ISSN 0956-9979 JULY 2000

# VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL



## *IN THIS ISSUE:*

• **VBS-pecially for you:** Marius van Oers' Technical Analysis on p.8 covers the hot topic of VBScripting in the wake of disruption caused by VBS/LoveLetter and JS/Kak.

• **Totally topical:** Eric Chien prefaces his upcoming VB2000 paper on the malicious threats that face *Palm* Personal Digital Assistants, on p.12.

• **Win-dows some, lose some:** 18 products, the occasional new entry amid some familiar faces, were submitted to this month's Comparative Review for *Windows 98*, which starts on p.16.

## CONTENTS

# COMMENT

*“ A program is the instructions themselves … ”*

## Viruses are not Speech

[*I thought this month’s Comment page warranted an introduction of sorts. With the recent fracas culminating in the arrest of the LoveLetter author in Manila and as we still await the sentencing of David L Smith,* VB *has given Mich Kabay the opportunity to voice his controversial opinions on the legal implications of virus writing. I’m fairly confident that some of you will take exception to this article and to the suggestions made within it. As ever,* VB *welcomes your reactions. Please direct your opinions either to Mich himself, mkabay@atomictangerine.com or to us here at* VB*, editorial@virusbtn.com. Ed.*]

In the early years of the last decade, a rogue publisher caused a ruckus in some quarters by publishing a textbook with detailed code for a number of viruses. I felt that it would be a good thing to see the publisher prosecuted for public mischief, if no other laws were found to apply.

However, some people who hate viruses and despise virus writers nonetheless felt strongly that no one should be prevented from publishing virus code in any form. For example, the slippery-slope argument was invoked by one prominent member of the anti-virus community, who said ‘My concern is that if we can justify the suppression of information as “undesirable” or “potentially dangerous”, is it that much further a jump to … suppression of other “information”?’

The problem became more difficult a few years later because of the International Traffic in Arms Regulations (ITAR) of the United States. These stupid regulations restricted exports of CD-ROMs or diskettes containing source code for strong cryptographic algorithms. Among the arguments used to attack this bizarre notion were claims that the ITAR infringed the authors’ rights to free speech.

I don’t think that computer programs, let alone virus code, should be considered speech at all.

Consider a wire-board controlling a card sorter. Is the wire-board speech? Not in any sense most people would use the word. How about a paper-punch tape controlling a machine tool? What about a useful computer program expressed as machine language codes? What about the more understandable FORTRAN? Or the even more English-like COBOL or PASCAL? And what of fourth generation languages that strive to accept input such as ‘SEARCH EMPLOYEE FILE USING KEY=ID FOR ID=2345’?

In my opinion it’s irrelevant to the argument over viruses how we *represent* computer programs. A program is the instructions themselves, not the medium in which they’re coded. A program in assembler is a program whether it resides on a hard disk, a floppy diskette, or a portion of a memory array. Indeed, that sequence of computer instructions would be the program itself even were it written on papyrus, chiselled in stone, signalled by semaphore or printed in a book.

Why should we accept an excessively broad definition of speech that includes self-replicating code that hides in people’s computer systems and destroys data, violates confidentiality, or sends out forged email in the victim’s name?

And does the fact that some viruses use speech (usually writing) in their payload mean that they should be considered as somehow privileged? I don’t think so, any more than I believe that the scrawled graffiti illegally painted by vandals on private property could ever be considered protected speech.

So the next time we debate the advisability (or even the theoretical possibility) of defining laws making virus-writing illegal, I hope we can brush aside any claim that punishing virus writers is somehow an infringement of their civil liberties.

*Michel E ‘Mich’ Kabay, INFOSEC Group, AtomicTangerine Inc, USA*

# NEWS

## Hard Cell

VBS/Timofonica was yet another script virus that mass-mailed itself to the victim's *Outlook* address book. Lukewarm on the heels of VBS/LoveLetter and the pair of damp squibs known as VBS/NewLove and W97M/Resume, the media seemed primed again for a 'virus destroys Internet' story. Timofonica gave them that and more – the chance to misrepresent it as a mobile phone virus.

The general media, and particularly its broadcast arms, labelled Timofonica 'the first cell phone virus'. Most print media managed to squeeze in that Timofonica's *payload* tried to send text messages *only* to randomly-generated numbers on a Spanish email-to-Short Messaging Service (SMS) gateway. This subtlety was lost on much of the broadcast media, with at least one source reportedly claiming the virus did not work on computers, but only on cell phones. Unfortunately, a couple of trigger-happy AV vendors also issued press releases which only added fuel to the fire of misreportage ▮

## Neo-NATO Viruses

Back in April, *NATO* tried to explain away the embarrassing appearance of one of its sensitive documents on computers at a London publishing company by suggesting that a computer virus may have been responsible. On 18 June, *The Sunday Times* reported *NATO's* admission that its 'scientists have created a computer virus "by mistake", causing military secrets to find their way onto the internet.' Reputed to have near-mystical powers, the virus 'plucks documents from the hard drives of computers and sends invisible attachments to emails'.

*The Sunday Times'* report suggested *NATO* believed the virus its own staff made was responsible for the earlier 'leak'. This seems unlikely to the jaded souls compiling this column. The 'document' described in the April incident sounds exactly like the 'extra' text tacked on the tail end of W97M/Thus.K's source code. Due to the way Thus replicates, it will carry any additional 'code' added to its VBA source, even if it is not valid VBA. This appears to be what has happened to create the .K variant. None of the known Thus variants has code to extract text from the host document and inject it into the virus' VBA source code, so it seems most likely that the extra text in Thus.K was put there *by NATO staff*. One should hope that whoever put the text there did not do in it in the name of 'science' ▮

## Hello and Goodbye

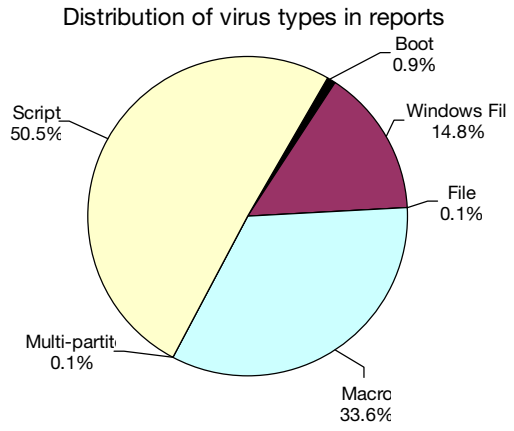
*Virus Bulletin* extends a warm welcome back to Technical Consultant Matt Ham, who takes over from Fraser Howard this month – good luck and best wishes all round ▮

### Prevalence Table – May 2000

| Virus | Type | Incidents | Reports |
|---|---|---|---|
| LoveLetter | Script | 654 | 32.2% |
| Kak | Script | 284 | 14.0% |
| Win32/Pretty | File | 166 | 8.2% |
| Marker | Macro | 127 | 6.3% |
| Laroux | Macro | 119 | 5.9% |
| Win32/Ska | File | 86 | 4.2% |
| Thus | Macro | 78 | 3.8% |
| Melissa | Macro | 63 | 3.1% |
| Freelinks | Script | 59 | 2.9% |
| Ethan | Macro | 42 | 2.1% |
| Tristate | Macro | 38 | 1.9% |
| Class | Macro | 28 | 1.4% |
| Win32/Fix | File | 20 | 1.0% |
| Proverb | Macro | 19 | 0.9% |
| Netlog | Script | 18 | 0.9% |
| Win95/CIH | File | 18 | 0.9% |
| Eight941 | Macro | 16 | 0.8% |
| Divi | Macro | 12 | 0.6% |
| Cap | Macro | 11 | 0.5% |
| Smac | Macro | 10 | 0.5% |
| Bobo | Macro | 9 | 0.4% |
| VCX | Macro | 9 | 0.4% |
| IIS | Macro | 8 | 0.4% |
| Others [1] | | 134 | 6.6% |
| Total | | 2028 | 100% |

[1] The Prevalence Table includes a total of 134 reports across 64 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 842 reports in May) have been omitted from the table this month.



# LETTERS

## Dear Virus Bulletin

### Mission: Impossible?

I found Randy Abrams' article on educating users interesting, but there are some parts of his advice which seem almost impossible to implement. In particular, I doubt that the advice to 'save the file to disk and verify that it is what you think it is' can be followed by anyone other than an expert. I am not even sure I understand the sentence following that – 'That is, if it has a picture icon, it must not have an .EXE extension.' – aren't all icons pictures?

Also, the relationship between extensions and icons are often changed by the installation of legitimate software, or user actions, so we cannot depend on a file type having a recognisable icon. I do not want to blame him unjustly for the actions of his employer, but I am sure Mr Abrams is aware that some operating systems deliberately conceal parts of file names from users, making it difficult for them to decide what the 'extension' is. The fact that the operating system settings can be changed to show the full file name only increases the possibilities for confusion.

I think that ordinary users can be trained to follow simple, unambiguous security rules, but our current complexity makes it difficult (impossible?) to frame the right rules.

*Allan Dyer*
Yui Kee Co, Ltd
Hong Kong

### Not Impossible, Just Difficult

[*Randy was invited to respond to these comments and concerns. Ed.*] I'm not suggesting users learn to identify file headers, but users can learn to determine if an extension is .TXT or .VBS. With respect to icons, in my training presentations I teach that an attachment with a JPEG icon must not have a .EXE extension when saved to disk.

With 30,000+ individual opinions at *Microsoft* sometimes justifiable but differing positions become defaults for products. Personally, I recommend displaying extensions. I feel this is part of computer education necessitated for security due to abuses by a few anti-social elements of the computing society.

Martin Overton (' … user AV education is generally a waste of time') states ' … it is ultimately (in most cases) a human being pressing the buttons, and this is the root of the problem'. People will open attachments, even when saved to disk first. Until AV protects people from themselves, education must be part of our strategy. If the industry is to advance technology, we must advance our users' education.

We count our failures and ignore successes. For each user who opened LoveLetter, scores of users *learned* not to. You probably have an extremely high success rate, but only count failures. Education is not perfect.

Perhaps some people won't learn. In a corporation of 5,000 people, each student success can easily mean 5,000 fewer viral emails, or hours less data recovery.

Mr Dyer correctly says 'complexity makes it difficult (impossible?) to frame the right rules'. Difficult is precisely why a higher quality of education is imperative. In my opinion, 'impossible' is what you tell an engineer to make him do something!

*Randy Abrams*
Microsoft Corp
USA

### Unfair Comment

I wasn't at all happy with the Comment page in the June issue. After a gratuitous plug for Joe Wells's new anti-virus venture, it settled into a pageful of postured pontification about the anti-virus industry.

Joe must have been really busy on LoveLetter day, 'monitoring both the perception and the reality of the virus problem' (as he puts it). And he uncovered some astonishing facts! Technical support lines were busy, and the Internet was pretty slow. Where would we have been without that useful research, eh? Apart from having slightly less traffic at an already stressful time, of course.

His conclusion? That 'we failed'. C'mon, Joe. If there's nothing you can do to help, at least get out of the way and don't sit on the sidelines whingeing. Considering the hard work that many genuine anti-virus experts did put in on that day; considering the thanks we received from those who felt we didn't fail them; and considering that you don't usually accept advertisements in *VB*, could I ask that you stop printing this sort of stuff in your publication?

*Paul Ducklin*
Sophos Plc
UK

### Point Taken, Partly

Joe Wells suggests that the AV industry has become a service-based industry and how, during the LoveLetter episode, it failed to provide that service. To this extent, I somewhat agree with his comments but there are some things that should be pointed out. The fact is that things could have been much worse – I think most users, readers, and even virus writers would agree. What did happen is that processes put in place, from Melissa and ExploreZip, were

tested by LoveLetter, and didn't fare too well. Unfortunately an episode like LoveLetter is one of the only ways to get these processes tested. What was put into place may have worked, but this virus hit so many more people that we will never know.

AV *has* become a service, and one that is now beginning to emerge from the 'best detection' product-based solutions to one that needs to tout the 'best services' to support your environment. From the product-based solutions, which are offered with the technical support add-in, will come a new era in AV control, services and support. For AV companies to survive this change they will need a complete service package of support and information, both Web-based and in other formats. I'm not going to go into what we are doing to test our new services – this isn't a sales pitch, or the place to do it. However, from LoveLetter I know we learned what additional measures can be taken, but I undertake no promise here, nor do I think that any of us can make a promise regarding what customers can depend on during the next 'outbreak'.

