if we know some characteristics of the virus, or if we have a sample ourselves.

We have been quite successful at catching new viruses before signatures are publicly available, catching over 30 copies of ExploreZip, for instance. As I was writing this article (January 2000) our automatic detection routines detected a suspicious email script which turned out to be the VBS/Kakworm. This was not picked up by our scanners, so our team prepared a sample for distribution to our anti-virus vendors while also researching the virus and updating *Skeptic*.

Speed of response is crucial to our organisation. We were able to update, test and roll out a new version of *Skeptic* within 20 minutes. Eventually, we caught 23 copies that day and 30 the following day.

Mail Flow

Once the scanner update is out of the way, the next job is to check that mail is flowing correctly. Our towers are mainly self-regulating. They check their own status and report back to our operations centre where we look at things like mail queue length, response times, delivery failures and so on.

All these details are monitored centrally so we keep an eye on what is happening around the world. On average, 1 MB of mail attachments takes just under one second to process and scan, although complicated formats such as ZIP and *PowerPoint* files can take longer.

Very rarely, a file will trip up one of the virus scanners, and either crash it, or send it into an endless loop from which it never recovers. This will be picked up by our error handlers or watchdog timers, and in such cases we work together with the appropriate anti-virus vendor to help improve their product's performance.

The system has been designed so that such rogue emails do not hold up normal mail delivery. However, we use a 'belts and braces' approach and do not take anything for granted, so we also have a series of housekeeping timers and monitors that check that everything is working properly. If anything unusual is detected, the towers attempt to elicit help with ever increasing urgency until the problem is acknowledged. Initially, the first contact is made by email and pager. Various routes are tried more and more frequently until cancelled by a support engineer.

We also keep an eye on mail volumes to check for unusual patterns. Today, for example, we noticed an increase in mail volumes from one of our customers. After investigation it turned out that their mail server was an open relay and was being used by hackers to send spam. We subsequently contacted the customer, and advised them on how to close up the security hole.

Support Calls

The bulk of our helpdesk calls tends to break down into five categories – 'Is this email I received a hoax?', 'Can you tell me more about the virus you caught?', 'I just heard about a new virus! Are we protected against it?', 'My system didn't detect the virus you are warning me about!' and 'I think a

virus got through your system'. Our first level support engineers deal with most of the first three categories, and the others get passed through for further investigation. Invariably, the 'viruses' that got through turn out to be joke programs or inactive macro viruses that have been imperfectly cleaned at one time or another.

We also have regular contact with some of the specialist journalists who deal with computer viruses. Our company has a unique view on the world of viruses. Every virus we catch is one that the sender was (presumably) unaware that they were infected with.

Most other AV companies will only get samples if the user is suspicious, and often they will not be contacted if their product detects and disinfects a virus successfully. We have our finger right on the pulse of the world of viruses, at least those that are prevalent in email.

Using our extensive database we are able to answer many frequently asked questions. These include – 'Which industry sectors are sent the most viruses?', 'What are the current up and coming viruses?', 'Which domains do viruses come from?', 'Which day are most viruses sent on?' (Thursday, for some reason).

We also have a large volume of email statistics at our finger-tips. We have access to information such as distribution of email sizes, most common attachment types, typical emails sent per user per day and so on. These can then be used for capacity planning purposes.

We take the occasional amusing support call. Recently, we noticed that one customer was sending large numbers of the same virus, so we gave him a call to see if he needed any help. Initially all was fine, and he seemed to be welcoming advice on how to get rid of the virus from his systems. After about five minutes, he suddenly broke down and confessed that he was trying to get his 'favourite' virus into our top ten statistics Web page!

The Best of the Rest

We are now getting more involved in actively seeking out new viruses. We use our own heuristics scanner to search *Office* macros for viruses and in addition to that we are regularly sent suspicious samples by our network of contacts. *MessageLabs* is planning to expand this side of its operation to become much more aggressive in the future.

Today, for example, a couple of interesting office macros were thrown up by the scanner, but on further investigation both turned out to be false alarms. We duly added them to the list of false positives, so we will not be troubled by them again. We also detected VBS/Kakworm, which I mentioned earlier.

One of our anti-virus vendors has released a new version of their scanner. Since our business is time-critical, we lose no time QA-ing the new product on our test tower. We pass through our collection of viruses, false positives and troublesome files (mostly ZIPs and PPT files). Once this is performed successfully, the scanner is rolled out to our production system as soon as possible.

Since we use three scanners we can actually cope quite happily with buggy products, (as long as they are not too buggy!), and have helped to test and report on beta products in the past. A crash by the anti-virus product does not affect the rest of the system; all that occurs is that the email in question will be scanned by two products instead of the usual three.

We also detect and report on the crash to the particular anti-virus vendor. On this particular day, however, no problems occur, and the product is soon rolled out and functions happily on all our production towers.

Towards the end of each month we start preparing samples for the WildList. As everyone in this business knows, replicating viruses is tricky, requiring many different operating system versions and also many different application versions. We have automated as many of the processes as possible, but it still takes over a day to get all the WildList samples ready.

Our original database design only allowed us to report at the virus strain level – we could say 'We caught 450 copies of Ethan this month'. Over the last few months we have been reworking our reporting engine, so from January 2000 onwards we have been reporting individual variants – 'We caught 400 copies of Ethan.A and 50 copies of Ethan.AT'. Hopefully, Shane Coursen will not be too upset by the fact that this means we will be sending him twice as many samples as usual!

And so ends a typical day – except there are no typical days here! I can honestly say my job at *MessageLabs* is the most fun job I have ever had. No two days are the same, and there are always lots of interesting things going on.

COMPARATIVE REVIEW

Unpacking a Punch

The *VB* Comparative bandwagon moves on to *Windows NT* (workstation) this month, seven months having passed since we last looked at this platform.

Fifteen products were submitted for review. There is the usual collection of names, the only noticeable absentees being products from *Trend Micro Inc* and *Panda Software*.

Detection Rate Tests

Unsurprisingly, the customary *VB* test-sets were used for the detection tests (Polymorphic, Standard, Macro and In the Wild) with the In the Wild (ItW) set aligned to the January 2000 WildList. The product submission deadline was 31 January 2000.

A fifth test-set was constructed from the ItW set – each sample was individually compressed, and the archive copied into its own directory. Nested archives containing each of these individual archives were also created. Both PKZIP and ARJ compression methods were used, thus creating six tests:

1. Samples individually ARJ'ed.
2. Samples individually ZIP'ed.
3. Contents of set 1, compressed within a single ARJ.
4. Contents of set 1, compressed within a single ZIP.
5. Contents of set 2, compressed within a single ARJ.
6. Contents of set 2, compressed within a single ZIP.

