ISSN 0956-9979 AUGUST 2001

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Chinese Whispers at the Whitehouse**: ‘Code Red’ caused Whitehouse administrators to change the IP address of the official Whitehouse Web site, and even penetrated the mighty *Microsoft*’s own IIS servers. Costin Raiu analyses Win32/Bady.worm, starting on p.5.

• **CE-ing the Future**: In the light of the Common Executable File format introduced in *Windows CE 3.0*, Péter Ször warns us to be prepared for the first *Windows CE* worms for *Pocket PC*s; see p.8.

• **ASP and Ye Shall Receive**: Do customers trust their AV vendor sufficiently to outsource their anti-virus protection? John Bloodworth investigates the ASP model, starting on p.9.

CONTENTS

COMMENT

What’s in a Name? 2

VIRUS PREVALENCE TABLE 3

NEWS

1. Pentagon on Red Alert 3
2. The Greatest Form of Flattery? 3
3. SirCam Heads for the Top 3
4. EICAR 2002 3

LETTERS 4

VIRUS ANALYSIS

Holding the Bady 5

TECHNICAL FEATURE

Pocket Monsters? 8

FEATURES

1. Trust and the ASP Model 9
2. Security Bulletin Gazing 11
3. The Electronic Crime Scene 12
4. Is it a Bird, is it a Plane? 14

INSIGHT

Up to the Challenge 16

CONFERENCE REPORT

Viva Las Vegas 18

PRODUCT REVIEW

F-Prot Antivirus 3.10 for Win95/98/ME 20

END NOTES AND NEWS 24

COMMENT

“ I’d love exact identification of all variants ”

What’s in a Name?

You would think that after approximately 14 years of anti-virus research, most of which have been models of cooperation at the technical level, the industry would agree on names for viruses.

After all, the benefits of standardized names are obvious. Anyone with half a brain can tell that vbs/vbswg.Z@mm is practically the same as vbs/vbswg.J@mm, and the name gives them the information that it’s a kit generation, a Visual Basic Script and it’s a mass mailer. What’s more, they know that if they are filtering off vbs email attachments at their gateway, they have little to fear from it, and in any case, there’s a good chance that their anti-virus program will detect it as a variant.

On the other hand, names like Anna, Mawanella, HomePages or the deadly Melt-your-computer-Blamma worm (yes, I just made the last one up) convey no information at all.

The odd thing is that there is both a formula and a procedure for creating and agreeing on standard names, and while nearly everyone agrees that this is a Good Thing, not enough people do anything about following the procedure for it to count. I can only conclude that they either don’t have the technology for it, or they just don’t care.

Technology issues are one thing, but ‘don’t care’ is just not good enough for a mature industry, and is an insult to the customers who buy the products and pay everyone’s bills.

The ‘don’t care’ brigade either say, ‘It’s already in my product by that name, and I’m not changing that for anything,’ or they argue, ‘I don’t care what you say; I think my name is better’ – neither of which is acceptable in 2001.

I believe the ‘don’t care’ brigade should be made to care, and the only way that will happen is if their products are penalized in some way, either in tests or in certification programs. Watch how fast they’ll learn to care then.

The ‘technology issues’ are divided into two groups – the products that detect families but no variants, and the products that simply identify viruses incorrectly. My opinion is that family identification is acceptable, but plain incorrect identification is one of the things I’d like to weed out and, again, penalization in tests or certifications will do it pretty quickly.

In a perfect world, I’d love exact identification of all variants, but to be realistic, there are too many samples to ever expect that again. I’d be quite happy if all products could just agree on the family name and use the same structure for naming. What’s more, I don’t think it is too much to expect exact identification of viruses in the Wild, because there really aren’t all that many.

The only really difficult problem is that of timing. Some companies, bless their hearts, perceive a marketing advantage in being the first to announce a new virus. Maybe they’re correct, but frequently, in the light of the subsequent few days, other researchers find that the new arrival is simply a variant of some existing family, and correctly and logically create a new name (which again, probably provides a lot of useful information).

What should happen next, but usually doesn’t, is that everyone should be big, grown-up boys and girls, and make their products, web pages and subsequent press releases conform to the correct name. The testers and certification bodies should give everyone a grace period of, say, a month, and then start penalizing.

So, what’s in a name? Despite what William Shakespeare would have us believe, I think the answer is ‘everything’. The naming system is in place, but it’s useless unless everyone makes an effort to abide by it. Developers! Please start using it before customers start demanding it.

Roger Thompson, TruSecure Corporation, USA

NEWS

Pentagon on Red Alert

The US Department of Defence has been battening down the hatches and bringing up its drawbridges in anticipation of attack by the self-propagating Code Red (aka Win32/Bady.worm). Public access to the majority of US Defence Department Web sites, including the Pentagon Web site, was suspended over a period of four days in order for the necessary security patches to be applied. At the time of printing, at least 300,000 Web sites are believed to have been infected with the worm – for a full analysis see p.5.

The Greatest Form of Flattery?