Each outbreak I have been through has been a bit different, but there are commonalities. These commonalities are what is needed for a foundation to offer all customers solutions and information in order to take the approach they need to protect their environment. In addition to the activity we need to take as an industry, there is going to need to be some measure of support by the user community. We can stand here all day and preach the gospel about don't do this, and don't touch that but until this happens, and I mean really happens, outbreaks may continue to happen. Users cause virus outbreaks, not AV companies, nor virus writers.

Finally, now that the researchers at *WarLabs* have made their assessment, what advice is forthcoming? I think that a research group should do research, generate a conclusion, and make recommendations. Perhaps such recommendations have been left for those at the helm.

*Vincent Gullotto*
AVERT
USA

## AV for German Magazines

In Germany, one can find problems with how AV products are tested in nearly all non-specialised computer magazines. Usually the results don't count for much – no real viruses are used in testing, categorisations are unsatisfactory – it often looks like a range of products is shaken up together and a winner picked at random.

This may sound unfair, but it was the reason behind the German *IT Business* magazine and five major anti-virus companies deciding to change things. This consortium drew up a list of useful test criteria and it was decided that results would be shared between *IT Business* and other interested German publications. Of course, only objective criteria would be tested, not subjective issues like the handling of the user interface.

Some time after this, the University of Magdeburg and I started the actual testing – sponsored by the German branches of major anti-virus companies. The idea was successful; *CHIP*, *FreeX*, *Network World*, *PC Shopping* and *PC-Welt* have all used our results as a basis for further tests. What is more, they were interested not only in the retail software test, but in the client/server-based products. It is clear that most users in small and middle-to-large German companies do not have a good overview of the products available along with their advantages and problems, because IT resources are rare and often there is not even a full-time Administrator available.

In March, we tested *Windows 98*, *NT*, *NetWare*, *Linux*, *Exchange* and *Lotus Notes* products under several very challenging conditions, including on-demand and on-access scanning, supported file and archive formats, some technical aspects and remote administration, update strategy and log files, groupware functionality and content filtering.

More information on the tests can be found at the Web site http://www.av-test.com (English) and www.av-test.de (German). At the moment, only XLS and DOC files are available, but we are developing tools to make everything visible in HTML. This includes the facility to make direct comparisons between products and choose the three best solutions to suit a user's criteria.

In September, the tests specified above will be repeated and additional tests on different mail gateways (SMTP proxies), *Windows 2000* and on *FreeBSD* will be performed. To supply supporting German magazines better with current results, ItW tests will be performed regularly each month and later each week on every platform using the most current WildList and WildCore.

*Andreas Marx*
University of Magdeburg
Germany

## Who's Zoomin' Who?

I see so many of these hoaxes that 'prey on people's ignorance' that I begin to wonder what it's really all about. It's publicized that it's about a few demented individuals (the virus authors) wanting fame and glory. This makes no sense to me. How can you get famous if you are anonymous? If the authors try to take credit, people won't believe them and will think they're nuts unless they can prove it and then they wind up in jail.

I wonder how the virus protection software companies make money once everyone has their latest software. Since it's obvious that the supposed few demented virus authors can't benefit from spreading hoaxes and/or viruses it must be the virus protection software companies themselves propagating fears in order to justify the sales of the newer versions of their software.

*Ron Hazard*
California, USA

# VIRUS ANALYSIS

## Smash and Burn

*Adrian Marinescu*
*GeCAD srl, Romania*

The evolution of *Windows* viruses has reached a certain maturity level this year. We have seen how many complex viruses designed for *Windows* operating systems use more and more techniques adapted from their DOS counterparts. Most of the methods used in DOS viruses are already used in Win32 – for instance, almost all polymorphic techniques are implemented. The only method not used under Win32 up to now is metamorphism, but in less than a year we are bound to see an ACG or Lexotran correspondent for Win32.

Stealth seems to be harder to implement under Win32 – even now, there are no true Win32 stealth viruses. There are a few *Windows 9x* viruses that have stealth capabilities, but with a lot of incompatibilities generated by the methods used, for example, interrupt tracing replaced by IFS chain parsing. Win95/SK was the first virus to use this method to get the first entry in the IFS chain. Some DOS methods have been improved on under Win32. For instance, the entry point obscuring method is more often used in Win32 infectors. In my opinion, Win32/Orez is the first Win32 virus to raise big problems about detection – right now there are only a few AV products able to detect it, and none able to clean it.

Win95/Smash.12288 is an interesting virus that was discovered in mid-April this year. In spite of the fact that it was not a problem for the anti-virus community, I consider it a very good indicator of how the evolution of Win32 viruses is set to develop in the next few years.

Smash is a dangerous, stealth, memory resident, polymorphic *Windows 9x* virus. It will not replicate under *Windows NT* due to the mechanisms and functions it employs – *Windows 95*-specific ones. The infection method is *Windows NT*-compliant – infected files will be loaded by the *NT* process loader without any error message.

### Running Infected Files

When an infected file is executed, the virus body receives control from the polymorphic engine. After using a simple pre-fetch trick to defeat weak code emulators, Smash tries to install itself resident in memory. Using a precalculated CRC table, it will import 17 API addresses from both KERNEL32.DLL and USER32.DLL. By calling the GetVersion API, it can establish whether it is already resident in the system.

Also using the same API, the virus can determine if the *Windows* version is suitable for the virus code – if the operating system is anything other than *Windows 9x* the virus code will simply restore the execution to the original entry point. Then, Smash uses a new method to execute its code in Ring0.

Many viruses use a straightforward method, based on CIH's code, to gain Ring0 residency by patching the Interrupt Descriptor Table and then generating an interrupt which will be executed under Ring0. After seeing tens of viruses using this same method, I was quite surprised to discover a new trick inside the Smash virus.

Smash uses an undocumented API from KERNEL386.EXE called PrestoChangoSelector(). This function was included by *Microsoft* in *Windows 3.x* for compatibility with DOS applications, but never documented. Using this API and thunking, the virus is able to execute a part of its own code as 16-bit code. Then, using the same trick as the Win95/SK virus family, it uses the DPMI server to get a block of memory and switch the execution from V86 mode to protected mode. Then it switches from protected to Ring0, copies its code to Ring0 memory, and calls it.

From Ring0, the virus allocates a block of memory in the VxD area, using the HeapAllocate kernel API. In this block, Smash will copy the virus code and set, then hook the IFS to receive control each time a file operation is requested. After the memory installation, the virus will re-encrypt its body to prevent process-scanners from detecting it in the infected process memory, and pass the execution to the original program.

### Infecting Files

When a File/Open request is made to the IFS Manager, the virus body receives control. First, it checks if the file is suitable for infection – it has to be a clean PE file. Smash contains two different infection methods: the first is for files importing GetModuleHandleA from KERNEL32.DLL and the second is for files without this import. The virus mixes the virus blocks, generates two polymorphic decryptors and encrypts the virus body corresponding to the generated layers.

Then, it adds a new section to the executable file. This new section's name is generated by choosing the name of an existing section, selecting a substring from it, and adding random characters to both the beginning and the end of the name itself. This way, many of the generated names appear less suspicious.

Smash writes the newly generated shape at the end of the file, registers the newly created section in the section table and then modifies the program's start address to run the polymorphic decryptor. It goes on to set the infection marker to be used by the stealth routine. All infected files will grow in size by 12 KB.

### Payload

If an infected file is executed on 14 July and Smash is not resident in the system, the virus will call its payload routine. It will display a SYSTEMMODAL message box on a blue screen with the following text:


Next, it will attempt to patch the C:\IO.SYS file, inserting a Trojan at the end of the file and modifying the program entry point. On the next reboot, the Trojan code receives control. It will trace Int13h by using a step-by-step technique widely employed in complex DOS viruses. This tracer will stop when the Int13h code is in the segment F000h. Then, it will display the following message: 'Formatting hard disk … '. Using the BIOS Int13h address, the Trojan will attempt to write trash data to the first cylinder of the first hard disk.

### Stealth

Until now, there have only been a few stealth *Windows 9x* viruses (Win95/Zerg and Win95/Filth). Smash is a full stealth virus for *Windows* applications. In the installed IFS hook, the virus will intercept the following IFS functions: FINDOPEN, FINDNEXT, CLOSE, OPEN and SEARCH. On each of the operations mentioned, Smash will check the file size – if it is bigger than 12,288 bytes, it will check the file's last write time. If, when divided by 32 the remainder equals 11, the file is assumed to be infected and Smash will substract 12 KB from its size to hide the virus body.

This method is not very safe – if a clean file is catalogued as being infected the file size reported by the IFS Manager will be incorrect and the file will be corrupted. The stealth feature makes detection very difficult when the virus is memory resident. It may be instrumental in making memory scanning and disinfection a mandatory feature in the near future.

### Polymorphism

Smash uses a complex polymorphic engine. The virus code is divided, using a technique similar to that of the BadBoy family, into 64 blocks of variable length. These blocks are indexed using a table of positions and lengths. Each block links itself to the following block, using that table. By changing the offsets of the blocks and reindexing the table, the functionality of the virus will not change, but its binary image will be different. To make detection harder, the mixed blocks are encrypted with two polymorphic layers. The first layer partially decrypts the virus code on the stack and calls the second layer, which fully decrypts the virus at the original location in memory, then calls the real entry point of the virus.

The garbage code generated in the polymorphic code is not very complex, and it can easily be emulated by most of today's code emulators without problems. Also, the virus code is encrypted only using operations on DWORDs, so cryptoanalytic methods can easily be employed.

### Epilogue

Win95/Smash.12288 is not very stable and it has not been reported to be in the wild. It is known to be able to replicate correctly under *Windows 98*, but it sometimes generates system error messages.

While right now script and macro viruses are the hottest subjects as far as the AV industry is concerned, *Windows* viruses continue to deserve special attention – after all, *Windows* is still the most popular operating system.

As for the future, we may soon see embedded *Windows* viruses carried by script worms. Just imagine a worm like VBS/LoveLetter dropping a complex polymorphic virus like Win32/Orez!

### Win95/Smash.12288

**Aliases:** None known.

**Type:** Memory resident, polymorphic, stealth *Win9x* infector.

**Infection:** PE executable files.

**Self-recognition in memory:** Calling the GetVersion API with eax='SMAS' and edx='HHH?' will return 'FUCK' in esi.

**Self-recognition in files:** Last write time attribute is set to a value which, if files are divided by 32, will result in a remainder of 11.

**Hex pattern in memory:**

```
8D74 2424 8B46 0883
F80B 0F84 9100 0000
```

**Hex pattern in files:** None possible.

**Removal:** Boot from a clean floppy disk, delete infected files and restore from backups. If the IO.SYS file is corrupted, restore the altered system file from the *Windows* recovery disk.

# TECHNICAL ANALYSIS

## Following the Script

*Marius van Oers*
*NAI, Netherlands*

Currently, scripting is a big issue – JS/Kak has featured high in the *VB* Prevalence Table since April 2000 and VBS/LoveLetter caused recent global havoc. VBScripting (VBS) is a subset of the *Microsoft* Visual Basic programming language. It was intended to control the interaction of ActiveX controls with the user. Some important files for scripting include VBSCRIPT.DLL, SCRRUN.DLL, JSCRIPT.DLL, WSCRIPT.EXE and CSCRIPT.EXE.

This year many systems will move to *Win2K*, *Office2K*, and *Internet Explorer* (*IE*) *v5* and the resulting VBScript level will be raised from v3 (and in some cases 4) to v5. This version allows access to the local files on a user's machine through the support of FileSystemObject (FSO) – a potential danger zone for the regular user.

### Outlook 97 and Forms

*Outlook 97* supports the use of Forms which can have VBScript code embedded inside them. 11 events are supported and are therefore vulnerable to exploits. However, in order to create/use/deploy Forms with VBScript code embedded, one needs certain rights on the *Exchange* server which are not set by default. Usually, when the user receives a Form with embedded VBScript, a warning message appears with an option to disable the 'macros'.

Also, Forms with VBScript code inside will probably be restricted to replicating within companies. Such Forms can be sent outside but whether the gateways will transfer the code correctly remains to be seen. Testing shows that the VBScript code embedded in Forms does not travel well.