Detection of the ItW samples within each of these six sets was measured during on-demand scanning, and, for those products that supported it, on-access scanning. For simplicity, within this review these results are expressed as number of missed samples and simple 'detected' percentages, as opposed to the more familiar normalized percentages.

The ability of each product to handle various types of file archives was also reviewed. For this a small set of files based on the *EICAR* test-file was used.

Complete detection rate results are provided within the large tables and a summary is presented beneath each product heading. A complete list of the samples used in each of the test-sets can be found at the URL detailed at the end of this review.

Performance Tests

The usual speed tests were performed – that is, on-demand scanning speeds returned against executable and OLE2 file scanning. Additionally, and in keeping with the emphasis upon archive handling in this review, the on-demand scanning speeds against archived executables and OLE2 files were also measured.

The scanning speed tests double up as false positive tests, and for the first time in *VB* Comparative Reviews, the criterion of 'no false positives' is added to the VB100% award. This includes only 'full' false positives, and not files flagged as 'suspicious'. To complement the scanning speed tests, the overhead of each of the on-access scanners has also been assessed. The usual process of measuring the time taken for a set of files to be copied between directories on a local drive was performed. A single machine (disconnected from the network) was used for all such tests. The results are provided within this review relative to a common baseline (with no on-access scanning) of 15 seconds.

Similar tests were performed to measure the overhead of scanning file archives for products that supported such a facility. Most of the products were designed not to support on-access archive handling, due to the large impact it can have upon performance. On-access archive handling results are presented as percentages of the baseline times measured without any real-time scanner active.

Aladdin eSafe Desktop v2.2 (31/01/2000)

ItW File	98.1%	Macro	91.9%
ItW File (o/a)	98.0%	Standard	95.3%
ItW Overall (o/d)	98.2%	Polymorphic	86.4%

A lot of anti-virus products adopt very similar user interfaces and one can step from one to the next with relative ease. Not so *eSafe Desktop* from *Aladdin Knowledge Systems* – familiarity with the rather individual interface is extremely helpful.

The detection rates observed were slightly disappointing, perhaps not living up to the claim of 'You are now free to connect and surf the Internet without fear of virus and vandal attacks' presented during installation. Ignoring the results against the Standard, Macro and Polymorphic sets, *eSafe* should have coped better with the ItW set, from which 21 samples were missed (including Win32/Oporto, the polymorphic W97M/Ded.A and a couple of the variants of VBS/Freelinks).

eSafe handles a good selection of archive formats but, unfortunately, does not fully support nested archives – it only detects the first infected file within a nested archive

VIRUS BULLETIN **©2000 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /2000/$0.00+2.50**

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Missed	%	Missed	%	%	Missed	%	Missed	%	Missed	%
Aladdin eSafe Desktop	0	100.0%	21	98.1%	98.2%	299	91.9%	273	86.4%	73	95.3%
Alwil AVAST32	0	100.0%	8	98.3%	98.4%	128	96.3%	98	89.1%	32	95.7%
CA InoculateIT	0	100.0%	0	100.0%	100.0%	1	99.9%	17	97.8%	5	98.9%
CA Vet Anti-Virus	0	100.0%	0	100.0%	100.0%	30	99.2%	265	94.4%	7	98.5%
DialogueScience DrWeb	0	100.0%	0	100.0%	100.0%	25	99.2%	0	100.0%	17	97.3%
Eset NOD32	0	100.0%	0	100.0%	100.0%	4	99.8%	2	99.5%	7	98.5%
F-Secure Anti-Virus	0	100.0%	0	100.0%	100.0%	0	100.0%	0	100.0%	4	99.1%
GeCAD RAV	0	100.0%	2	99.8%	99.8%	24	99.3%	17	97.8%	13	98.0%
Grisoft AVG	0	100.0%	14	97.3%	97.4%	49	98.6%	124	91.8%	42	97.3%
Kaspersky Lab AVP	0	100.0%	0	100.0%	100.0%	8	99.7%	0	100.0%	1	99.8%
NAI VirusScan	0	100.0%	10	98.1%	98.2%	19	99.6%	17	97.8%	17	97.3%
Norman Virus Control	0	100.0%	0	100.0%	100.0%	6	99.7%	289	90.7%	4	99.1%
SoftWin AntiVirus eXpert	1	96.4%	50	95.3%	95.5%	66	98.1%	1573	82.7%	189	89.0%
Sophos Anti-Virus	0	100.0%	0	100.0%	100.0%	82	97.6%	191	95.1%	24	97.8%
Symantec Norton AntiVirus	0	100.0%	0	100.0%	100.0%	54	98.4%	265	94.2%	5	98.9%

and therefore scored poorly on the archived ItW sample tests. Detection of the individually archived (ARJ and ZIP) samples was as for the uncompressed, with 21 samples being missed.

Data for the speed tests is incomplete due to the fact that *eSafe* consistently hung the test machine whilst scanning a number of executables in the Clean sets. Consultation with the developers identified this problem to be due to a recently discovered bug.

Alwil AVAST32 v3.0.219 (31/01/2000)

ItW File	98.3%	Macro	96.3%
ItW File (o/a)	99.7%	Standard	95.7%
ItW Overall (o/d)	98.4%	Polymorphic	89.1%

Despite sporting a somewhat updated interface from that seen in previous incarnations (though still bearing the cartoon mouse), *AVAST32* was the same as ever to test. Previously, it has skirted close to earning a VB100% award but failure to scan a variety of file types resulted in samples infected with Win95/Babylonia, VBS/Freelinks and VBS/BubbleBoy kept the award at bay once more.

On-access detection was measured by setting *AVAST32* to scan on file writes, and then using XCOPY to copy the test-set to a local drive. Detection rates were higher than those observed on-demand, predominantly because the product defaults to include 'All Files'. As observed in previous Comparatives, a couple of samples infected with the 1003- and 1019-byte variants of Win95/CIH were missed from the ItW set during on-access scanning.

Speedwise, *AVAST32* sits at the slightly slower end of the range exhibited by the other products. Sadly, a false positive was registered in the Clean set, a file unjustly being reported as infected with Tequila.2468.

The ARJ compression format was not handled by the product submitted, neither were nested archives. Both these factors caused poor overall figures in the archived ItW set. Eight infected samples were missed from the set of individually zipped samples – Set 1 – the same as were missed during the regular ItW tests. This was thanks to the omission of certain file types from the default extension list.