Two fake *MS* Security Bulletins were circulated recently, in a new ploy to spread viral code. The emails imitated official *Microsoft* bulletins, complete with software patches and links to hoax Web sites. The first claimed to be a fix for I-Worm.Magistr, and contained W32/Pet_Tick.G, while the second warned of an unnamed Internet virus and contained a W32/Leave variant. The hoax Web sites have since been closed down. Some of the signs that aroused suspicion over the emails included improper punctuation and poor sentence construction (we can, at least, rely on *Microsoft* to write its press releases in decent English). It seems that, if the distributors of these viruses wish to be more successful in their bid to spread malicious code, they could do with taking some grammar lessons.

SirCam Heads for the Top

As W32.SirCam@mm put in its first appearances last month, the AV community rushed to update their scanners. A few days later, it was possible to download AV updates and view descriptions of the worm (although these vary somewhat between Web sites – perhaps due to the complex nature of the viral code). It would appear that, despite the fast action of AV developers to provide updates, this virus has been highly successful, undoubtedly helped by the fact that it does not rely solely on *Microsoft Outlook* to spread. We expect SirCam to top the next set of prevalence tables.

EICAR 2002

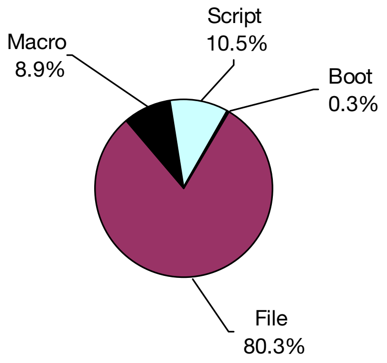
A call for papers has gone out for the 11th EICAR Annual Conference (aka 3rd European Anti Malware Conference), which takes place in Berlin, 8–11 June 2002. Papers pertaining to malicious codes and unwanted side-effects or malfunction, information age, warfare and society, cryptography and the protection of privacy, new media and e-commerce, electronic payments, are of interest. Research papers, case studies, research in progress short papers, panels, symposia, workshops and tutorials are welcome. Abstracts must be submitted by 1 December 2001.
For further information visit the EICAR Web site at http://Conference.EICAR.org/.

Prevalence Table – June 2001

Virus	Type	Incidents	Reports
Win32/Magistr	File	1223	40.2%
Win32/Hybris	File	731	24.0%
Win32/MTX	File	262	8.6%
VBSWG	Script	139	4.6%
Win32/BadTrans	File	109	3.6%
Laroux	Macro	99	3.3%
Kak	Script	65	2.1%
Divi	Macro	49	1.6%
Haptime	Script	43	1.4%
LoveLetter	Script	42	1.4%
Marker	Macro	26	0.9%
Win32/QAZ	File	24	0.8%
VCX	Macro	22	0.7%
Win32/Msinit	File	21	0.7%
Win32/Funlove	File	19	0.6%
Win32/Navidad	File	16	0.5%
Class	Macro	12	0.4%
Ethan	Macro	11	0.4%
Win32/Ska	File	11	0.4%
Win32/BleBla	File	10	0.3%
Melissa	Macro	9	0.3%
Netlog	Script	9	0.3%
Pica	Script	9	0.3%
Solaris/Sadmind	File	9	0.3%
Barisadas	Macro	8	0.3%
Others [1]		60	2.2%
Total		**3045**	**100%**

[1] The Prevalence Table includes a total of 60 reports across 27 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

Time for a Change?

The DOS comparative review in the July 2001 edition of *Virus Bulletin* raised questions within *Sophos*. *Sophos Anti-Virus* did not score 100% in all of the detection tests. At *Sophos*, we pride ourselves on providing the highest quality virus protection and have received a VB 100% award as recently as April 2001. So what went wrong?

A little investigation soon revealed the answer: very little! So, why didn't *Sophos Anti-Virus* score 100% across the board? Two reasons:

1. A very small number of viruses, that are not in the Wild, were not detected by *Sophos Anti-Virus*.
2. The tester had not configured *Sophos Anti-Virus* to scan all the files in the test set. Therefore a number of viruses were not detected, simply because the infected files were not scanned. This was the only reason that a 100% score in the 'in the Wild' test was not obtained.

The first of these is a known issue that we can, and will, address. The second is the reason for this letter. *Sophos Anti-Virus* does protect a typical system against 100% of the 'in the Wild' viruses during normal operation (see *VB* April 2001 comparative). It can also be configured to do so in the *VB* DOS test, although it does not by default.

Virus Bulletin tests with only the default options. Yet, as *Virus Bulletin* acknowledged, most customers will seek advice before using the DOS version of an anti-virus product. On contacting our technical support department they would be given the correct options to set, in order to solve their problem as quickly as possible. This is not the same for all situations, so cannot always be set as the default. Despite this, it would be quite straightforward to modify *Sophos Anti-Virus* to score 100%, by default, in the *VB* test. However, doing so would not benefit our customers, so is not on our agenda.