### Outlook 98 and HTML

*Outlook 98* supported a new file format which made it possible to send emails in HTML. Web pages written in HTML can have events as well, for example, the onload event can be triggered upon accessing a page. Users must perform this function manually, so it does not present a significant vulnerability unless they are forced by an automatic script to call a specific Web page or change the default homepage settings.

There is a danger that VBScript code embedded inside HTML emails may go unnoticed. Recently, a great number of .VBS file attachments have caused trouble. This usually requires two stages; the user needs to open up the email message and double-click on the file attachment. However, the two-step operation is not always needed. An HTML email may have embedded VBScript code inside.

While many users think it is still safe simply to open up emails, with default security settings malicious code could exploit some vulnerabilities and be running without them knowing it. One of the more familiar exploits is the scriptlet.typelib vulnerability – many VBS viruses (including BubbleBoy, see *VB*, December 1999, p.6) make use of it. JS/Kak embeds its code in HTML email and the average user has no idea of its malicious potential. Worse, the Preview Pane might already activate the embedded VBScript code without the user having opened the email.

### Countermeasures

Always use current anti-virus software and update it regularly. AV software usually has a combination of specific and generic drivers, as well as heuristics. The problem with the latter is that the generic/heuristics drivers can result in false IDs. For example, there is regular user code out there that makes use of Outlook.Application, CreateItem(0) which some automated systems use for process failure notification.

Also, anti-virus (client) scanners might not be able to catch email worms. It is certainly better (in corporate environments) to use email and gateway scanners as well. Deploying AV scanners – running with different security-level configurations – at various entry-points is very effective (see p.14 of this issue). For emergency outbreaks, it is usually also easier and faster to maintain/update scanners at gateways then at all clients. If there is an outbreak of a VBS mass-mailing virus, email/gateway filtering on the email Subject might work, but only if the subject header is constant. Recently, we saw how some viruses select a header from multiple stored entries. It is also possible to use variables like date and time which could render subject filtering useless.

Make use of filtering on attachments and blocking either selected or all attachments. Who needs to receive .EXE or .VBS files by email at work? Hardly anyone. For distributing packages, it is better just to point to the link than physically to send the package with the email. The drawback is that a file extension means very little – a file name can have any extension. Instead of relying on this, it would be better to have the OS check for the real file type, regardless of the extension. Icons can be deceiving.

Filtering on email content is possible, however this is not going to work very well as it is easily changeable, either manually or by code. It makes more sense to filter out embedded scripts. With JS/Kak there is no email file attachment, just script code embedded inside. Most users have no idea that malicious code is activating until it is too late. It would be a good idea always to have a setting to wipe embedded script code.

HTML emails cannot easily be blocked by the user. When using *Outlook* Forms with VBScript code inside, the user gets a warning message about the presence of scripting code. Users opening up email do not know if the email was sent in regular *Outlook* message format or in HTML. If it is the latter, there is always a chance the user might face malicious scripting code. So, I think there should also be a *Microsoft* warning box when trying to open up email messages in HTML.

Microsoft security patches fix some of the vulnerabilities. The patch addressing the scriptlet.typelib vulnerability changed the mistakenly set 'safe for scripting' control but that is for specific controls only (see *VB*, April 2000, p.5). This does not mean all other controls are always safe. In fact, there are more controls that can be abused, for example the ActiveMovie control.

The popular idea is to run with High *IE* security settings, but this is hardly convenient. Manually downloading a file from a Web site is prohibited. In theory, you should always run with High *IE* security settings and lower them to Medium temporarily when you wish to overrule them. This switching is annoying and for most people it is not common practice. It is easy to forget to set it back to High when, for example, a lengthy download completes.

In my opinion, the more secure way is to change the *IE* security settings manually, using custom level. Of course, other settings can be tuned as well, like Java settings, etc. It is a matter of trust. If people tuned their settings, embedded code like JS/Kak's would not go unnoticed – a warning box would appear to enable/disable the code. It is tricky as disabling the code does not actually remove it, it just stops it from running. If, however, you reply/forward such an email, the malicious code is still there and travels with it.

Most people do not know what to change specifically – they might overlook something and think they are safe, giving a false sense of trust. What is more, this practice is hard to administer in corporations as not everybody is doing the same job and might have different requirements.

Many regular Web sites use scripting code to enhance their site. If you run in custom level and 'paranoid' mode, you get lots of message boxes asking whether to enable/disable the code when you set it to notify you. It is frustrating not being able to browse a few pages without hitting the OK button on the warning page. So people probably will go back to Medium custom *IE* security settings soon enough. It is actually recommended to set it to 'disable' instead of 'prompt'. Some Web sites simply do not work well with High security settings blocking ActiveX controls. Some do not show specifically blocked components, but there are those which do not work well at all – what you want to see is blocked, by mistake of course, but it happens sometimes.

There is a new *Outlook* patch (another one!) which blocks attachments like .EXE. While this might work well, it is not a convenient solution. It remains to be seen how many people want to use it. The idea of a pop-up box upon sending an email to multiple people is a nice thought but many people use address books with contacts etc and may get used to the message and not always pay close attention once the real warning is triggering.

This patch only covers *Outlook*, not *Outlook Express*. If I rename a .EXE file .123 and attach it, the new *Outlook* patch will probably not block it. It only takes a small script to rename it and voilà. What if there does not seem to be a file attachment at all? What if I take an embedded script with binary code for a .EXE inside? If I start a debug script it could go anywhere unnoticed on many systems. Who uses security patches anyway? First of all, people must know there *is* a patch, realize what it is for and actually spend time getting it downloaded and installed. This poses a problem if you have many mobile workers in your firm.

Does education work? I think it works partially, but for a short time only. The fact that VBS/LoveLetter had such a big impact and got so much media attention meant people knew at least something about it. The fact that system administrators might have set up one or more of the above guidelines was, in my opinion, one of the reasons that VBS/NewLove did not become such a huge issue. However, people have a tendency to forget, and think 'it won't happen to me'. Sadly, absolute faith in AV scanners is not a guarantee – a bit of user suspicion never hurts.

**New Vulnerabilities**

One of the most remarkable of LoveLetter's side-effects is that apparently it can send faxes if, as it goes over all the entries in all the address books, it encounters a fax server on the *Exchange* server. Another story involved someone getting a LoveLetter message on his pager. The recently discovered VBS/Timofonica could send a notification message to a telephone equipped with email with a randomly generated email address.

WAP phones might become vulnerable too. Right now, they are not very common. The question is whether WAP phones will become standard as, apart from the speed issue, they require rebuilding company info/services into another format – WML. [*A paper on mobile phones and viruses will be presented at VB2000. Ed.*] It could become even worse with handheld organizers, *Windows CE* and InfraRed ports (see p.12 of this issue). It is not really a question of replicating code yet, but problems could certainly arise.

*Outlook 98* supports the use of SRC, with which you can embed files such as a real .JPG file. So far, this has not been abused. The recently encountered, so-called CHM (compiled HTML file) vulnerability, also known as 'stealth-bomb', makes use of the ActiveMovie signed control exploit. Upon viewing an HTML page/email files can be copied to the system and executed. This is currently limited to *Windows 9x*. Thus, scenarios involving a combination of a script file (for mass-mailing) and binary file(s) (a back-door component for example) could be lethal in the future.

# BOOK REVIEW

## Bookworms – and Viruses

*David Harley*
*Imperial Cancer Research Fund, UK*

**Virus Proof – the Ultimate Guide to Protecting Your PC**
**Phil Schmauder**
Prima Tech/Jamsa Media Group
US:$34.99/Canada:$48.95/UK:£32.49

Books containing computer virus information tend to fall into four main classes:

- More-or-less competent but out-dated books primarily on viruses, by authors who were active in the field at the time of publication. Sadly, most of these pre-date the rise of the *Office* macro virus, VBS viruses, and the other changes in virus and anti-virus technology.
- Books on security in general written by security experts. Since some security experts are more virus-literate than others, such books vary in value.
- Books by authors who are not known anti-virus experts. These have been fairly rare, but there seems to be an incoming wave.
- Books on security written by individuals who are not necessarily recognised security experts. Characteristically, these have titles like 'The Gullible Book-Buyer's Guide to Internet Security' and incorporate lots of jokes, tip boxes and screenshots. Some of these, to be fair, are more informative than you might expect.

Despite its title, Phil Schmauder's book fits somewhere between the last two categories. While more virus-oriented than most dumbed-down security books (which typically devote a single chapter to malware), this one spreads its material rather thin and includes several sections whose relationship to the virus problem is unclear. Apparently, this book is intended for 'All Users'. However, it assumes a very low level of computer literacy. It uses a standard educational format – sections are described as lessons, not chapters, and follow a plan: introductory paragraph; summary of key concepts to be discussed; main discussion; 'What You Must Know' summary of key points (these, generally, simply reiterate the previous summary); and a couple of pages of uncommented, random screenshots.

I suggest that the author knows little of virus/anti-virus technology or culture. He does not distinguish between hackers and virus writers, and believes that macro viruses are intrinsically difficult to detect because they are embedded in documents. He is aware that virus programs 'sometimes' attach themselves to other programs, but appears to believe that 'typically' they disguise themselves as other programs such as *Excel*, and that macro viruses are primarily spread by hackers who mail them to their victims. He defines a virus not by its ability to replicate, but by its presumed intent to cause damage or steal data. He claims that viruses that attach themselves to other programs are called Trojan Horse viruses, and that AV signature scanning works by looking for text messages within viral code. Polymorphic viruses, he says, hide from scanners by changing the text message they display each time they are executed. There are very few references to further sources of information, so trusting readers have to look hard to find antidotes to this (sometimes dangerous) misinformation.

Some of the advice given on reducing virus risks is passable, if unremarkable (do not run unsafe code, use AV programs, be cautious with email, use safe browser settings). The section on backup, though technically uninformed, at least emphasises a real need. However, there appears to be no attempt to discuss formal backup strategies, or to address virus-specific backup problems (with long-term gradual corruption, for example). The author's understanding of encryption, firewalls, phreaking, Denial of Service attacks, hoaxes and spam ranges from superficial to laughable in my opinion, and the four 'case studies' of high profile threats are, to my mind, incompetent. The appendix describes *VirusScan* (a 30-day trial CD v4.0.3 is included), and is as near as the book gets to describing AV software.

Clearly, this book is not intended for the virus expert, or for anyone whose interest in the field justifies the cost of a subscription to *Virus Bulletin*. Even if the information it contained were of a higher quality, it would lack the detail needed for a computer science course-book at the most elementary level. Sadly, I think the same amount of misinformation could occupy a quarter of the space, if the almost entirely irrelevant source code, screen-shots, uninformative charts and repetitions were removed.

I suspect the code examples included are, like the redundant Y2K chapter, there to bulk out the content, and do not constitute much of a threat. Nevertheless, they focus on breaking things, not defending them. This might matter less if there were any attempt to place the virus problem into a legal and ethical context.

IT professionals in search of technical insight or managers hoping to improve their understanding of how to manage the malware problem will be sadly disappointed. This is targeted at the inexperienced newcomer to computing and the Internet, alarmed by the sensationalist media stories about Melissa and the so-called LoveBug. The 'average' reader will acquire more misconceptions than useful information from this book, and would be better off reading some FAQs, or even investing in Robert Slade's ageing but largely sound *Guide to Computer Viruses*.

# OPINION

## What the Headlines Don't Say: Malware Marches On

*Christine M Orshesky*
*i-secure Corporation, USA*

From the media coverage, whether hyped or not, it is clear that few organizations or indeed individuals escaped the direct effects of many of the recent malware incidents. Again, a malicious code attack occurred and LoveLetter, its numerous variants, and the many newer worms and viruses wreaked havoc with cost estimates reportedly running into the billions. There were, however, many indirect affects too and it seems prudent to reflect on what the headlines did not say and how all of us were and will be affected, whether infected or not.

Given the costs associated with LoveLetter and other malware-related incidents, there is an obvious financial affect, including recovery time and loss of data. The harder costs to articulate are the lost business opportunities and lost productivity. Denial of service was one of the biggest effects of many of the recent malware incidents.