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Desktop	0	100.0%	22	98.0%	98.1%	299	91.9%	273	86.4%	73	95.3%
Alwil AVAST32	3	89.2%	3	99.7%	99.3%	128	96.3%	98	89.1%	32	96.0%
CA InoculateIT	3	89.2%	0	100.0%	99.5%	1	99.9%	17	97.8%	5	98.9%
CA Vet Anti-Virus	0	100.0%	0	100.0%	100.0%	30	99.2%	765	91.7%	10	98.3%
DialogueScience DrWeb	n/t	n/t	3	99.8%	n/a	43	98.8%	0	100.0%	14	97.4%
Eset NOD32	0	100.0%	0	100.0%	100.0%	4	99.8%	1	99.61%	7	98.5%
F-Secure Anti-Virus	0	100.0%	0	100.0%	100.0%	1	99.9%	0	100.0%	4	99.1%
GeCAD RAV	n/a	n/a	2	99.8%	n/a	24	99.3%	17	97.8%	13	98.0%
Grisoft AVG	0	100.0%	15	97.8%	97.9%	55	98.5%	292	89.1%	59	95.7%
Kaspersky Lab AVP	0	100.0%	0	100.0%	100.0%	8	99.7%	0	100.0%	1	99.8%
NAI VirusScan	0	100.0%	1	99.9%	99.9%	19	99.6%	17	97.8%	1	99.8%
Norman Virus Control	3	89.2%	0	100.0%	99.5%	6	99.7%	288	90.7%	4	99.1%
Sophos Anti-Virus	0	100.0%	0	100.0%	100.0%	82	97.6%	191	95.1%	24	97.8%
Symantec Norton AntiVirus	0	100.0%	0	100.0%	100.0%	54	98.4%	265	94.2%	12	97.9%

CA InoculateIT v4.53 (28/01/2000)

ItW File	100.0%	Macro	99.9%
ItW File (o/a)	100.0%	Standard	98.9%
ItW Overall (o/d)	100.0%	Polymorphic	97.8%

Complete on-demand and on-access ItW file coupled with no false positives in the Clean set was not enough to earn the first *Computer Associates'* (*CA*) offering, *InoculateIT*, another VB 100% award. Unfortunately, failure to detect infected boot sectors with invalid BPBs was to blame – a fact that has been reported in previous Comparatives.

Earlier reviews have commented upon the slight instability of the on-access scanner. Thankfully, such worries seem unnecessary now – *InoculateIT* behaved impeccably throughout testing. The detection rates measured for on-access scanning mirrored those on-demand, with only 25 samples missed across all the test-sets. The bulk of these misses were due to the complex polymorphic virus Win95/SK.8044. Other than this, a number of VBS viruses were missed, including VBS/Fool and VBS/Tune.B. A single document template infected with Iseng.A was missed in the Macro set.

The speed and overhead tests reveal *InoculateIT* to be no slouch in the engine department. Scanning speeds of well over 1500 KB/s were registered for both executable and OLE2 file scanning.

On-demand detection in the archived ItW set was perfect – all of the 712 samples within each of the six sets were detected. Following on from *AVAST32*, *InoculateIT* is another product which supports the on-access scanning of archives. In keeping with the fast on-demand archive scanning, the overhead of on-access archive scanning was approximately 150% – the smallest observed out of the five products providing such a facility.

CA Vet Anti-Virus v10.1.7.1 (31/01/2000)

ItW File	100.0%	Macro	99.2%
ItW File (o/a)	100.0%	Standard	98.5%
ItW Overall (o/d)	100.0%	Polymorphic	94.4%

Striding ahead of its *InoculateIT* stablemate, *Vet Anti-Virus*, the second of *CA's* products, provides another excellent performance earning its fourth successive VB 100% award. Its high ItW detection was matched in the archived ItW set, where *Vet Anti-Virus* managed to detect all of the infected samples in each of the six sets.

Product	File formats handled (on-demand scanner)										Nested archives?	O/A archive handling?
	ZIP	ARJ	GZIP	RAR	LZH	TAR	LHA	UUE	MIME	CAB		
Aladdin eSafe Desktop	●	●	●		●	●	●				No	No
Alwil AVAST32	●							●	●		No	Yes
CA InoculateIT	●	●	●		●	●	●	●	●		Yes	Yes
CA Vet Anti-Virus	●	●	●		●			●	●	●	Yes	No
DialogueScience DrWeb	●	●		●							Yes	Yes
Eset NOD32	●	●		●							Yes	No
F-Secure Anti-Virus	●	●	●	●	●	●	●	●	●	●	Yes	Yes
GeCAD RAV	●	●	●				●				No	No
Grisoft AVG	●	●		●							Yes	No
Kaspersky Lab AVP	●	●	●	●	●	●		●	●	●	Yes	Yes
NAI VirusScan	●									●	No	No
Norman Virus Control	●	●			●						Yes	No
SoftWin AntiVirus eXpert	●	●			●		●				Yes	No
Sophos Anti-Virus	●	●	●	●		●					Yes	No
Symantec Norton AntiVirus	●	●			●		●			●	Yes	No

Samples of Win95/WinExt.A and Win32/NewApt.F accounted for some of the misses in the Standard set. *Vet* still fails to detect the polymorphic XM/Soldier.A, along with a variety of other samples in the Macro set, including both W97M/Opey.U and W97M/Thus.G. Once again, failure to detect the A and B variants of ACG contributes to a slightly lower percentage against the Polymorphic set. All 500 samples of Baran.4968 were missed from this set during on-access scanning.

DialogueScience DrWeb v4.16 (31/01/2000)

ItW File	100.0%	Macro	99.2%
ItW File (o/a)	99.8%	Standard	97.3%
ItW Overall (o/d)	100.0%	Polymorphic	100.0%

Despite achieving complete on-demand ItW file and boot detection, *DialogueScience's DrWeb* does not earn another VB 100% award thanks to missing three *PowerPoint* files infected with the C variant of O97M/Tristate, and registering a false positive in the Clean set. The on-access component of *DrWeb*, *SpiDer Guard*, treats *PowerPoint* files as archives. By default, archives are unpacked during on-demand scanning, but not during on-access scanning, which explains why the *PowerPoint* files remained undetected. Elsewhere, the misses were predominantly due to recently introduced samples. Additionally, it was not possible to verify boot infections with *SpiDer Guard* – access to infected floppies was not denied, and no on-screen warning messages were observed. Thus on-access detection of the ItW boot samples has not been measured. Hopefully, the situation will be resolved before the next Comparative.