This leads me to suggest that it may be time to review the way that *Virus Bulletin* reviews anti-virus products. Current reviews, especially comparatives, focus almost exclusively on detection rates. This focus is demonstrated by the VB 100% award, which suggests that 100% is 'good' and 99.99% is 'bad'. Note the .99, *Virus Bulletin* does go to two decimal places to highlight differences between products. Clearly detection rates are important, but is 100% always so much better than 99.21% (for example)? Particularly when only one component of a product is tested in one mode of operation, which may not be the way that users employ the product. When the differences are this small other factors must be considered.

Rather than focusing on this 100% or nothing detection test, how about testing whether products are achieving consistently high detection results (95–98% over a year?) and then looking at other factors? I want to be absolutely clear that I am not suggesting dropping the 100% test because it is too difficult (it isn't), but simply that it is not helpful to people trying to decide which product to buy. (Yes, buy – this is commerce, not science!) The detection rates could be consigned to a single table and the rest of the space given over to a more complete 'comparative review', looking at the products as a whole. For example, *Sophos Anti-Virus* is designed to protect complete networks. It encompasses a number of software tools running on a range of operating systems, backed up by technical support. Potential users evaluate it on that basis. Surely *Virus Bulletin* would be more useful to them if it did that as well?

Previous suggestions of this sort have been met with the response ‘It’s too difficult’. I’m afraid that that is just not good enough. These are complex products protecting complex environments against an ever-changing threat. Reviewing them properly is almost certain to be difficult, but this is what *VB* has chosen to do. *Virus Bulletin*, its readers, and the software developers, would all benefit from a more complete analysis. I would love to see *Virus Bulletin* provide a range of information, draw conclusions and, dare I say it, make recommendations to help users choose the right product for them.

Richard Jacobs
Sophos Anti-Virus
UK

VB Responds

It has often been proclaimed that *VB*’s comparative tests are not indicative of real-world behaviour. You may shout because you see this as a bad thing, but we shout equally because we see it as a good thing. The VB 100% award states that a certain criterion has been reached, that is the format: tried, tested and a known quantity.

To this extent the work of *VB* is a science, not commerce. We do not ever say, ‘This product is awful, do not buy it.’ We do say, ‘This is a problem encountered under test conditions, if it is likely to affect your implementation of the software then you should be aware of it.’ Yes, the comparative reviews concentrate on detection rates, yet as a result of this, there is a large amount of other data produced which finds its way into the body of the test. Standalone reviews, on the other hand, have been known to contain absolutely no reference to detection and are directly complementary to the comparative tests.

As for making recommendations, this is precisely the way that anti-virus software should *not* be reviewed. A far better way is to present information to allow an informed opinion. *VB*’s information is not designed to tell the mindless what to do or buy, it is written for people who know what they want but want to know the issues in depth.

Matt Ham
VB Technical Consultant

Erratum

Virus Bulletin apologises for an error in the letter from Andreas Marx, ‘Another Scheme of Retrospective Testing?’ in the July 2001 issue of *VB*. The letter read ‘…we are working with the University of Hamburg to try to achieve something in this area.’ In fact, Andreas Marx and the University of Hamburg are working on the same issues, but are not working together at the moment.

VIRUS ANALYSIS

Holding the Bady

Costin Raiu
Kaspersky Lab, Romania

After working for over four years with macro and script viruses, I recently came across a piece of malware which gave me cause to dig out my old toolbox and blow the dust off my old disassemblers and debugging tools.

Although the last time I dug out my old toolbox was actually not such a long time ago (that occasion was due to another curious piece of binary data – the executable from the macro virus Class.EZ), this time the reason was not only a little different, but proved to be much, much trickier, and harder to figure out in its deeper internals.

The Bug

On 18 June 2001, *Microsoft* released its 33rd Security Bulletin for this year, dealing with a simple buffer overflow in one of the DLLs used by the indexing service ‘idq.dll’. Credited to the people from *eEye Digital Security*, the bug proves, once again, that *Windows NT* and the server software running on *NT* systems are not spared by the most common security vulnerabilities of Unix systems, the buffer overflows.

The original security advisory from *eEye* did not include an exploit, but it wasn’t long before a couple were written and started to crawl around. One of them was even posted to the *SecurityFocus* Web site, in the exploits section for this specific vulnerability, therefore becoming generally available to the masses.

However, by far the most interesting exploit came in the form of a computer worm, which not only exploits the vulnerability, but replicates the exploit further, to other servers from the Internet. Initially, the worm was named ‘Code Red’, but the common name selected by the AV industry for this worm is ‘Bady’, either in the form of ‘Win32/Bady.worm’ or the more complex, but even more CARO-compliant name, ‘worm://Win32/Bady.A’.

The Worm

The worm code is written in Win32 Intel assembler, and is 3569 bytes long, if we count the data used and carried by the worm along with the executable code.