This may seem rather obvious since many organizations shut down their Internet and email access during such incidents as a way to prevent being infected. But the effects were far more devastating due to our increasing reliance on Internet and email access. Businesses could not communicate with their customers or, in many cases, they could not perform their services or produce their goods. In addition, advisory messages were not received and warnings could not be disseminated widely and quickly. Untold numbers of commercial organizations were literally paralysed by the lack of connectivity.

The social effect was reflected in an increase in our paranoia about the things we receive in email and from whom we receive them. Those who have worked in and around information security will be the first ones to say that some paranoia is healthy. There is a point, however, where it becomes paralysing.

The newer malware attacks arrive in your mailbox primarily *because* they enticed some other unsuspecting recipient's curiosity. Once opened, the malware sends itself to everyone in the recipient's address book. As we all know, these address books are very large and include those inside and outside our organizations or circle of 'friends' as we fulfil our desire to stay in touch, seemingly with everyone.

When it arrived in your mailbox, the sender was most likely someone you knew, maybe someone you do business with, someone who regularly sends you jokes, or even someone you normally trust – making it difficult to know whether the message and its attachment could be trusted as well. Our increasing dependence on email as our primary source of communication could become crippling if such attacks continue to increase.

Given the number of systems that were infected by such things as LoveLetter around the world, surmising how many systems are and will continue to be vulnerable to similar attacks is not hard to estimate. There are ways to prevent such attacks from taking hold but it can be time-consuming to implement them on a widespread basis.

There are still many systems that become infected with detectable or preventable malware and continue to spread these infections. So unless there is a concerted effort to close some of the longstanding holes in our systems, attacks such as this will repeatedly find vulnerable systems to exploit and continue to plague the on-line community.

The biggest and possibly most subtle effect of the evolving malware attacks is the dissemination of information from your systems to remote sites. Not only do worms and other malware attacks send email with potentially sensitive data or documents (e.g. Melissa) to all of your friends, acquaintances, and even those you hardly know – which announces to the world your infection – they also allow potentially sensitive information to be inferred about your system. This makes you a potential target for future attacks.

In essence, viruses (e.g. Marker) and email worms (e.g. LoveLetter) have caused your system to knock on the door of a remote site asking for permission to enter. With some simple and widely available tools, it is easy for the remote system administrator (or others) to determine who was knocking, including such details as your identity and whereabouts. Over time, and given enough systems knocking at the door, it is possible to develop a fairly accurate picture, thereby telling the remote system that you are probably vulnerable to other attacks.

Unfortunately, the technology advancements and the corresponding, ever-present threats to your information have placed a tremendous burden on every member of the on-line community to stay aware of the threats and the ways to protect against them. Unless you were asleep during May, no one escaped the media coverage of LoveL etter, whether on your local television news, on-line news service, or newspaper. It is this awareness, one of the more positive affects of this incident, which will help prevent future malware epidemics.

History tends to repeat itself. There are new malware attacks evolving daily. Will we listen to the lessons learned from this and past incidents to prevent future ones? Or will the on-line community continue to be plagued by attacks that affect our social and financial structure?

FEATURE

# Palm Breach

*Eric Chien*
*Symantec, Netherlands*

In the 1980s, no one left home without their *Filofax*. Today, no one leaves home without their Personal Digital Assistant (PDA). However, while *Filofaxes* contained important names and numbers, PDAs are more than just an address book. Combined with Internet access, the functionality of the PDA is moving towards a desktop computer combined with a cellular phone, all small enough to put in your pocket.

This article will touch on malicious threats to the *Palm* Personal Digital Assistant as a preview to a presentation at the *Virus Bulletin* conference in Florida in September. The VB2000 presentation will not only provide a more in-depth look at the *Palm*, but also *EPOC32* and *Windows CE* devices. In addition, potential solutions to PDA threats will be presented including demonstrations of prototype applications in detecting malicious PDA code both on the PDA and associated devices.

The leading platform for handheld computing devices is *Palm* operating system. According to *IDC*, *Palm* OS controlled 78.4% of the handheld market share in 1999. Overall, *IDC* expects Personal Digital Assistants to exceed 18.9 million units by 2003. With more than 4,000 applications for the *Palm* OS, devices running *Palm* OS are at the greatest risk of malicious code.

*Palm* OS does not use a traditional file system. The file system is optimized for synchronization with a primary device (the desktop computer) and for the limited storage area available. Data is stored in memory blocks called records. Related records are grouped in databases where every record belongs to one and only one database. For example, a database may be a collection of all address book entries or all calendar entries.

A database is analogous to a file. The difference is that data is broken down into multiple records instead of being stored in one contiguous chunk. When modifying a database, the changes only take place in memory, unlike the traditional desktop method of temporarily storing it in RAM and then writing it out to storage. Such memory storage provides a home for new application databases (executable code), which can be introduced in a variety of different ways.

### Vectors of Delivery

Any method that allows the introduction of executable code onto the *Palm* device represents a vector of delivering potentially malicious code. While there are many methods of introducing code, 'HotSyncing' currently represents the primary method and, in the future, Internet access will actually pose the greatest threat. In the following paragraphs a brief description of three potential vectors of delivery is presented.

**HotSync:** The primary method by which applications are transferred onto the *Palm* is via the HotSync functionality. This is used primarily to synchronize data stored on the device with data stored on the desktop computer, back up data to the desktop computer, and install new applications to the *Palm* from the desktop computer.

Currently, this provides the easiest means of introducing malicious code. For example, to install a new program on the *Palm*, the user may download the new program from the Internet and save it to a desktop computer. Then, using the HotSync functionality, the program is transferred from the desktop computer to the *Palm*. Now saved to the *Palm*, the user can run the new program, which could be anything from a new chess game to a malicious program that emails out all your contact records.

**IrDA:** The *Palm* contains IR (InfraRed) communication capabilities. Such capabilities are compliant with IrDA (Infrared Data Association) specifications. Thus, the user can directly interface with the IR capabilities of the *Palm*. However, the majority of programs utilize the *Exchange Manager*. The *Palm Exchange Manager* provides a simple interface for *Palm* OS applications to send and receive data from a remote device using standard protocols. With IR capabilities, the *Palm* is able to receive and send applications and thus, potentially malicious code. Currently, devices are designed to trigger an incoming data alert message. However, this message can be disabled. This requires specific agent code on the receiving device. Via IR, malicious programs could potentially speak to other infected devices exchanging information and code all unbeknownst to the users.

**Network Access:** By adding optional modem hardware to the *Palm* or utilizing newer wireless models, one has access to many standard Internet protocols. In general, clipped Web browsing is available and so is email access with attachments. The user can easily receive emails with *Palm* applications attached, save those attachments, and execute them. Such applications could contain malicious code. Also, the net library allows *Palm* OS applications easily to establish a connection with any other machine on the Internet and transfer data to and from that machine using the standard TCP/IP protocols. Thus, malicious code is not limited to utilizing the *Palm* mail client or Web browser, but can open listening server ports allowing remote access, sending of confidential data, or receiving additional malicious code. Such network access is an open invitation to fast spreading worms.

While the vectors of delivery provide the doors to enter the *Palm* device, it follows that architectural design provides the keys for opening or exploiting those doors.

### Programmability

Many of the applications which run on *Palm* OS are programmable. A third party program can interact with the other programs through a standard application-programming interface. Specifically, applications can send launch codes to each other. Using these launch codes an application can direct another application to perform some action or modify its data.

For example, a malicious program could send a launch code to query all the email addresses in the Address List application. Then, the same program could send a launch code instructing the email application to queue and send email messages with itself as an attachment. All of this functionality can be performed without user input, and without the user's knowledge.

Such programmability easily allows for email type worms like W97M/Melissa and VBS/LoveLetter. How far and how fast such threats may spread is discussed later.

### File System

The file access functions in *Palm* OS allow the user to read, write, seek, truncate, and do everything else you would expect to do with a desktop-style file. Such functionality is all that is needed for a viral threat to spread. Viral threats may find other application databases on the device and append themselves to those application databases, changing the entry point of the program thereby ensuring future execution and continued replication.

The *Palm* does not employ any inherent access control to databases and records. System application databases are easily modified as regular user applications. This allows malicious code not only to modify system files, but also to destroy system files. With a single click, one could wipe out all the applications and data on the device.

### Libraries

The *Palm* OS is distributed with many libraries including the net library allowing *Palm* OS applications to establish a connection easily with any other machine on the Internet, and the IR (InfraRed) library allowing a direct interface to the IR communications. Such libraries make programming high-level threats very easy.

Without low-level knowledge of IR communications, a user could easily create an agent that monitors incoming IR data requests. By monitoring incoming IR requests rogue executables could communicate with other infected devices. Also, the net library allows programmers to create Berkeley sockets-style network programs. Programs like these could range from a small SMTP engine, creating email capabilities on a device that may not even have a mail client, to a server listening for incoming commands allowing hackers remote access.

### Spreadability

While the creation of viruses, worms, and Trojans are all possible for the *Palm* OS, their potential in-the-wild spread is influenced by a variety of factors. It would not be surprising if a malicious threat is discovered tomorrow; however, it would be surprising if such a threat posed an immediate widespread threat.

Firstly, while *Palm* holds the largest market share of Personal Digital Assistant users, the number of PDA users is magnitudes lower than the number of PC users. In addition, at this moment the number of network connected PDA users is also orders of magnitude lower than the number of people with access to the Internet. Thus, a malicious *Palm* OS application would not spread nearly as fast as, for example, a *Windows* worm.

Secondly, the model of data exchange for PDAs is still asymmetrical. Users still download applications and data from a few primary sources rather than a situation where many PDA users exchange information with many other PDA users. This symmetrical nature of code exchange can dramatically increase the threat of viral spread as demonstrated with macro viruses.

As the cost of PDAs continues to decrease and they become standard productivity devices issued in the corporate space, the threat increases dramatically. If we reach a day where we check email via our *Palm* and trade documents and other executable attachments via our *Palm*, the chances of malicious code being inadvertently executed will rise. In addition to this, if the marketplace consolidates to a single vendor, the susceptibility of the average PDA user will rise. Once such executable code is run, the possibilities are limitless. *Palm* devices as discussed are open for infection and can aid email worms by their robust programmability.

### Summary

*Palm* is only one of many vulnerable devices. Unfortunately, there is not a digital device that is 100% secure. To be 100% secure, one should revert to the old *Filofax*.

However, on the bright side, while there is a threat there are also potential solutions. Those who are interested in further details regarding threats to PDAs or corporations which are beginning to consider a PDA as a standard productivity device are encouraged to attend the *Virus Bulletin* conference this September.

In Florida, I will demonstrate such malicious programs and also prototype solutions, which can detect and block such threats. I will explore in more technical detail the functions that allow threats to be created and some simple steps that can be taken today to reduce one's susceptibility.

# TUTORIAL

## Safe Hex in the 21st Century: Part 2

*Martin Overton*
*ChekWARE, UK*

How do you get a 100% secure and virus-free system and keep it that way? Unplug the computer, take it outside, place it on the ground and drive over it a number of times with a large steamroller. Once its nice and flat, liberally splash with petrol and flambé for a number of hours. Once extinguished, place the contents into a safe deposit box. *Voilà*, a secure and virus-free system that will stay that way. In other words, there is no such thing as a fully secure system that is usable (in the normal sense) that cannot be infected/affected by malware. Got it? Good! Please pass the message on.

If we don't break out of the 'virus-scanner-is-king' mind set that many have fallen into then we will be doomed to keep repeating the same mistakes forever. Virus scanners have their place *but* (and it's a big one) virus scanners are no longer enough. We (the consumers) have taken the evolutionary short-cut to virus protection when what we really should have done was take the longer, twisting path towards proper security and system integrity which would have minimised many of the current threats.

There follow some suggestions for dealing with the current malware problems:

- Do not open attachments coming from unknown sources. Delete them.
- Before opening a file apparently coming from someone you know, if possible ask the sender if they actually meant to send it. If not, delete it and tell the sender that they may have a virus. If yes, scan it before opening it.
- Disable script (and HTML) support for mail and news.
- Disable Java and JavaScript in your browser and enable them only when required.
- Uninstall the *Windows Scripting Host* (*WSH*) if you don't need it.
- Set the boot sequence to C, A in the BIOS.
- Change *Explorer* to show all file extensions.
- Make backups of your data regularly.
- Encourage the use of safe file formats for data exchange (such as PDF).
- Encourage the use of URLs in emails rather than attachments.
- Set up a good solid 'Acceptable Use Policy' for email and Web use, and get your staff to sign it.
- Teach 'Safe Hex' to your support and other technical staff. For the non-technical staff, make sure that any policies are written in a suitable style so that there is no room for misinterpretation.