Though not handling a great number of archive formats, *DrWeb* coped successfully with the ZIP and ARJ files presented to it in the archived ItW test-set. It detected all of the archived ItW samples in each of the six sets.

Performance tests showed *DrWeb* returning moderate scan rates in keeping with the bulk of products. More noticeable was the on-access scanning overhead which was fairly high for both uncompressed and compressed file scanning – the latter resulting in an overhead of over 2000%, significantly larger than that for the other four products.

Eset NOD32 v1.13 (31/01/2000)

ItW File	100.0%	Macro	99.8%
ItW File (o/a)	100.0%	Standard	98.5%
ItW Overall (o/d)	100.0%	Polymorphic	99.5%

In picking up another VB 100% award in this Comparative, *NOD32* maintains its record of receiving the VB 100% in each test to which the product has been submitted. On-demand and on-access detection differed by only one sample – a single sample (from the 500 in the test-set) of the polymorphic

	Hard Disk Scanning Speed									
	Executables			OLE2 files			Zipped Executables		Zipped OLE2	
	Time (min:sec)	Throughput (kB/s)	FPs [susp]	Time (min:sec)	Throughput (kB/s)	FPs [susp]	Time (min:sec)	Throughput (kB/s)	Time (min:sec)	Throughput (kB/s)
Aladdin eSafe Desktop	n/t	n/t	n/t	1:08	1166.7	0	n/t	n/t	1:34	793.7
Alwil AVAST32	11:00	828.7	1	3:54	339.0	0	5:58	445.3	3:54	318.8
CA InoculateIT	4:24	1925.8	0	0:30	2644.5	0	3:06	857.1	0:41	1819.7
CA Vet Anti-Virus	8:45	1041.8	0	0:46	1724.6	0	4:56	538.6	1:23	898.9
DialogueScience DrWeb	18:59	480.2	1+[17]	0:51	1555.6	[1]	8:35	309.5	1:06	1130.4
Eset NOD32	2:27	3720.6	0	0:21	3777.8	0	2:54	916.2	0:48	1554.3
F-Secure Anti-Virus	17:44	514.0	0	3:35	369.0	0	2:16	1172.2	0:27	2763.2
GeCAD RAV	24:01	379.6	1+[1]	0:58	1367.8	0	11:09	238.3	1:00	1243.5
Grisoft AVG	10:24	876.5	7+[2]	0:18	4175.5	0	5:28	486.0	0:57	1311.2
Kaspersky Lab AVP	6:03	1506.7	[2]	1:13	1086.8	0	4:53	544.1	1:43	724.3
NAI VirusScan	3:58	2298.0	0	0:36	2203.7	0	7:57	334.2	1:42	731.4
Norman Virus Control	4:54	1860.3	0	0:52	1525.6	0	40:16	66.0	7:12	172.7
SoftWin AntiVirus eXpert	20:55	435.8	28+[64]	0:51	1555.6	[18]	8:04	329.4	0:56	1332.3
Sophos Anti-Virus	4:27	2048.4	0	1:20	991.7	0	3:04	866.4	1:21	921.1
Symantec Norton AntiVirus	10:01	910.0	0	0:58	1367.8	0	5:20	498.2	1:02	1203.3

W97M/Splash.A was missed during on-demand scanning. The remainder of the misses were partly attributable to JS/Kak.A, W97M/Garb.A, variants of VBS/Tune, and the Win95/WinExt.A worm.

NOD32 displayed the highest overall on-demand scanning rates, returning throughputs of over 3500 KB/s for both executable and OLE2 file scanning. Archive scanning was a little more moderate, but still faster than the average observed across all the products. All of the samples within sets 1 and 2 of the archived ItW test-set were detected. Unfortunately, only the first nine samples within each of the nested archives (sets 3 to 6) were detected, thus causing a fairly poor overall score against this test-set.

F-Secure Anti-Virus v5.02.5528 (27/01/2000)

ItW File	100.0%	Macro	100.0%
ItW File (o/a)	100.0%	Standard	99.1%
ItW Overall (o/d)	100.0%	Polymorphic	100.0%

F-Secure Anti-Virus (*FSAV*) has undergone something of a makeover since its last appearance in a *VB* Comparative. A quick glance at the results shows that the high detection rates associated with this product still remain. Only four samples, all from the Standard set, were missed across all of the test-sets – these were VBS/Tune.B, VBS/Fool and the E and F variants of Win32/NewApt.

If high detection rates have come to be associated with *FSAV*, then so has a degree of sluggishness, owing to the use of two engines (*AVP* and *F-Prot*). Though not the slowest scanner, *FSAV* was at the slower end of the pack. Interestingly, *FSAV* is the only product to return greater throughputs (almost twice as large) for archive file scanning compared to non-compressed file scanning.

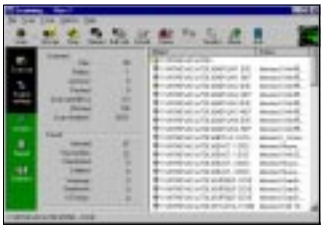
Scanning logs are now generated in HTML, with a hyperlink to the *F-Secure* on-line virus description library for each reported infection. Though a nice feature for users, setting *FSAV* to scan a large virus collection resulted in various 'out of

memory' errors, and no scanning log was produced whatsoever. The test-sets were scanned individually therefore, and a separate log for each was thus generated successfully.

FSAV handles an impressive array of archive formats, and provides the option to enable real-time archive scanning if so desired. During on-demand scanning of the archived ItW set, all of the individually compressed samples were detected (sets 1 and 2), but four compressed (ZIP or ARJ) HLP files infected with Win95/Babylonia.A were missed from each of the nested archives (sets 3 to 6). The same samples were also missed during real-time scanning of the archived ItW set, and a single Babylonia.A-infected executable was also missed from all of the sets.

GeCAD RAV v7.6.360 (30/01/2000)

ItW File	99.8%	Macro	99.3%
ItW File (o/a)	99.8%	Standard	98.0%
ItW Overall (o/d)	99.8%	Polymorphic	97.8%



After a couple of VB 100%-worthy performances in the latter half of 1999, *GeCAD's Romanian Anti-Virus* (*RAV*) puts in another strong performance this time around. Not strong enough for a VB 100% award, however, thanks to missing an OCX (ActiveX control) file infected with the recently seen Win32/Funlove and a VxD infected with the polymorphic Win95/Fono. Furthermore, on-access ItW boot sample detection rates could not be measured since *RAV Monitor* provided no such facility in the submitted product.