Due to the nature of this exploit, what was probably one of the trickiest parts was to transfer control to the worm code from the instructions that receive control after they smash the stack. In fact, this is so tricky, that it can work only under very specific conditions, thus limiting the possibility for the worm to spread. Also, it was so tricky that it took

```
“%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00”
```

Figure 1.

four hours before I realized why the worm simply crashed my test system, and didn’t want to work.

Basically, the worm sends 224 ‘N’ (0x43) bytes via an HTTP GET request, and it appends a few bytes of executable code after them. The small piece of executable code is encoded in the URL it sends for processing to the ISAPI server extensions, and looks like that shown in Figure 1 above.

The buffer overflow will fill the stack with the 224 ‘N’ bytes expanded to two-byte UNICODE representations of the form {0x4e, 0x0}, which are used as return address when the subroutine in which the buffer overflow took place returns.

After that, the execution flow will hopefully hit one of three eight-byte-long sequences designed to prepare (again!) the stack for another jump, which is designed to hit the real worm code. The jump is performed in quite a tricky manner, and it relies on the fact that at a certain address in memory we can find a specific instruction, a two-byte-long ‘call ebx’.

However, the respective instruction, which is supposed to be located in the memory image of the standard system module ‘msvcrt.dll’ (Microsoft Visual C Runtime Library) at offset 7801CBD3h, is in its place only if the respective library is version 6.10.8637 – exactly the one distributed with *Windows 2000*, Service Pack 0, which is exactly 295000 bytes long.

So, if either SP1 or SP2 has been installed on the machine, the worm will be unable to spread. The same is true if the machine is running *Windows NT 4.0*, and in all these cases, the WWW Publishing Service of IIS will simply crash when attacked.

However, if the system runs the ‘good’ version of ‘msvcrt.dll’, the worm performs the jump correctly, and reaches its main code, which begins to take the steps necessary for the worm code to carry the infection further.

First, it will allocate stack space to store 134 (86h) DWORDs, and it will also take care to wipe it using 0CCh bytes. Next, the worm tries to obtain the location of the very useful API GetProcAddress, using a method which is actually very common to most PE infectors. For this, the worm scans the memory range 77E00000h–7800000h, incrementing in steps of 64K, looking for a ‘MZ’ signature. Obviously, this check attempts to find the memory image of ‘kernel32.dll’ (which, for example, is found at offset 77e80000h in the initial release of *Windows 2000*).

If the worm does not find the ‘MZ’ signature of ‘kernel32.dll’ in that range, it will attempt to look for the same thing starting from 0BFF00000h, obviously assuming that maybe the system is not *NT*, but Win9X (for example, in Win98 the ‘kernel32.dll’ module is located at the address: 0BFF70000h).

Check and Cross Check

After finding the possible address of the ‘kernel32.dll’ PE image in memory, the worm will perform a couple of additional checks to be certain that it is indeed the ‘kernel32.dll’ module. For this, it will check that it is a PE file and then find the export table to check if the module name matches ‘KERNEL32’.

If the respective checks fail, the worm code continues scanning. It’s amusing to note how careful the author was here to find the correct address of the kernel module image in memory while, a few instructions ago, it simply assumed that ‘msvcrt.dll’ contains a {0FFh, 0D3h} sequence (call ebx) at 7801CBD3h. I think this was due to the author using the respective code from some PE virus, and he/she didn’t bother to remove the Win9X part. Also it seems useless to perform such careful checks for the ‘kernel32.dll’ module, when the earlier assumption regarding ‘msvcrt.dll’ has already been made.

After finding the correct address in memory of ‘kernel32.dll’, a short subroutine is called to determine the offset of the ‘GetProcAddress’ exported entry. This subroutine will simply parse the export table, and verify if any of the entries is indeed ‘GetProcAddress’.

Next, ‘GetProcAddress’ will be used to obtain the address of other common APIs, which are LoadLibraryA, GetSystemTime, CreateThread, CreateFileA, Sleep, GetSystemDefaultLangID and VirtualProtect. Of these, LoadLibraryA will further be used to load and obtain the memory offsets of the images of ‘infocomm.dll’, ‘WS2_32.dll’ and ‘w3svc.dll’. The worm then extracts the TcpSockSend subroutine address in ‘infocomm.dll’, as well for the addresses of the ‘socket’, ‘connect’, ‘send’, ‘recv’ and ‘closesocket’ subroutines in ‘WS2_32.dll’.

Replication and Payload

Next, the worm spawns 100 threads in memory which are designed to carry the main replication code as well as the payload. However, due to a bug, the worm will try to spawn even more threads for each thread created, therefore quickly eating a huge amount of resources, meaning it is less likely to go unnoticed on an infected server.

Each thread runs exactly the same code, which acts as follows: first, the worm attempts to open a file named ‘c:\notworm’. If successful, the worm will start to issue

‘Sleep’ calls of about 24 days, ad infinitum. However, if the respective file is not found on disk, the worm continues in its progress.