Should companies re-evaluate how email attachments are dealt with and what types are acceptable and safe? Yes! To paraphrase that great philosopher Forrest Gump, 'Email is like a box of chocolates, you never know what you're gonna get.' If you and your users have not already developed a healthily paranoid attitude (assume that all attachments/emails are suspect and may contain a virus), then you/they will continue to be victims. What can you do to minimise these risks?

### To Scan or Not to Scan?

Scanning may no longer be enough, but it *is* still needed so let's look at the best way of using this tool.

I am somewhat uneasy about using ISP scanning within a corporate scenario for the following reasons:

1. Privacy – if they are scanning your email they may also be reading it, especially if it is flagged as suspicious.
2. Encryption – this will walk right through ISP scanning (unscanned).
3. Liability – if the ISP informs a sender and the intended recipient that he has sent a virus, could this cause loss of confidence? What if it is a false positive or worse a false negative and the recipient gets infected, who's to blame, where's the legal recourse?
4. Over-reliance – I can hear board managers all over the world stating that they no longer need to protect their workstations, file/print servers or scan email ever again as their ISP will do it for them and think of all the money they'll save for their company.

Gateway (SMTP) scanning is a definite must as it can make the biggest impact on virus penetration within your company. There are a few caveats. It is best to use a different virus-scanning vendor from the one you use at the desktop or file/print servers. Encrypt/decrypt at the gateway so that viruses cannot sneak through, or block/quarantine all encrypted email, password-protected ZIPs (and other compression formats).

*Lotus Notes* or *Exchange* scanning is another very important scanning point as these servers can act as 'viral foxholes' allowing malware a safe haven from where they can strike out again. Once more, I would strongly suggest using

a different scanning engine from those on the desktop and file/print servers. You may even want to use a different scanning engine from the one on your SMTP email gateway. In security, diversity of products/technologies is in itself a protection mechanism and should be encouraged at every (practical) opportunity.

Content scanning (and lexical analysis) is fast becoming an extremely powerful function for limiting and immediately blocking new threats, be they infected attachments, documents, embedded scripts, as well as pornography and other unacceptable content. Why? Simply because you decide what is acceptable and allowed and are always in control. If you want to you can block all scripting languages' executable attachments, *Microsoft Office* files, etc.

### VBA and Macros

VBA is the successor to the separate (but similar) *Office* product-specific Macro languages. Is *Office* now a mini-operating system itself, or is it just that *Office's* tentacles infiltrate the underlying 'real' OS? I don't know and I really don't care, except where this impinges on my (and others) security and productivity. Let's call it a mini-operating system and look at the problems this brings.

Macros are the biggest threat to most companies, as documents (and other *Office* files) are passed around with wild abandon. This is compounded by the number of people using *Word* as their default email editor. What can you do to minimise this threat?

Stop using DOCs. Use pure Rich Text Format for your Word documents. Some macro viruses intercept File SaveAs RTF and save a file with a .RTF extension which actually contains a DOC format file! So it needs to be true Rich Text Format. Also RTF files can contain OLE components that in themselves could be a threat. Use Adobe Acrobat (PDF format) as this is currently not known to be capable of carrying a virus. Tell people that you would rather they sent you CSV files than XLS. Finally, use the in-built protection in *Word, Excel* and other *Office* products.

I predicted (at VB'99) that *Visio* would soon be targeted by virus writers as it uses VBA, so it was no great surprise when it came to pass just weeks before *Microsoft* took ownership of the company. What can you do? Add *Visio* files to the list of formats that you might consider filtering at the SMTP gateway and ensure that your virus scanners can detect viruses in *Visio* files.

### Getting Your Backup

Regular backups of data on your system are still very important. You can replace program files easily enough from master disks, but corporate data is worth a lot more to your company and is hard to replace if damaged or destroyed. In many firms data is the very lifeblood of the company. So, don't bleed to death, back up that data before it's damaged or destroyed by a virus or other malware.

### Make a WSH

A new class of virus now uses what is effectively Visual Basic Scripting language. This scripting language can be used to perform any task and use any application it can access. This has already been used by a number of new VBS/*WSH*-based script viruses/worms/Trojans.

Allowing this support to be turned off (as required) can effectively render this new threat dead in the water. The latest virus to take advantage of *Windows Scripting Host* was the recent and infamous ILOVEYOU virus aka VBS/LoveLetter.A aka LoveBug. It is highly advisable to turn off *Windows Scripting Host* if you do not need it. At the very least block all scripting languages coming in to your company in email.

How do you tell if you have got *WSH* installed? The simplest way is to search the hard disk for WSCRIPT.EXE. If it exists (usually in the *Windows* SYSTEM or SYSTEM32 directories) then it almost certainly is installed.

*Windows Scripting Host* is installed by default in *Windows 98, NT 2000* and on any version of *Windows 95* and *NT* when *Internet Explorer 5.x* is installed. Be aware though that it can also be installed as a separate entity on systems that do not have *IE 5.x* (such as *Windows 95* or *NT 4*). While I'm covering scripting, I would strongly suggest that if you are using *Outlook* or *Outlook Express* you ensure that your standard build has HTML format for email and news disabled, as this in itself will help to slow down (or stop) some malware that uses embedded scripts.

### Trust me, I'm … Signed

This is a technology that has been touted as the solution to many of the recent problems with malware. This has its place as part of a multi-layered approach to malware protection, but it is not a total solution as it is fatally flawed in one respect, you have to trust the signer absolutely! Does the following sound familiar? Software vendors assure you that their code/control/application is safe until a bug is exploited by a piece of malware and you have to download the patch. This is what happened with the Kak worm. The counsel for the prosecution rests its case.

### Recipe for Success

Use scanners wisely and update them regularly. Deploy a diversity of products at different points. Encourage the use of safe alternatives. Instil a healthy paranoia into your staff and back it up with solid policies. Train your technical and support staff. If you do not use scripting languages in your company, disable them before someone disables/damages your systems. Set up an AV section on your Intranet and point your staff to that. Monitor vendor and security sites/mailing lists and ensure you patch your systems and applications when new vulnerabilities are found. Change the boot sequence on your systems, and finally, make regular backups of at least your data. Safe computing!

# COMPARATIVE REVIEW

## In it to Win 98 it!

*Matt Ham*

This month's review is something of an oddity in recent years, being the first where two reviewers have worked together on a Comparative since the FitzGeraldian era. It is also the first time that this writer has reported upon a Comparative Review since those long gone days, and thus the slow trickle of changes observed by the outgoing writer have become a great avalanche for the returning one. Whether these changes are for better or for worse, or in some cases have occurred at all, varies with the product, and of course there are some new faces available for the delight and delectation of our avid readers.

The number of products submitted for review has increased since the last *Windows 98* Comparative (see *VB*, November 1999, p.16) – sixteen were submitted then, eighteen now, two extra sheep having wandered hopefully into the *VB* fold. Delayed by the lack of a March 2000 WildList, and the subsequent late announcement of the April list, this Comparative sees the resumption of scheduled *VB* testing. So, are there any wolves in sheep's clothing, or is it all mutton dressed as lamb this month?

### Test Procedures

The customary *VB* test-sets were used for testing, the ItW set aligned to the April 2000 WildList, which was announced on 25 April. Accordingly, products were submitted by a 26 April deadline. A variety of viruses were added to the test-sets, the most notable new entries being a selection of JS/Kak variants, the .A and .B variants of BAT/911, and samples infected with the polymorphic W97M/Service.A. Relevant to the ItW and Standard sets, was the (somewhat unexpected) addition of JS/Unicle, which proved to be something of a nemesis to all but the luckiest.

As ever, performance tests included the measuring of on-demand scan rates and on-access scanning overheads. The means by which such properties have been assessed have been described in previous Comparatives.

### Alwil AVAST32 v3.0.247 (25/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.7% | Macro | 97.1% |
| ItW Overall (o/d) | 99.7% | Standard | 97.8% |
| ItW Overall (o/a) | 98.9% | Polymorphic | 90.1% |

Starting with no surprises, *AVAST32* still required an altered version of the on-access test procedure – with deletes being applied to created/modified files for a copy run of the test-set. The product was noticeably sluggish but this was forgiveable when combined with good detection rates.

The bulkiest misses for *Alwil* occurred with the polymorphic macro viruses – W97M/Service.A and the .E and .F variants of W97M/AntiSocial accounting for over half of all the product's misses on-demand. There also seems to be something of a blindspot at the other end of the complexity scale with 32 misses on members of the W97M/Minimal family newly introduced to the Macro set.

Other than macro woes, the newcomers of JS/Unicle.A and BAT/911.A and .B were also undetected. Concerning both BAT/911 and JS/Unicle a brief discussion can be found in the conclusion, since both bring up interesting points. Missing JS/Unicle cost *AVAST32* a VB 100% award from an on-demand viewpoint, though a smattering of wild non-macros missed on-access provided something of a contrast. *Alwil* is seemingly concentrating its efforts in macro viruses into those which are in the wild, while their non-macro problems are mainly due to differences between the on-access and on-demand components of the product.

### CA InoculateIT v4.53.524 (25/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.7% | Macro | 100.0% |
| ItW Overall (o/d) | 99.7% | Standard | 99.9% |
| ItW Overall (o/a) | 99.7% | Polymorphic | 97.8% |

*InoculateIT* showed a defiance of the usual status quo in this latest test by being relatively superior on-access. The results were, however, none too shoddy in either department. JS/Unicle was again a sticking point in both varieties of test, along with macro list entry W97M/Story.F.

Another WildList miss at first appeared for the *PowerPoint* incarnations of O97M/Tristate.C, this being one of those spotted on-demand but not on-access. The simplest explanation of this, that the on-access product is not checking all extensions scanned by the on-demand component, is clearly incorrect since similarly infected *PowerPoint* samples were detected successfully in the Macro test-set. Odd indeed, but not unexpected since the *InoculateIT* on-access scanner is, as traditionally has been the case, particularly unstable and, as here, not always totally effective.

### CA Vet Anti-Virus v10.1.8.6 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.7% | Macro | 99.6% |
| ItW Overall (o/d) | 99.7% | Standard | 99.2% |
| ItW Overall (o/a) | 99.7% | Polymorphic | 92.3% |