RAV's main weakness, when presented with the archived ItW set, was its inability to cope with nested archives. Accordingly, none of the archived samples compressed within the single ZIP or ARJ archive in sets 3 to 6 were detected. Detection of the individually archived samples was achieved: the same two samples as were missed in the conventional detection tests were missed in set 1 (ARJ compression used). Against set 2 (containing individually zipped samples), in addition to these two samples, a handful of others were also missed.

Grisoft AVG v6.0.116 (31/01/2000)

ItW File	97.3%	Macro	98.6%
ItW File (o/a)	97.8%	Standard	97.3%
ItW Overall (o/d)	97.4%	Polymorphic	91.8%

The detection rates observed for *AVG* appear a little lower than those observed in recent Comparatives. Most obvious was the failure to detect a series of ItW viruses – namely Win32/Oporto, the destructive Win32/Kriz.4029, VBS/BubbleBoy and the JO variant of XM/Laroux. A number of samples were missed elsewhere in the test-sets, the performance being poorest in the Polymorphic set where samples infected with ACG.B, Win95/SK.8044 and Win95/SK.7972 were missed.

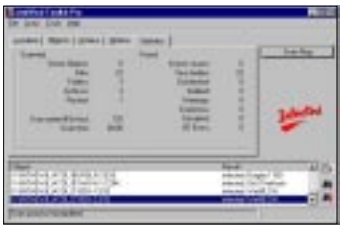
AVG performed identically with each of the sets within the archived ItW test-set – the same samples were missed in each as were missed in the regular (non-compressed) test-sets.

Traditionally fairly anonymous in the performance tests, it was surprising to observe *AVG* reproducibly returning very high throughputs during OLE2 file scanning. Sadly, a number of false positives were registered during scanning of the Clean set, caused by the overkeen heuristics. The overhead of the relatively recently introduced on-access scanner was in keeping with the bulk of other products.

Kaspersky Lab AVP v3.0.132.4 (29/01/2000)

ItW File	100.0%	Macro	99.7%
ItW File (o/a)	100.0%	Standard	99.8%
ItW Overall (o/d)	100.0%	Polymorphic	100.0%

Three viruses account for all the samples missed by *Kaspersky Lab's AVP Platinum* tested in this review. The misses were W97M/Opey.U, the potentially destructive W97M/Thus.G from the Macro set and VBS/Tune.B from the Standard set.