It will check whether the current day is between 20 and 27 and, if so, it will run the part of the payload which consists of sending 18000h times one-byte-long TCP/IP packets to the IP address ‘C689F05Bh’ which, in a more readable form, is 198.137.240.91, and which currently resolves to the name ‘www.whitehouse.gov’. [*This is no longer the case, since the IP address of the Whitehouse Web site has been changed - Ed.*]

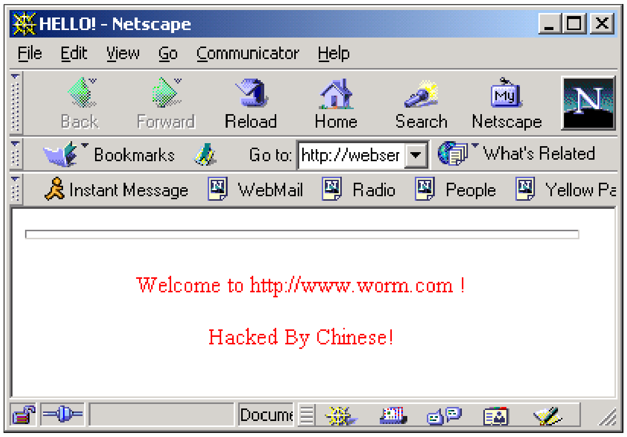
Next, the worm will run its random number generator routine, the purpose of which is to provide targets for infection. The routine uses two things as seeds for the stream of random numbers: the current second/millisecond fields of the current system time, and the thread number.

Combined, these two could produce a lot of different IP streams. I say ‘could’ because of the way the algorithm works – the entropy provided by the ‘second’ and ‘millisecond’ fields of the current time is lost in the computations, so that leaves us with exactly 100 possible streams of IP addresses, which only depend on the thread index, again, only in the range 0–99.

Therefore, whenever a copy of the worm receives control, it will start hitting a predictable invariant stream of IP addresses, thus highly limiting its ability to spread. For example, the stream of IP addresses generated by the first thread in the worm will always start with the following values: 7.107.254.83, 252.118.171.204, 198.83.139.183, 33.250.241.248, and so on.

Interestingly, this mistake seems to have been noticed by the author too; after the initial version of the worm became widespread, another ‘fixed’ version was reported. The second version seems to have its random number generator routine fixed, thus having much better chances to spread over the Internet.

The worm has another interesting payload which is run only if the current system codepage is 0x409, US English.


HELLO! - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape
Bookmarks Go to: http://webser What's Related
Instant Message WebMail Radio People Yellow Pa
Docum

Welcome to http://www.worm.com !

Hacked By Chinese!

Figure 2: Web page.

First, the worm will run a Sleep call set to two hours, and after that, it prepares to launch the payload. For that, it will scan the import table of ‘w3svc.dll’ for an API named TcpSockSend. After finding it, the worm replaces it with a pointer to a subroutine inside the worm copy which sends a specific Web page whenever a request to the HTTP server arrives. The Web page is shown in Figure 2.

It should be noted that, while patching the export table of ‘w3svc.dll’, the worm takes care to write-enable the area of memory in which the module is stored. This is required in order to patch the address of TcpSockSend, the function hooked by the worm. From here, the worm will simply loop again, trying other IP addresses.

Conclusions

There has been much debate around the fact that this may be the first modern worm that doesn’t exist at any time in a file, nor use temporary files during replication, as for example *Linux*/Cheese or *Linux*/Ramen do.

‘Bady’ certainly exists only in memory or as a TCP/IP stream sent around the Internet, thus making it the perfect example for everyone’s definition of the term ‘worm’. But besides that, the truly important thing about it is that the impact of this worm could have been much, much worse.

If the worm had been written a little more carefully, to infect more than just *Windows 2000* systems running IIS4/5 with the indexing service installed and to use a really ‘random’ stream of target IPs, then stopping it would have been much more difficult. However, regarding detection, unfortunately the AV world was, in its majority, unprepared to handle ‘Bady’. To detect and stop this worm, scanner plug-ins for firewalls are needed and, unfortunately, these are not very common. Also, to detect and clean it in memory, a couple of improvements to the scan engines are probably needed, such as the possibility to scan the memory associated with a thread launched in the memory space of a module attached to a process …

W32/Bady.worm

Aliases:	Code Red, CodeRed.
Type:	Network-propagated worm.
Infects:	*Windows 2000* machines running IIS4/5 with ISAPI enabled.
Payload:	Attempt to flood www.whitehouse.gov between 20th and 28th of each month – hooks all HTTP requests on systems with codepage 0x409, and sends a custom page back to the clients.
Removal:	Stop the WWW service on the affected machine, install the *MS* recommended patch, then restart the WWW service.

TECHNICAL FEATURE

Pocket Monsters?

Péter Ször
Symantec Corporation, USA

Recently, I came across an article on the Internet concerning *Windows CE* virus security issues. The author of the article believes that very few, if any, *Windows CE* viruses will be created successfully.