From vague memories of the past *InoculateIT* was generally a less likeable creature than *Vet Anti-Virus*. This led to some commentators being rather scathing about *CA's* choice of the *Vet* line to be the basis of their free offering to the world. On the basis of the detection rates shown in this

| On-demand tests | ItW Boot | | ItW File | | ItW Overall | Macro | | Polymorphic | | Standard | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Missed | % | Missed | % | % | Missed | % | Missed | % | Missed | % |
| **Alwil AVAST32** | 0 | 100.0% | 1 | 99.7% | 99.7% | 104 | 97.1% | 178 | 90.1% | 32 | 97.8% |
| **CA InoculateIT** | 0 | 100.0% | 1 | 99.7% | 99.7% | 0 | 100.0% | 17 | 97.8% | 1 | 99.9% |
| **CA Vet Anti-Virus** | 0 | 100.0% | 1 | 99.7% | 99.7% | 18 | 99.6% | 340 | 92.3% | 8 | 99.2% |
| **Command AntiVirus** | 0 | 100.0% | 0 | 100.0% | 100.0% | 0 | 100.0% | 1 | 99.9% | 9 | 99.1% |
| **DialogueScience DrWeb** | 0 | 100.0% | 2 | 99.4% | 99.4% | 8 | 99.7% | 100 | 97.3% | 9 | 99.2% |
| **Eset NOD32** | 0 | 100.0% | 0 | 100.0% | 100.0% | 8 | 99.7% | 100 | 97.3% | 7 | 99.1% |
| **F-Secure Anti-Virus** | 0 | 100.0% | 0 | 100.0% | 100.0% | 0 | 100.0% | n/t | n/t | 21 | 99.7% |
| **FRISK F-PROT** | 0 | 100.0% | 0 | 100.0% | 100.0% | 0 | 100.0% | 1 | 99.9% | 9 | 99.1% |
| **GeCAD RAV** | 0 | 100.0% | 1 | 99.7% | 99.7% | 40 | 98.9% | 17 | 97.8% | 21 | 98.1% |
| **Grisoft AVG** | 0 | 100.0% | 5 | 98.8% | 98.9% | 20 | 99.4% | 124 | 92.0% | 34 | 98.2% |
| **Kaspersky Lab AVP** | 0 | 100.0% | 1 | 99.7% | 99.7% | 8 | 99.7% | 0 | 100.0% | 5 | 99.8% |
| **NAI VirusScan** | 0 | 100.0% | 1 | 99.9% | 99.9% | 7 | 99.8% | 6 | 99.2% | 4 | 99.9% |
| **Norman Virus Control** | 0 | 100.0% | 1 | 99.7% | 99.7% | 4 | 99.8% | 286 | 91.2% | 1 | 99.9% |
| **Panda AntiVirus** | 0 | 100.0% | 24 | 97.1% | 97.2% | 44 | 98.9% | 1336 | 86.0% | 59 | 97.1% |
| **Softwin AVX** | 1 | 96.5% | 23 | 98.2% | 98.1% | 8 | 99.7% | 376 | 90.5% | 101 | 95.1% |
| **Sophos Anti-Virus** | 0 | 100.0% | 1 | 99.7% | 99.7% | 21 | 99.4% | 191 | 95.2% | 45 | 98.2% |
| **Symantec Norton AntiVirus** | 0 | 100.0% | 0 | 100.0% | 100.0% | 21 | 99.4% | 264 | 94.7% | 17 | 99.2% |
| **VirusBuster** | 1 | 96.5% | 122 | 85.9% | 86.3% | 264 | 93.9% | 2042 | 79.3% | 166 | 91.6% |

review, however, *CA* at first glance seemed to have made the correct choice as far as detection rates are concerned.

Admittedly, for in-the-wild scanning the usual suspect of JS/Unicle was the preventor of a VB 100% award for *Vet Anti-Virus*, and this was consistent, as in fact were all results, between the on-access and on-demand tests. Of the rest of the files, however, *Vet* missed a larger total number, which makes *InoculateIT* clearly better. Or not, since *Vet's* misses are almost all due to two polymorphic viruses, so actual viruses detected are comparable despite samples detected being fewer. This definitely shows the perils of using straight numbers as a guide to performance, and leaves the 'which is better' debate beween these two products as up in the air as ever. It also leaves a sense of relief that *VB* is not constrained to put a ranking on every product as is so common in general industry magazines.

One area where *Vet* has slipped is, however, scan rates. Once the speed merchant to beat in the throughput tests, *Vet Anti-Virus* now sits in the middle of the pack.

## Command AntiVirus v4.58.3 (23/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 100.0% | Macro | 100.0% |
| ItW Overall (o/d) | 100.0% | Standard | 99.1% |
| ItW Overall (o/a) | 100.0% | Polymorphic | 99.9% |

With a new nifty trick supplied to *Virus Bulletin* for disabling on-access messaging (via the Registry), the ease of testing of *Command's* offering was markedly up from past tribulations. Detection-wise too, all was sweetness and light, with *Command* being able to claim the first VB 100% award of this month's products. Of the small number of samples missed BAT/911 was one – though only in its .PIF portions. The slight vagaries of on-access versus on-demand were again apparent, with VBS/First.C showing as infected on-demand and not on-access.

As always with those products where detection is high and problems few, there is little to write but the pleasant and so we move quickly on in hope of features to criticise.

| On-access tests | ItW Boot | | ItW File | | ItW Overall | Macro | | Polymorphic | | Standard | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number | % | Number | % | % | Number | % | Number | % | Number | % |
| **Alwil AVAST32** | 0 | 100.0% | 4 | 98.9% | 98.9% | 104 | 97.1% | 178 | 90.1% | 55 | 96.4% |
| **CA InoculateIT** | 0 | 100.0% | 1 | 99.7% | 99.7% | 0 | 100.0% | 0 | 100.0% | 1 | 99.9% |
| **CA Vet Anti-Virus** | 0 | 100.0% | 1 | 99.7% | 99.7% | 18 | 99.6% | 340 | 92.3% | 8 | 99.2% |
| **Command AntiVirus** | 0 | 100.0% | 0 | 100.0% | 100.0% | 0 | 100.0% | 1 | 99.9% | 10 | 99.0% |
| **DialogueScience DrWeb** | 0 | 100.0% | 2 | 99.4% | 99.4% | 8 | 99.7% | 100 | 97.3% | 11 | 99.1% |
| **Eset NOD32** | 0 | 100.0% | 0 | 100.0% | 100.0% | 8 | 99.7% | 100 | 97.3% | 7 | 99.1% |
| **F-Secure Anti-Virus** | 0 | 100.0% | 0 | 100.0% | 100.0% | 0 | 100.0% | n/t | n/t | 21 | 99.7% |
| **FRISK F-PROT** | 1 | 96.5% | 0 | 100.0% | 99.8% | 0 | 100.0% | 1 | 99.9% | 31 | 97.8% |
| **GeCAD RAV** | n/t | n/t | 1 | 99.7% | n/a | 37 | 98.9% | 18 | 97.8% | 21 | 98.1% |
| **Grisoft AVG** | 1 | 96.5% | 6 | 98.7% | 98.6% | 23 | 99.3% | 292 | 89.4% | 51 | 96.6% |
| **Kaspersky Lab AVP** | 0 | 100.0% | 1 | 99.7% | 99.7% | 8 | 99.7% | 0 | 100.0% | 5 | 99.8% |
| **NAI VirusScan** | 0 | 100.0% | 1 | 99.9% | 99.9% | 7 | 99.8% | 698 | 95.6% | 6 | 99.7% |
| **Norman Virus Control** | 0 | 100.0% | 1 | 99.7% | 99.7% | 8 | 99.7% | 292 | 90.9% | 1 | 99.9% |
| **Panda AntiVirus** | 0 | 100.0% | 28 | 96.5% | 96.7% | 84 | 97.8% | 1336 | 86.0% | 87 | 95.9% |
| **Softwin AVX** | n/t | n/t | 23 | 98.2% | n/a | 8 | 99.7% | 374 | 90.6% | 101 | 95.1% |
| **Sophos Anti-Virus** | 0 | 100.0% | 1 | 99.7% | 99.7% | 25 | 99.2% | 191 | 95.2% | 45 | 98.2% |
| **Symantec Norton AntiVirus** | 0 | 100.0% | 0 | 100.0% | 100.0% | 21 | 99.4% | 264 | 94.7% | 17 | 99.2% |
| **VirusBuster** | 3 | 89.6% | 125 | 85.6% | 85.7% | 267 | 93.9% | 2042 | 79.3% | 167 | 91.4% |

## DialogueScience DrWeb v4.17 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.4% | Macro | 99.7% |
| ItW Overall (o/d) | 99.4% | Standard | 99.2% |
| ItW Overall (o/a) | 99.4% | Polymorphic | 97.3% |

Thankfully, *DrWeb* does not disappoint on the niggles front, though not through a lack of detection capability. As might be expected from previous entries JS/Unicle.A was not detected, which was enough to deny *DialogueScience* a VB 100% award this month. Elsewhere, misses included BAT/911 in its .PIF forms and W97M/Service.A. As far as differences between on-access and on-demand were concerned, a couple of extra file viruses slipped through on-access, with no readily discernable rhyme or reason.

Where *DialogueScience* can be heartily upbraided, however, is the matter of its glorious retro interface, which although no doubt fashionable in nightclubs is most unpopular with this reviewer. The on-access boot scans in particular were hampered by lack of configurability and a distinctly 16-bit ambience. This is likely the case behind the scenes too, as *DrWeb* is resource-hungry when performing on-access scans and dawdles in the scanning race.

## Eset NOD32 v1.35 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 100.0% | Macro | 99.7% |
| ItW Overall (o/d) | 100.0% | Standard | 99.1% |
| ItW Overall (o/a) | 100.0% | Polymorphic | 97.3% |

After that slight diversion into the land of the unusual back into the predictable, and another VB 100% award for *Eset*. *NOD32* does, to its credit, remain one of the more interestingly styled products on offer, as well one of the least amenable for witty comments at its expense. It has an excellent rate of scanning combined with accuracy, an enviable position to be in. W97M/Service.A proved a sticking point for detection, as did a smattering of assorted JS/Kak worm variants though none of those encountered in the WildList as used.

This leaves space to comment that one of the related files to JS/Unicle did slip through *Eset's* scanning, probably for the very good reason that *Eset* had not received it. Part of JS/Unicle's payload involved downloading this missed file from an ftp site (now closed). *VB* did not include this EXE as part of the WildList set since such downloads are open to change at the whim of the site owner.

Quite what anti-virus companies should do about such malware, where, in a twist of the usual Trojan behaviour, the name cannot change but the contents can, is left for the moment as an exercise for the enthusiastic reader.

## F-Secure Anti-Virus v5.10.6152 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 100.0% | Macro | 100.0% |
| ItW Overall (o/d) | 100.0% | Standard | 99.7% |
| ItW Overall (o/a) | 100.0% | Polymorphic | n/t |

Slightly more serious complaints may be levelled at *F-Secure Anti-Virus* (*FSAV*), though once more this is not primarily through a lack of detection ability. Detection was sufficient to find all in the wild specimens, with only a bunch of the usual suspects remaining undetected in the Standard test set. There may well have been some misses in the Polymorphic set too, but an executive decision was made not to bother doing these tests.

Before cries of anguish, wailing, gnashing of teeth and stern emails erupt, this was not simply a case of the tester deciding to skip a few days of work. In the long established tradition of log files proving to be the curse of Comparative testing, *FSAV* have added yet another unpleasantness, by providing log files in HTML format. These are vast, epic and sprawling affairs, sufficient to slow first to a crawl and next crash the test machines when scanning any decent sized collection, thus the Polymorphic sets were not testable in any convenient way.

Admittedly, the Polymorphic test-sets are a somewhat harsh test for any application where the log is constantly open, but whole new vistas of possible problems are unveiled with an HTML log. In the past, unique formats and .TXT files have been the norm, the files thus being uninfectable by any virus. Now the *F-Secure* team have introduced an infectable log. Infect this with script virus and lo and behold you could have infected log files. Does *FSAV* scan its own log files? Well, whether or not it does, the situation could be distinctly messy.

*FSAV* missed a VB 100% award due to a false positive, an act which might well be considered divine justice in response to the HTML logs.

## FRISK F-PROT v3.07b (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 100.0% | Macro | 100.0% |
| ItW Overall (o/d) | 100.0% | Standard | 99.1% |
| ItW Overall (o/a) | 99.8% | Polymorphic | 99.9% |

Back into the rant-free world and to earlier halcyon days. Unfortunately for *FRISK*, though, these days are so far in the past that they include the odd spectacle of missed in-the-wild boot viruses, in this case the decrepit Michelangelo (see *VB*, November 1999, p.20). Other misses were no surprise, though the ability to scan (on-access) the test-sets turned out to be another challenge by the software rather than the samples.

In parallel with the ability to miss boot viruses this product has also harked back to the days before networks and the real-time component had major problems with this new fangled connectivity. These problems resulted in blue-screen crashes until the scanning was performed locally, a 'feature' to bring pleasure to only the most masochistic. The version-specific bug (associated with mapped drives) proved an exception to the rule for the traditionally reliable Icelandic product.

## GeCAD RAV v7.6.360 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.7% | Macro | 98.9% |
| ItW Overall (o/d) | 99.7% | Standard | 98.1% |
| ItW Overall (o/a) | n/a | Polymorphic | 97.8% |

*GeCAD* continues in its attempts to gain the *VB* whimsy crown (*Eset* and *Alwil's* selection of beetles and geigeresque illustrations being the main competition) with a move away from their traditional shock tactics. The original operating theatre graphics have been replaced by those of a more relaxing domestic pet, though the setup has been altered in a most original way.

Upon first loading *RAV*, options are given for customisation. Colour scheme is selected first, then the rather more *outré* 'voice'. This gives a choice between graceful or macho, which brought visions of the computer declaring loudly 'I spit on your feeble infection attempt!'