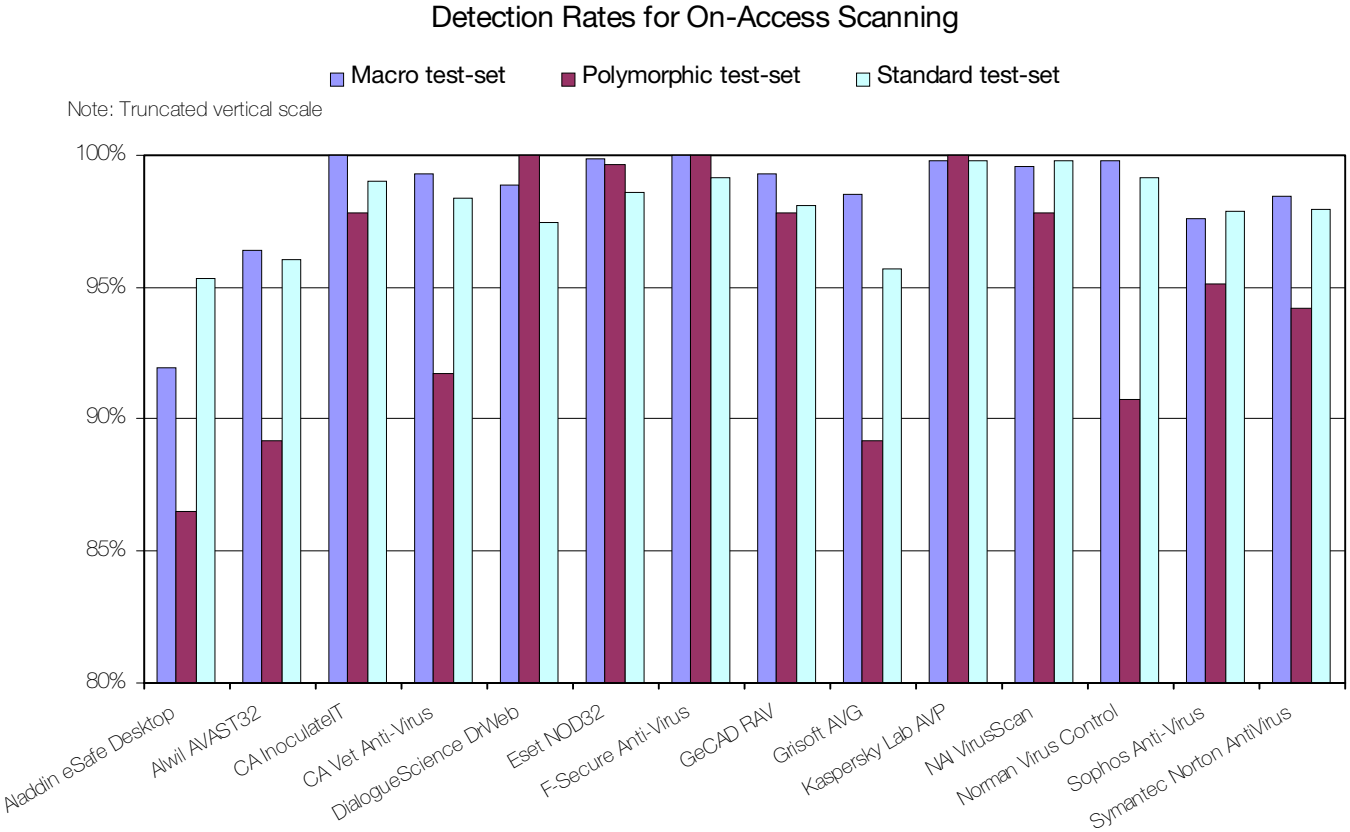


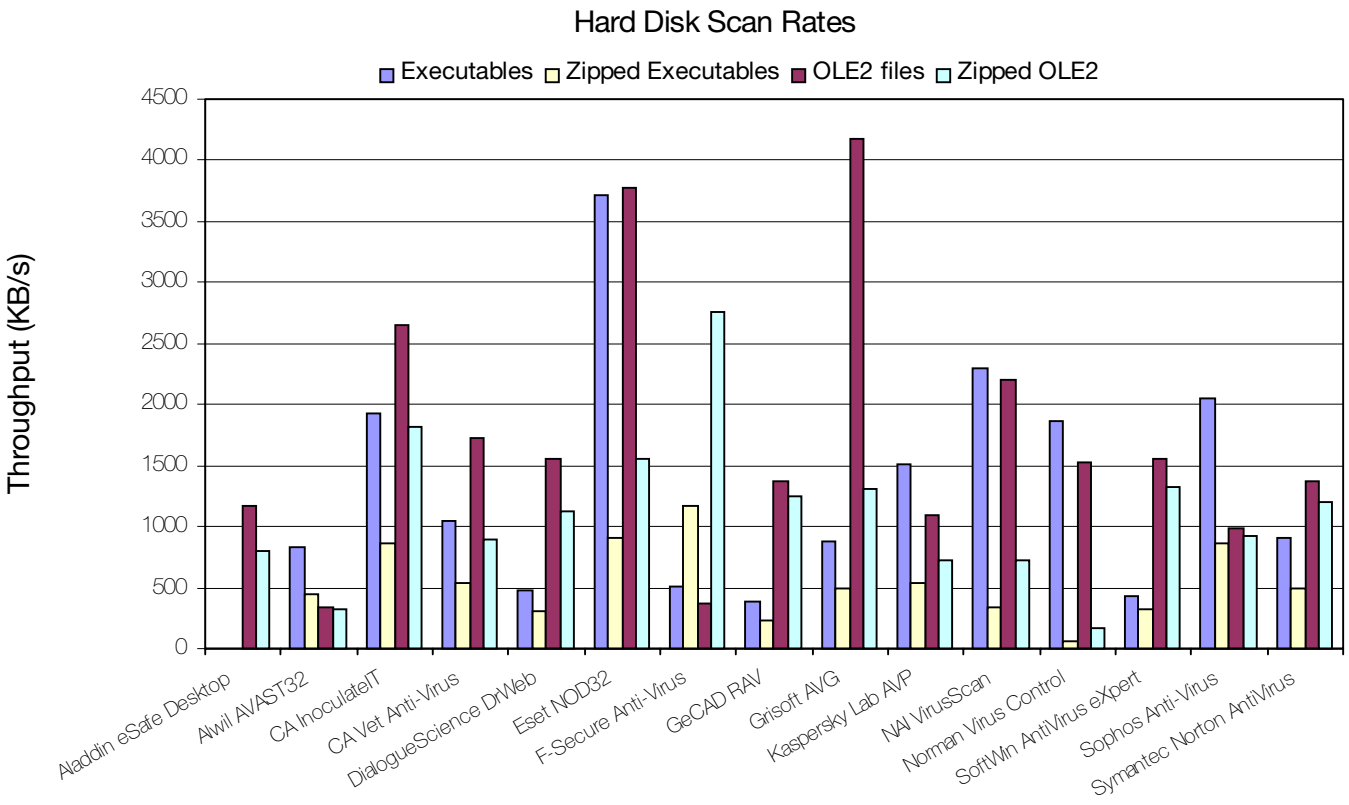
Complete ItW file detection was maintained when *AVP* was pointed to the archived ItW set, with all samples being detected across each of the 6 sets during both on-demand and on-access scanning. As can be seen, the overhead of on-access archive scanning was fairly large (as might be expected, hence the exclusion of such a facility in the majority of the products) at just over 1200%.

NAI VirusScan v4.0.3a.4062 (26/01/2000)

ItW File	98.1%	Macro	99.6%
ItW File (o/a)	99.9%	Standard	97.3%
ItW Overall (o/d)	98.2%	Polymorphic	97.8%

Compared to recent performances by *VirusScan*, the detection rates presented here are slightly disappointing. This was due mainly to the product failing to scan sufficient file types in its default configuration. VBS, HLP and OCX files (among others) were skipped, thus causing a variety of misses to be registered across the test-sets. These misses included samples of VBS/Freelinks, Win95/Babylonia.A, and Win32/FunLove from the ItW set, keeping the VB 100% award at bay. The submitted version of *VirusScan* only supported the ZIP archive format, and did not scan within nested file archives. Only set 2 in the archived ItW





test-set yielded any detections therefore, the only misses mirroring those listed above for the regular (non-compressed) ItW tests.

Speedwise, *VirusScan* returned high throughputs for both executable and OLE2 file scanning, and the overhead of the on-access was in line with that of the other products. No false positives were recorded against the Clean sets.

Norman Virus Control v4.73 (28/01/2000)

ItW File	100.0%	Macro	99.7%
ItW File (o/a)	100.0%	Standard	99.1%
ItW Overall (o/d)	100.0%	Polymorphic	90.7%

Norman Virus Control (*NVC*) puts in another strong performance, but failing to detect three boot sectors infected with ItW viruses (those with invalid BPBs) prevents it from picking up its fourth successive VB 100% award. The weakest area of detection was observed in the Polymorphic set, owing to samples infected with the A and B variants of ACG, Win95/SK.8044 and Win95/SK.7972 being missed.

NVC returned fairly fast scanning speeds against the executable and OLE2 file sets, but slowed down dramatically when scanning the same files zipped. When faced with the archived ItW set, *NVC* ploughed though the individually archived samples, detecting all of the samples (sets 1 and 2) successfully. A large number of the samples were missed in the nested archive sets – only 547 samples were detected in sets 3 to 6.

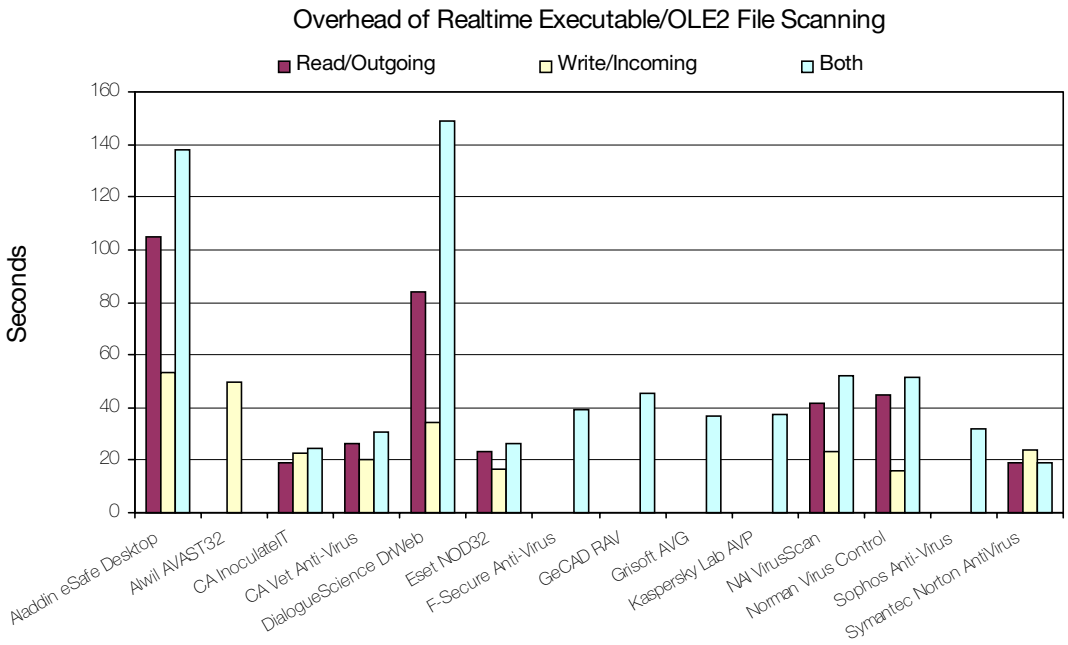
SoftWin AntiVirus eXpert (31/01/2000)