The author found that *Windows CE* does not support macros in *Microsoft* products such as *Pocket Word* or *Pocket Excel*. Furthermore, he pointed out that *Windows CE* machines (small footprint and mobile 32-bit devices) use a number of different processors, making it difficult for binary viruses to replicate on the platform. (This statement probably needs to be revised a little in the light of *Windows CE 3.0*, as I shall illustrate.)

In my 1998 *Virus Bulletin* conference paper I predicted that we might see binary viruses on *Windows CE* platforms. However, it does not surprise me that, to date, no *Windows CE* viruses have been encountered. The variety of processors used in *Windows CE* devices is probably the main reason behind this. However, I believe the situation might change over time.

In this article I shall illustrate a problem that is very similar in its nature to the upconversion problem of *Microsoft* document macros. This time I will be talking about executable files.

Pocket PC Situation

Windows CE 3.0 was introduced last year as the underlying operating system on *Microsoft*'s *Pocket PC*. The *Pocket PC* is a little bulky and expensive; last year you could buy one for around $500. This year the price has dropped sharply, and it is available for around $300. The system is packed with new features and may become very popular. However, some of these new features are not so welcome from the point of view of virus security.

The Executable Problem

Until now, developers have experienced significant difficulty in creating and distributing *Windows CE* executables. This is due to the number of different processors used in *Windows CE* devices.

Executables were developed in binary format as Portable Executable (PE) files, but each was only compatible with the processor on which it had been compiled to run. For instance, an SH3 processor's PE file header will contain the machine type 0x01A2. Its code section will contain code that is compatible for that architecture only.

It has also been difficult to install and support compatible software on these devices – again, because the executable file had to be compatible with the device in question.

All current *Windows* viruses that we know of are dependent on *Intel* platforms. Although an application could easily be created and compiled to run on SH3 platforms, *Windows CE* is ported to about eight processors, including the SH3, SH4, MIPS, and so on. Thus a native *Windows CE* virus would be unable to spread easily between devices that use different processors.

Virus writers might be able to create a Win32 virus that drops a *Windows CE* virus via the *Microsoft Active Syn*c. That virus could easily send emails and propagate its *Intel* version (with an embedded *Pocket* version). However, it would only be able to infect a certain set of handheld devices that use a particular processor.

Recently, *Microsoft* introduced a feature that made the *Windows CE* developers' job a little easier. This feature was first discussed at a *Microsoft* conference in 1999, but was not released until *Windows CE 3.0* (*Pocket PC*).

Introducing the Common Executable File (CEF)

In *Windows CE 3.0*, a new executable file format is supported: the Common Executable File (CEF) format.

CEF executables can be compiled with *Windows CE* development tools such as *eMbedded Visual C++ 3.0*. A CEF executable is essentially a Portable Executable (PE) file. (The PE file format is supported on all major 32-bit *Windows* platforms as well as on the upcoming 64-bit *Windows XP* on the IA64.)

Common Executable Format is a processor-neutral code format that enables the creation of portable applications across CPUs supported by *Windows CE*. In *eMbedded Visual C++*, CEF tools (compilers, linkers and SDK) are made available to a developer in the same way as specific CPU targets (such as MIPS or ARM).

When a developer compiles a CEF application, the compiler and linker does everything but generate machine-specific code. You still end up with a DLL or EXE, but the file contains intermediate language instructions instead of native machine code instructions.

CEF enables *Windows CE* application developers to deliver products that support all the CPU architectures that run *Windows CE 3.0* and above operating systems – and because CEF is an intermediate language, it is easy for processor vendors to add a new CPU family that runs CEF applications. For instance, HP Jornada 540 comes with a built-in device translator layer. The CEF file might have an

.EXE extension when distributed, so nothing changes from the user's perspective.

Device Translator

The device translator is specific to a particular processor and *Windows CE* device. The device version translates a CEF executable to the native code of that processor when the user installs the CEF executable on the device. This occurs seamlessly, with no indication to the user that anything is happening other than a brief pause for translation after the executable is clicked on. An operating system hook catches any attempt to load and execute a CEF EXE, DLL, or OCX file automatically and invokes the translator before running the file.

For instance, if the *Pocket PC* is built on an SH3 processor, the translator layer will attempt to compile the CEF file to an SH3 format. The actual CEF executable will be replaced by its compiled SH3 native version, changing the content of the file to native executable completely. Integrity checking? You can forget about it.

Virus writers might take advantage of the CEF format in the near future. A 32-bit *Windows* virus could easily install a CEF version of itself onto the *Pocket* device. This way the virus could run on all *Pocket PC* devices since the OS would translate the CEF executable to native format on each device.

This is a major problem for anti-virus vendors. How can the AV detect such executables once the compilation has happened? At least we do not need to worry about down-conversion issues. The process happens only one way from CEF to a number of processors that have such translation layers.

Conclusion

At the end of last year, when I discovered the CEF problem while researching my *Pocket PC*, I also considered what would happen if similar desktop translators were implemented. That would certainly be troublesome.

For a while it looked as if *Microsoft* had not thought of this idea. However, *Microsoft .NET* is coming up with its own extended PE format that contains Microsoft Intermediate Language (MSIL) as code.