As far as performance is concerned *RAV* missed the VB 100% award on the basis of the no doubt guessable JS/Unicle, though there were some scares since the 'files scanned' counter bore no resemblance to the number in fact processed. File scanning was otherwise not fraught with any great perils, boot sectors were another matter.

Rather than performing the trick of blue screens and floppy problems, *RAV* opts for a more refined approach to these glitches, namely by combining the two in one neat package. On-access boot scanning has never been a strong point of *RAV*, at least from the ease-of-use perspective, though until now it could at least be performed without blue screens – another victory for retro problems.

## Grisoft AVG v6.0.116 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 98.8% | Macro | 99.4% |
| ItW Overall (o/d) | 98.9% | Standard | 98.2% |
| ItW Overall (o/a) | 98.6% | Polymorphic | 92.0% |


Having taken over the mantle of mighty speed king, at least in the area of OLE files, *AVG* has changed much since its first arrival on the scene. ItW misses yet again included JS/Unicle, with Win32/Kriz providing the rest of on-demand misses. The less commonly encountered .OCX extension sample of Win32/Funlove added to the undetectables on-access. On-access boot scanning was once again a problem, though at least completed with no crashes. Michelangelo was again the culprit, which would no doubt leave its author amused if he were aware of this and if indeed he has not died due to advanced age.

The speediness of a product is often directly related to false positives and lack of detection, so these are areas of interest with *AVG*. Sure enough, both the main Clean set and the zipped executable set showed false positives. Detection, on the other hand, was not particularly bad, though undetected samples were something more of a mixed bag than with other products – only a distinct weakness with dedicated Win32 viruses being particularly notable.

## Kaspersky Lab AVP v3.0.132 (23/04/2000)

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

The inevitable JS/Unicle miss again prevented a VB 100% award for *AVP*, which does not exactly make for fascinating reading. Other misses were also nothing to write home about – the .PIF parts of BAT/911 and a triad of sundry macro viruses. Most notable for *VB* testers, though possibly less so for the rest of the known world, there is now an option to disable on-screen alerts during on-access scanning, which improved reviewers' quality of life greatly.

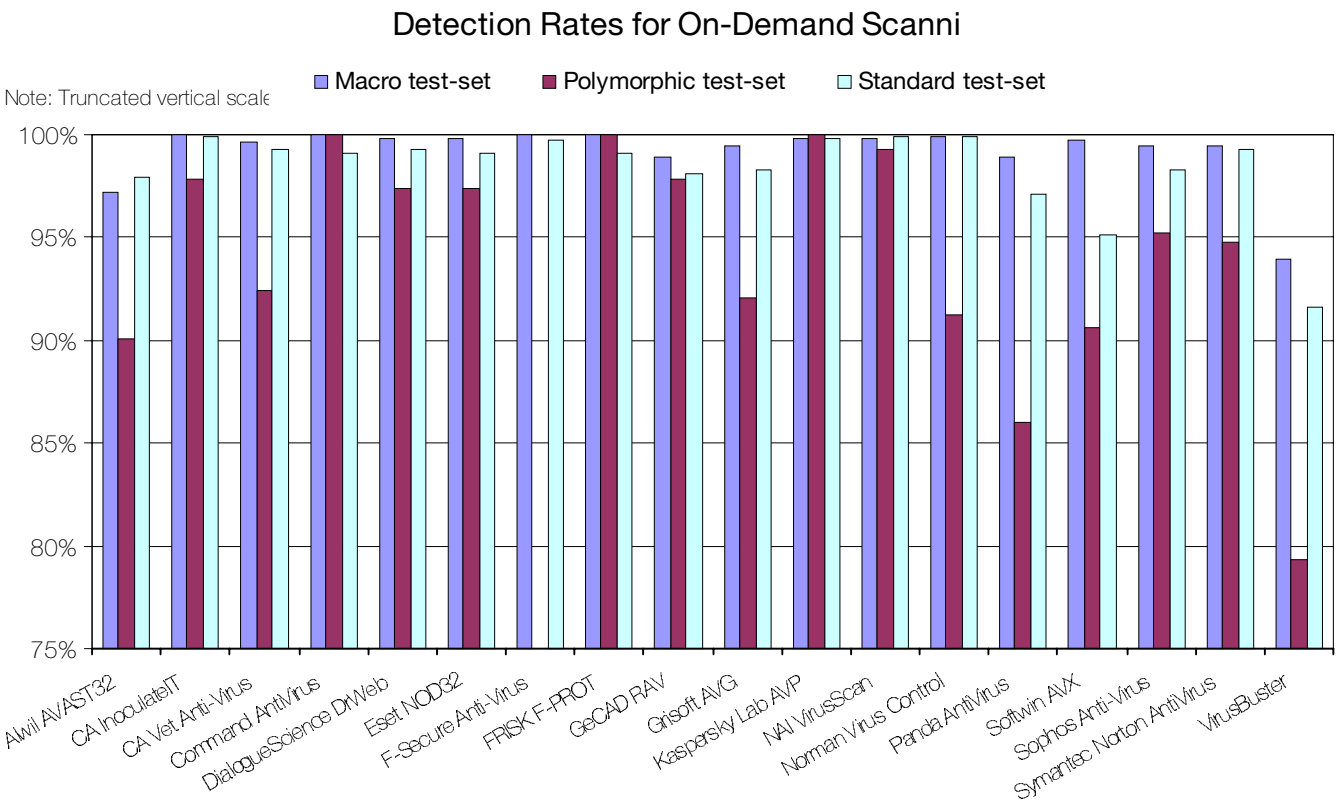
*AVP* now rests towards the slower end of the pack, but other than this there is little of evil repute to malign it with, and so on to the next victim.

## NAI VirusScan v4.5.0.4075 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.9% | Macro | 99.8% |
| ItW Overall (o/d) | 99.9% | Standard | 99.9% |
| ItW Overall (o/a) | 99.9% | Polymorphic | 99.2% |

Where you may be forgiven for guessing that only JS/Unicle prevented a VB 100% award, for once this would be erroneous. An extensionless O97M/Tristate.C sample was the bugbear for *VirusScan* on this occasion, a welcome breath of novelty in the testing procedure. Also of note was the wide disparity in polymorphic detection when operating on-demand and on-access. Russel.3072.A and SatanBug.5000.A proved easily, if slowly, detected by the on-demand scans, though patchily detected on-access. With such antiques, this is something of a surprise.

A new installation routine, in 50s style and demonstrating the less than purely corporate leanings of the *NAI Windows* product, led on to a not particularly revolutionary front end. Thus, no great new problems were to be expected and none were encountered.




## Norman Virus Control v4.80 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.7% | Macro | 99.8% |
| ItW Overall (o/d) | 99.7% | Standard | 99.9% |
| ItW Overall (o/a) | 99.7% | Polymorphic | 91.2% |

It has always been tricky to find exciting faults with *Norman's* scanners, and with JS/Unicle around to provide a topical reason this situation seems destined to continue. On this occasion, however, a small amount of excitement can be added in the form of two false positives in the Clean set. Not being a particularly fast or slow product there is, however, no great deal of discussion possible on the subject of this small failing.

As far as misses in the other test sets go ACG.A and .B plus Win95/Sk.8044 made up the majority, mostly by dint of being polymorphics and thus being scanned in large numbers. As far as other executables went, however, initial tests revealed a single executable infected with Vcomm.637 to be the only undetected non-polymorphic. Due to the suspicious nature of this observation, a subsequent retest was performed, which revealed the observation to indeed be bogus. Quite why this sample was missed initially remains a Comparative mystery.

## Panda AntiVirus v6.17.20 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 97.1% | Macro | 98.9% |
| ItW Overall (o/d) | 97.2% | Standard | 97.1% |
| ItW Overall (o/a) | 96.7% | Polymorphic | 86.0% |

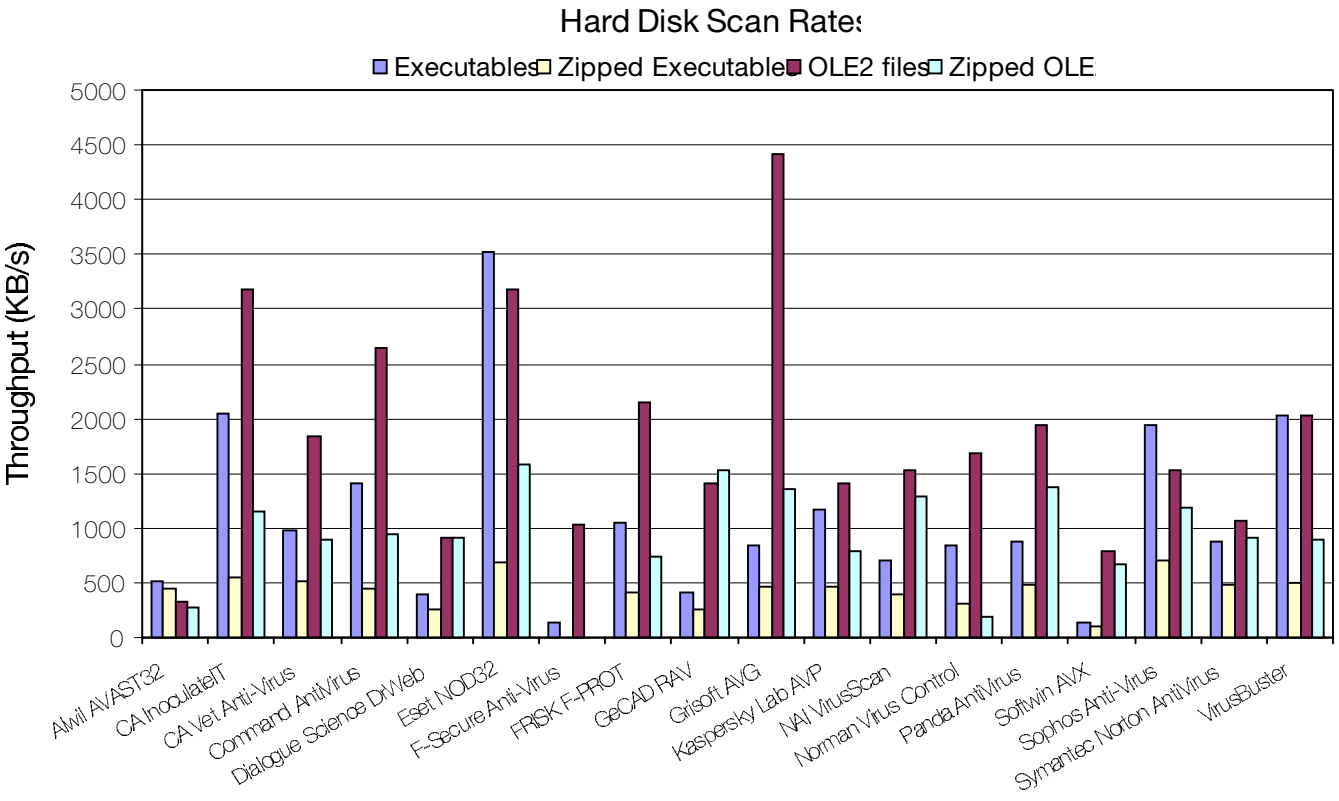
A product where niggles fight with good points in a deadlocked struggle, *Panda AntiVirus* (*PAV*) suffered a number of stability issues, and oddities in its reports. On-access scanning tests proved impossible without failure over a network, thus scanning was performed locally after several different configurations failed to fix the problem.

In common with the other product developers, *Panda* is keen to earn itself a VB 100% award. The feat was not achieved in this review due, quite simply, to a wholly inadequate default extension list. Sadly, a series of omissions from this list (somewhat unbelievably including the .SCR extension) caused *PAV* to miss a variety of files from the ItW set. No doubt the developers will be looking forward to the next Comparative, by which time the extension list will hopefully have been updated.

## Softwin AntiVirus eXpert v2000 (25/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 98.2% | Macro | 99.7% |
| ItW Overall (o/d) | 98.1% | Standard | 95.1% |
| ItW Overall (o/a) | n/a | Polymorphic | 90.5% |

The first of the two newer products, as far as *VB* is concerned at least, *Softwin* shared with its fellow newcomer a miss in the on-demand boot sector tests. On-access boot tests were another blast from the past since they were not present – a feature which will, we hope, be added as soon as possible. Results elsewhere, however, were promising, with only speed of processing being a particularly weak point. Presumably this will be slowed even further as extra




definitions are added, and it could be tricky to keep it within manageable margins. This is a point to follow in future appearances of *AntiVirus eXpert* (*AVX*) in *VB* Comparative reviews.