ItW File	95.3%	Macro	98.1%
ItW File (o/a)	n/t	Standard	89.0%
ItW Overall (o/d)	95.5%	Polymorphic	82.7%

A new face in the *VB* Comparative crowd, and the second product from Romania, is *AntiVirus eXpert* (*AVX*) from *SoftWin*. As expected given its virgin status, *AVX* missed a number of samples across the test-sets.

Unfortunately, on-access detection rates have not been measured because, due to a bug, the *AVX* on-access scanner failed to block access to infected files. Hopefully, the on-access component of *AVX* product versions submitted to *VB* future Comparatives can be reviewed as normal.

Whereas low detection might not be a surprise against the Standard, Macro and Polymorphic test sets, a slightly higher percentage might have been expected against the ItW set. Sadly, a number of viruses were missed here, including Win95/Babylonia.A, Win32/Oporto, TMC_Level-69, Win95/Fono, X97M/Manalo.E and X97M/PTH.D, to name but a few.



AVX worked its way happily through the archived ItW test-set, detecting 663 samples in each of the six sets, the missed samples mirroring those missed during the above tests.

In terms of performance, only the on-demand scanning speed of *AVX* has been assessed due to the aforementioned bug. Scanning speeds of approximately 450 and 1550 KB/s were returned for executable and OLE2 file scanning respectively. The throughputs dropped only slightly to just over 300 and 1300 KB/s for scanning of the zipped files.

It will be interesting to see how *AVX* measures up in subsequent reviews – one would predict a significant increase in the detection rates, which, if realised, would certainly make *AVX* a competitive product in the *VB* Comparative product arena.

Sophos Anti-Virus v3.30 (01/02/2000)

ItW File	100.0%	Macro	97.6%
ItW File (o/a)	100.0%	Standard	97.8%
ItW Overall (o/d)	100.0%	Polymorphic	95.1%

Complete on-demand and on-access ItW file and boot detection coupled with no false positives in the Clean set earns *Sophos Anti-Virus* (*SAV*) its tenth VB 100% award.

SAV missed its traditional sprinkling of viruses, a proportion of which are detected if the 'full' scanning mode is enabled, as opposed to the default 'quick' mode. New misses included samples infected with W97M/Divi.B, X97M/Weit.A, the F and G variants of W97M/Verlor, PE samples infected with Win98/Caw.1416, the E and F variants of Win32/NewApt, and the Win32/WinExt.A worm.

The tested version of *SAV* is the first in which only the archive handling product is supplied. Dealing with a variety of archive formats, *SAV* skipped happily through the archived ItW set, successfully managing to detect all of the samples within each of the six sets.

In terms of on-demand scanning speed, *SAV* is positioned at the upper end of the bulk of products, for both compressed and non-compressed file scanning. The overhead of *InterCheck*, *SAV's* on-access component, is reasonably small at a little over 100%.

Symantec Norton AntiVirus 2000 v6.00.03 (24/01/2000)

ItW File	100.0%	Macro	98.4%
ItW File (o/a)	100.0%	Standard	98.9%
ItW Overall (o/d)	100.0%	Polymorphic	94.2%

Rounding off this Comparative, *Symantec's Norton AntiVirus* (*NAV*) managed to carry on where it left off last time around, earning its ninth VB 100% award.

The bulk of the misses were registered in the Polymorphic set, thanks to samples infected with the A and B variants of ACG. A number of recent additions to the Macro set were also missed, including W97M/Melissa.AL, W97M/Thus.G, the B, C and D variants of W97M/Lys and the F and G variants of W97M/Verlor. Only a handful of samples were missed in the Standard set, most notably, and in common with a number of products in this review, the E and F variants of Win32/NewApt.

On-Demand Detection of Archived ItW sample test-set	Archived ItW Set Number											
	1		2		3		4		5		6	
	No. Missed	% Detected	No. Missed	% Detected	No. Missed	% Detected	No. Missed	% Detected	No. Missed	% Detected	No. Missed	% Detected
Aladdin eSafe Desktop	21	97.1%	21	97.1%	711	0.1%	711	0.1%	711	0.1%	711	0.1%
Alwil AVAST32	n/a	n/a	8	98.9%	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
CA InoculateIT	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%
CA Vet Anti-Virus	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%
DialogueScience DrWeb	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%
Eset NOD32	0	100.0%	0	100.0%	703	1.3%	703	1.3%	703	1.3%	703	1.3%
F-Secure Anti-Virus	0	100.0%	0	100.0%	4	99.4%	4	99.4%	4	99.4%	4	99.4%
GeCAD RAV	2	99.7%	7	99.0%	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Grisoft AVG	14	98.0%	14	98.0%	14	98.0%	14	98.0%	14	98.0%	14	98.0%
Kaspersky Lab AVP	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%
NAI VirusScan	n/a	n/a	10	98.6%	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Norman Virus Control	0	100.0%	0	100.0%	165	76.8%	165	76.8%	165	76.8%	165	76.8%
SoftWin AntiVirus eXpert	49	93.1%	49	93.1%	49	93.1%	49	93.1%	49	93.1%	49	93.1%
Sophos Anti-Virus	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%
Symantec Norton AntiVirus	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%	0	100.0%

The archive handling capabilities of *NAV* were, like *SAV* and four other products before it, impeccable. All of the 712 samples in each of the six sets were detected successfully.

Performance-wise, *NAV* did not let itself down, returning above average on-demand scanning throughputs, and displaying a relatively small on-access scanning overhead of approximately 120%.