The Just In Time (JIT) compilation of .NET executables might raise only memory scanning issues in the short run. However, the system is still at the Beta stage. The .NET documents reference a mysterious ngen (Native Image Generator) application to make .NET executables run faster using JIT if no cached native image is available. This sounds scary enough to me!

In the very near future a new level of compatibility could appear between various platforms that could make handheld viruses a real problem. In the meantime, be prepared for the first *Windows CE* worms for *Pocket PC*s.

FEATURE 1

Trust and the ASP Model

John Bloodworth
McAfee, UK

The overriding feeling on the subject of ASP-provided anti-virus at last year's *Virus Bulletin* conference was that, while the idea itself may be sound, many customers do not trust their AV vendor, for a variety of reasons.

Has this view changed over the past few months and, if so, are customers more receptive to outsourcing their anti-virus to external providers? If not, why not? And what can be done to gain that trust?

There are many issues both for and against outsourcing any part of a corporate security policy, such as privacy, skills shortages, trust etc. However, there must be a fine line across which outsourced policy stops being a liability to companies and starts to become a viable and attractive solution to the widespread and ever-increasing issue of anti-virus protection.

I believe that the objections raised against ASP solutions when the subject was raised at last year's *VB* conference were most likely not objections to the solution itself. Rather, people were pointing out internal issues which prohibit the use of such solutions in some companies. Since businesses already use anti-virus software, I find it very difficult to believe that they would not trust their AV vendors to provide a service which removes a lot of the problems of keeping anti-virus software up to date.

Over the last couple of years we have seen home users embrace AV services provided over the Internet – for example *McAfee Clinic* and *Trends Housecall*, where users can scan their hard drives to find and remove malicious code. We know that people use and trust these kinds of services when they are at home, so how do businesses see these services?

Small Business ASP

There are some very obvious candidates who would benefit from this kind of service: the small and non-technically focused companies. This kind of business generally has less than 100 employees and, as a result, does not have a dedicated IT infrastructure team – or might have one or two IT staff who are generally too busy to ensure that anti-virus updates are performed every week.

In this kind of environment, having anti-virus software which is controlled from an external source and is, in essence, versionless, would allow those responsible for the IT infrastructure to focus on other issues, safe in the knowledge that their AV software is always up-to-date.

Historically, the barrier to companies being able to benefit from this kind of service has been lack of bandwidth. Trying to provide constant anti-virus maintenance through an ISDN connection or even a dial-up modem would be difficult as, generally, the small bandwidth would already be stretched to capacity dealing with the company's email and Web browsing needs. And, in cases where the connection was not in such constant use, the line might be dropped to reduce overheads. These days, as ADSL, cable modem etc. become more common, smaller businesses are beginning to benefit from the sort of bandwidth that was, until recently, enjoyed only by larger companies who could afford the steep costs involved in installing and leasing permanent connections to the Internet.

ASP and the Larger Business

If lack of bandwidth has been a barrier to companies using AV services over the Internet, you might ask why larger businesses that have had fast Internet connections for a number of years have not embraced ASP technology.

The first reason is that larger companies generally have staff who are dedicated to looking after the anti-virus software within the business, and they are reluctant to allow an outside source access to each and every PC on their network. This level of caution is commendable, but prohibitive to an ASP service.

The second reason is that, because the businesses have dedicated staff to look after AV software, there is a reluctance to hand this task over to a third party. Relieving staff of the duty of updating AV software may, initially, sound like a good thing, allowing those staff to devote their attention to other matters. However, businesses are always on the lookout to cut costs so there is the potential, should ASP services be taken on board, that those who make budgetary decisions will see an opportunity to reduce both the number of staff and the budget of the IT department. Therefore, those who look after the IT infrastructure may refrain from presenting the ASP services as a viable option for fear of losing staff and/or budget.

There is another, very good reason that many companies will be reluctant to make use of ASP services. Many companies have an IT policy in place, meaning they must test any software against the standard build of their desktop PCs to ensure that it does not interfere with any of the applications already installed. Every time a new set of AV signatures is released, they must be tested against this standard build before they can be rolled out to all the desktops. Businesses feel they need to go through this practice to ensure that no loss of business continuity is caused by installing new software. However, this is a prohibitive obstacle to ASP-provided anti-virus, and companies who follow such a policy will not be able to utilize such services.

It comes as no surprise that the majority of companies that do make use of ASP services fall into the 'small business' category. But, given their objections, it is surprising to note that some larger businesses have subscribed to these services. Not all larger businesses suffer from the problems I have outlined above, and as you will see below, these services can complement the AV solutions already in place.

The larger companies that have taken on such services are those that have realized that ASP services are not an 'all-or-nothing' solution. It is possible for a larger company to retain control over the majority of their anti-virus installations, using conventional AV software and management tools for the majority of their desktops, while using the ASP-provided software in a complementary role.