As well as the by now *passé* missing of JS/Unicle.A, TMC_Level-69 was also missed from the In the Wild set. Elsewhere a mixed selection of viral files passed through the detection net. Certainly a product which looks set to be among the top performers with a little improvement.

## Sophos Anti-Virus v3.33 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 99.7% | Macro | 99.4% |
| ItW Overall (o/d) | 99.7% | Standard | 98.2% |
| ItW Overall (o/a) | 99.7% | Polymorphic | 95.2% |

A not particularly happy outing for *Sophos Anti-Virus* (*SAV*) this time around, with numerous misses in areas where detection could have been simply obtained. The failure to detect all the JS/Unicle samples was added to by a lack of HTM scanning in this release which led to JS/Kak samples passing undetected through the test. In the Standard set, BAT/911's .PIF and .BAT components were also passed wholesale as non-viral. The HTM scanning has since been added as standard, but the lack in the intervening time can be considered rather inopportune.

This particular problem was perhaps less worrying than the missing of a selection of a few polymorphic virus samples within ACG.A and Win95/Sk.8044, since *SAV* has traditionally encountered few problems in the Polymorphic sets. One suspects that *Sophos* will be relieved that such a performance came at a time when few other VB 100% awards were received, and will be looking for a major improvement in the next *VB* Comparative.

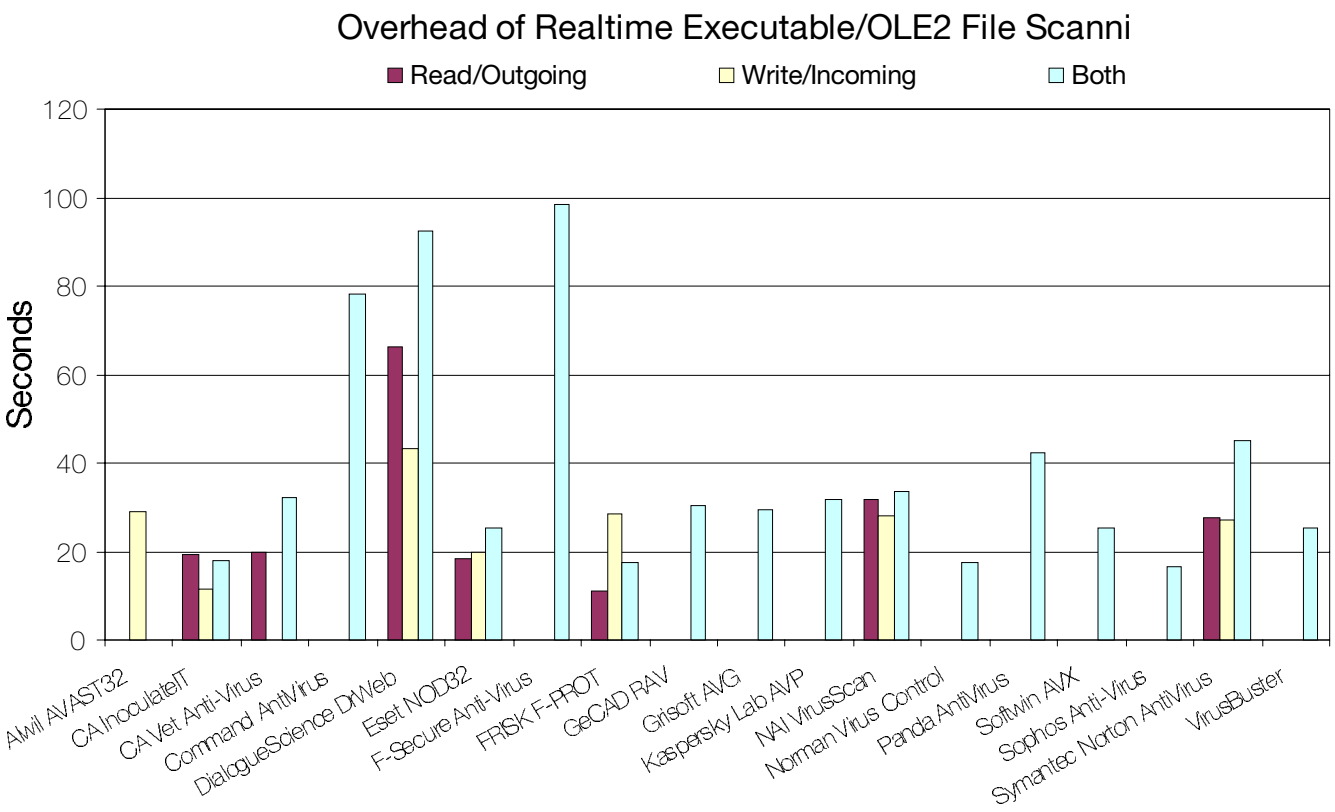
## Symantec Norton AntiVirus v5.02.04 (24/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 100.0% | Macro | 99.4% |
| ItW Overall (o/d) | 100.0% | Standard | 99.2% |
| ItW Overall (o/a) | 100.0% | Polymorphic | 94.7% |

The last of the three VB 100% awards this month, *Symantec* will no doubt make marketing capital of this slightly hollow victory. Polymorphic detection remains the *Norton AntiVirus* (*NAV*) weakpoint outside the WildList arena, though the misses here are not particularly damning given that they all fell within the samples of ACG.A and ACG.B. With results constant on-access and on-demand *NAV* definitely has cause to feel pleased with itself, but not perhaps to the same degree as some products which nevertheless failed to gain a VB 100% award this month.

## VirusBuster v3.0 (26/04/2000)

| | | | |
|---|---|---|---|
| ItW File | 85.9% | Macro | 93.9% |
| ItW Overall (o/d) | 86.3% | Standard | 91.6% |
| ItW Overall (o/a) | 85.7% | Polymorphic | 79.3% |




A second newcomer to *VB Win9x* Comparatives, *VirusBuster* had much the same baptism of fire as a number of the now well-respected products already reviewed. When reviewing a product for the first time there is always a niggling fear that there will be equal numbers of hits and misses, leading to maximum possible work, though in this case the detections were respectable if not particularly watertight. *VirusBuster* had slightly less detection ability on-access than on-demand, though this can be seen to be a common problem even with more mature products.

ItW and macro detection could in both cases be taken into the realms of good rather than OK detection by an improved implementation of *Word 97* scanning, whether by virus data or engine tweaking, since the vast majority of these misses fell into this category. More tricky to deal with might be the distinct weakness on the Polymorphic sets, though a slighly better than average scan rate should alleviate extra overhead on this front.

### Summary and Conclusions

A degree of comment concerning a couple of the samples included this month would seem to be in order. Firstly, the VB 100% awards are totally altered if JS/Unicle.A is omitted from calculations.

JS/Unicle was declared in the wild just after having been sparsely spotted (by two WildList reporters – the minimum required for a virus to make it to the list) and is a low threat (if at all) to the majority of AV users. It only operates correctly in a unicode environment, thus cutting out its threat in most of the more important market areas of those products submitted. This led to its not being a priority and not being available for some companies, thus the sparse detection in this test. However, JS/Unicle.A is on the WildList, and thus affects the allocation of *VB* scores in this, and future, Comparative Reviews. This provides yet another another opportunity to point out that VB 100% awards in one Comparative should not be used as some variety of 'buyer's guide', for it is in the short term an award where luck plays its part.

Aside from the three products earning themselves the VB 100% award this month – *Command AntiVirus*, *Eset NOD32* and *Symantec Norton AntiVirus* – some other products performed admirably against the test-sets as a whole. Readers are encouraged to view the entirety of the results therefore, and not simply flick through the VB 100% awards. The next Comparative Review (*NetWare*) will feature in the September issue.

**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 98*. The workstations were rebuilt from image back-ups, and the test-sets were stored in a read-only directory on the server.

**Virus Test-sets:** Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Win98/200007/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–S Audubon Road, Wakefield, MA 01880, USA

Tel (781) 213 9066, Fax (781) 213 9067



# END NOTES AND NEWS

**Registration bookings are now being taken for VB2000, *Virus Bulletin's* 10th international conference, which takes place on Thursday 28 and Friday 29 September 2000 at the Hyatt Regency Grand Cypress Hotel in Orlando, Florida.** For your full colour conference brochure containing programme details of the line-up of technical and corporate sessions, evening social events, product exhibition and hotel accommodation information, contact Karen Richardson; Tel +44 1235 544141, email VB2000@virusbtn.com or download a PDF copy from http://www.virusbtn.com/.

***Sophos* is to host a two-day Anti-Virus Workshop on 19 and 20 September 2000** at the organization's training suite in Abingdon, Oxfordshire, UK. On 21 September a one-day course entitled 'Best Practice for Anti-virus' will take place at the same location. Contact Daniel Trotman for details of how to register; Tel +44 1235 559933, or email courses@sophos.com.

The 17th world conference on Computer Security, Audit and Control focuses on all aspects of e-commerce. **CompSec 2000 takes place from 1–3 November 2000 at the Queen Elizabeth II Conference Centre in Westminster, London, UK.** For details visit the Web site http://www.elsevier.nl/locate/compsec2000 or contact Gill Heaton; Tel +44 1865 373625.

***F-Secure* is collaborating with the *GartnerGroup* on a series of on-line security seminars** on wireless connectivity. Topics cover a wide range of issues including proactive protection against viruses, secure broadband connectivity and policy compliance. To register for seminars scheduled between July and December contact Callie Dean; Tel +1 408 938 6700 or visit http://www.F-secure.com/securityonline/.

There are currently opportunities for companies wishing to exhibit at the **Windows 2000 eNTerprise Exhibition and Conference** which take place in the Grand Hall at Olympia, in London's Earls Court from 21–23 November 2000. Contact Deborah Holland for more details; Tel +44 1256 384000.

**The 16th Annual Computer Security Applications Conference (ACSAC)** will take place from 7–11 December 2000 in New Orleans, Louisiana, USA. Email publicity_chair@acsac.org or visit the Web site http://www.acsac.org for more information.

In mid-June, reports of a new Internet worm began to filter through to AV companies and newsgroups. **VBS/Stages (also known as Life_Stages worm, IRC/Stages.worm and variants thereof)** propagates via a number of mechanisms – *Microsoft Outlook*, *mIRC*, *PIRCH* – and also copies itself to mapped network drives. The worm body (39,936 bytes) spreads as a LIFE_STAGES.TXT.SHS file. Reputedly, most of the worm has been written by Zulu, the Argentinian virus writer also allegedly responsible for both VBS/BubbleBoy and VBS/Freelinks. While Stages is not as fast-spreading as LoveLetter – it will only start mailing if the number of entries in the address book is greater than 100 – it has infected a large number of corporate sites, and shut down numerous mail-servers.

**UK-based ASP *MessageLabs* has set up an Email Control Centre** through which ISPs, among them UUNET, Star Internet, VIA Net and INS, can route their mail for scanning and security checking. The *MessageLabs* system which processes and analyses the email includes scanning for viruses, filtering content, automatic compression of large files and storage for accurate records. For more information visit the Web site http://www.messagelabs.com/.

***Symantec* has announced an agreement with *Yahoo!*** whereby its carrier-class anti-virus technology is integrated into *Yahoo! Mail* to provide free AV protection for *Yahoo! Mail* users. Email file attachments will be scanned and if necessary cleaned and the user notified of any infection. For details contact Lucy Bunker; Tel +1 1628 592222 or email Lucy.Bunker@symantec.com.

The organisers of **iSEC Asia 2001, to be held at the Singapore International Convention and Exhibition Centre from 25–27 April 2001**, are looking for exhibitors for the event. The conference and exhibition covers IT security topics from anti-virus through encryption to biometrics and digital signatures. An early bird discount incentive runs until the end of July. For more information and a booking form contact Stella Tan; Tel +65 322 2756 or email stella@aic-asia.com.

**A recent poll conducted by *Flagship Marketing* on behalf of *Iomega*** found that small businesses in France and Germany were increasingly concerned about data loss due to computer viruses. UK-based small businesses, however, appeared to be 'dangerously complacent' in their attitudes to data protection and virus threats.