Summary and Conclusions

The October 1996 issue of *VB* saw the first Comparative Review of products for the *Windows NT* platform. Back then the products were still in gestation – only four of the thirteen provided real-time protection, and a number were only slightly developed from their *Windows 3.x* brethren, with little familiarity with the *NT* operating system. The situation is different now – most notably, the provision for on-access scanning is a necessity, and duly all of the fifteen products in this review comply.

Detection-wise, things have tightened up as well, with six of the products achieving complete on-demand and on-access detection of the ItW file and boot viruses. Happily, none of these products triggered any false positives in the Clean set (although *AVP* sailed close to the wind in flagging a couple of samples as suspicious), and thus each earns the VB 100% award for this review. So, congratulations to these six – *Computer Associates' Vet Anti-Virus*, *Eset NOD32*, *F-Secure Anti-Virus*, *Kaspersky Lab AVP*, *Sophos Anti-Virus* and *Symantec Norton AntiVirus*.

Investigation of the archive handling capabilities of the products proved interesting, and the results provide an additional yardstick by which to judge performance. Looking at the detection rates within the archived ItW set, six products managed to detect all of the ZIP'ed and ARJ'ed (sometimes recursively) samples, namely *InoculateIT*, *Vet Anti-Virus*, *DrWeb*, *AVP*, *SAV* and *NAV*. Two other products came close (*FSAV* and *NVC*), but failed to detect all of the samples within the recursive archives.

Five of the products submitted offered on-access archive handling – a feature whose inclusion in a product currently remains up to the individual product developers. Looking at the large overheads that were observed in testing, it is clear that all of the products are a long way from being able to set real-time archive scanning by default.

[*For the purposes of this PDF, this Comparative has been modified to correct an error in the printed version.*]

Technical Details

Test Environment: Server: *Compaq Prolinea 590*, 90 MHz Pentium with 80 MB of RAM, 2 GB hard disk, running *NetWare* 4.10. Workstations: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, CD-ROM and 3.5-inch floppy, all running *Windows NT 4.0 SP5*.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/200004/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

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 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

VB, 50 Sth Audubon Road, Wakefield, MA 01880, USA

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

The fourteenth annual Vanguard Enterprise Security Expo 2000 will be held at the Atlanta Hilton and Towers, Atlanta, Georgia, on 15 and 16 May 2000. For further information contact *Vanguard*; Tel +1 714 9 390377, or see http://www.vipexpo.com/.

Sybari Software* has added support for *Sophos Anti-Virus to its anti-virus and security groupware solutions, *Antigen for Exchange* and *Antigen for Notes*. In an unrelated announcement, ***Sophos* announces the release of *Sophos Anti-Virus* (*SAV*) *for Notes/Domino v2.0*.** For more details visit the *Sophos* Web site; http://www.sophos.com/.

The fifth Ibero-American seminar on IT security and computer virus protection will take place from 22–27 May 2000 at the Informatica 2000 International Convention and Fair in Havana, Cuba. The principal topics include anti-virus software, Internet security, e-commerce security and systems audits. For further details contact José Bidot, the Director of *UNESCO*'s Latin American Laboratory; Tel/Fax +53 7335965 or email jbidot@seg.inf.cu.

In early March, *Kaspersky Lab* launched its on-line Internet sales site. All the popular versions of *AVP* for DOS, *Windows*, *Office 2000*, *Exchange* and *Linux* are currently available to buy on-line. There are plans to add workstation products – *AVP Inspector*, *AVP for OS/2* and server products – *AVP for NT/Novell NetWare*, in the near future. See http://kasperskylabs.com for more details.

The *Computer Security Institute* (*CSI*) has released details about its 10th annual Network Security conference and exhibition this year. **NetSec 2000 will be held at the Hyatt Regency Embarcadero in San Francisco from 12–14 June.** For more details contact *CSI*; Tel +1 415 9052626 or visit http://www.gocsi.com/.

***Norman Data Defense Systems* announces a new release of *Norman Virus Control* (*NVC*) specifically for small businesses.** The *Small Business Edition* provides licensing to cover 50 modes, including a maximum of 3 servers. Prices start at £19.85 (+VAT). For details visit http://www.norman.com or email Dawn_Cook@norman.com.

In a recent battle of the heavyweights, **German computer magazine *PC Welt* pitted *Network Associates' Anti-Virus Emergency Response Team* (*AVERT*) against *Symantec's Anti-Virus Research Centre* (*SARC*)**. *AVERT* was declared 'the winner' when it scored top marks across the board in categories ranging from 'quality of response' and 'helpfulness of Web sites' to 'ease of transmission procedure', following *Welt*'s anonymous submission of a combination of infected and uninfected files to both labs.

Now available for the first time in the UK from *Centerprise International*, *GoBack* was originally designed in the USA by *Wild File Inc*. Suitable for use with *Windows 95* or *98*, *GoBack* constantly tracks hard drive activity, monitoring any changes saved to the hard disk. It only uses 10% of the disk and does not impact on system performance. Following a system crash, *GoBack* takes the computer back in time to the 'safe state' it was in before the error occurred. *GoBack* retails at £69.99 but is on special offer for a limited time at £55.99. For details email GOBACK@centerprise.co.uk.

***Panda Software* announces the release of a *Windows 2000*-compliant version of *Panda Antivirus Platinum* for single users.** The product retails at £49.99 including VAT and is for use on *Windows 95/98/NT/2000*. For details contact Shari Lovidge; Tel +44 1372 824278, email slovidge@pandasoftware.co.uk or visit the Web site http://www.pandasoftware.co.uk/.

***Microsoft* has voiced its full support for a new Computer Crimes Lab at *DuPage College* in Illinois, USA, which it helped to establish. The new Lab is aimed primarily at aiding the study of all aspects of cybercrime by police and other** law enforcement officials. *Microsoft* has also set up a 24/7 hotline, staffed by experts in law enforcement and investigative techniques, for exclusive access by students at the new Lab.

InfoSec 2000 will take place at the National Hall, Olympia, London from 11–13 April 2000. The show includes exhibitions and talks on various subjects including virus protection, firewalls, network security, e-commerce and Web security. There will also be a series of 46 free, on-floor seminars on topics such as *Windows 2000* and *Linux*. For more details contact Yvonne Eskenzi; Tel +44 2084 498292 or email yvonne@eskenzi.demon.co.uk.

For more information on the **10th Annual International Virus Bulletin Conference in Orlando, Florida**, or to view the presentation programme, visit http://www.virusbtn.com/vb2000/.