Take, for example, a large retail company with many retail outlets. The majority of the company's desktop PCs are located in the main offices, where conventional AV software is used. However, each of the retail outlets has just a handful of PCs and no one with the IT knowledge required to maintain them. Historically, the logistics of keeping the AV software on those PCs up to date has been difficult due to the wide geographical spread of the outlets and the expense of maintaining a full WAN infrastructure to each and every outlet. In this kind of environment, businesses have realized that the use of ASP AV software at remote sites allows them to feel secure in the knowledge that all of those PCs will be kept up to date with little or no central intervention from the IT staff.

Conclusion

ASP solutions do not address all of the objections raised against them, and they are not meant to. ASP anti-virus is a complementary solution to conventional AV software and it will continue to be so for a long time to come. The ASP solution is designed to enable smaller companies to maintain their AV protection without the cost of dedicated staff to look after it. It is also designed to aid dedicated AV administrators in maintaining up-to-date protection of areas of their IT infrastructure that are difficult and expensive to reach, *not* to remove the need for the administrators themselves.

The introduction of ASP-level anti-virus services mirrors a change in the way anti-virus is seen and provided. Today, companies are demanding solutions rather than just software in a box and, as such, ASP AV software can now be leveraged as a new layer in the array of products available to help complete the solutions that are offered.

There will always be businesses that will neither use nor require ASP AV software, either because of policies which prevent third-party control over their desktop PCs or simply because they already have the ability to easily maintain an up-to-date AV solution across their entire environment. This is to be expected. ASP software is not designed to replace every piece of conventional AV software currently in use; it is designed to offer yet another option to businesses when they are forming the solution that provides the best protection for their networks.

FEATURE 2

Security Bulletin Gazing

Aleksander Czarnowski
AVET Information and Network Security, Poland

More than a decade ago, when viruses first started to appear in Poland, I realized that predicting future threats would be one of the things that customers would want. After all, customers are not only interested in protection against today's threats, but also want to be protected against those of tomorrow.

Predicting the Future

Guessing at the future is something that is done on a regular basis at almost every IT security company. And I must admit, some vendors are quite good at this game.

How can one predict future threats? Well, current security vulnerabilities might give us a good idea. If we analyse a dozen of *Microsoft*'s Security Bulletins we will see some potential problems which might be taken advantage of by malware.

MS00-57

Let's start with an old problem that is still found in the Wild: Microsoft IIS Unicode directory traversal vulnerability. This problem was identified around October 2000 and *Microsoft* released a patch with the *MS00-57* security bulletin. Despite its name, this vulnerability allows an attacker not only to travel around the vulnerable Web server directory structure, but also to execute any desired program remotely by sending a URL. This could be used in payloads or infection routines, especially considering that sending a URL is a trivial task. This vulnerability shows that very dangerous security problems can exist in complex applications. It also demonstrates that malware can make use of such problems. For example DoS.Storm.worm exploits this vulnerability.

MS01-023

Security Bulletin *MS01-023* describes buffer overflow in ISAPI Extension, which leads to remote server compromise. This Bulletin was originally published on 1 May 2001, but there are still some vulnerable hosts connected to the Internet. There is one important difference between this and the last hole. The first problem affects both IIS 4 and IIS 5, while the second affects only IIS version 5. Even if worms were to infect only servers with IIS 5, they would still be quite effective.

It is worth mentioning that applying the security settings described in the 'Securing IIS' documents published by *Microsoft* could stop many attacks against IIS server.

Local Vulnerabilities

Worms and viruses could be wonderful compromise tools because they usually run in the user context with all the user's privileges. In many systems (like Win9x) this means full access to all resources. Even in Win32 systems with more advanced security mechanisms (like *NT*/*2000*), users have the ability to install new software (so called 'PowerUsers'). This is more than enough to take a control of a system.

In *Windows 2000* (below SP2) there is 'Debug Register' vulnerability. Any unprivileged process can set up breakpoints in other processes. This is possible due to fact that X86 DR0-7 debug registers are global for all processes. In the end, it is possible to stop any process or to escalate privileges. DR0-7 registers are also used in anti-debugging techniques. Such vulnerability allows malicious code to gain additional privileges even when executing as a non-administrative user.

MS01-028

There are other local vulnerabilities that can be used by viruses. On reading Security Bulletin MS01-028, one might think that this problem was created specially for macro viruses. Due to error, or rather lack of checking for macros during opening RTF documents that link to a *Word* template, no macro warning will be displayed and the macro security mechanism will be bypassed.

Other Platforms

It would not be fair for me to talk only about Win32 vulnerabilities. Other platforms have dangerous security problems too. In the Unix world there are still vulnerable BIND daemons, which allow root privileges to be gained remotely. The vulnerable glob() function was used in a dozen ftp servers for Unix. One of the recent *Solaris* vulnerabilities in yppasswd also allowed remote compromise of the host. And (RedHat) *Linux* had vulnerabilities in LPRng that were exploited by *Linux*/Ramen.worm.

So what will the future hold? Well, we have more and more complex applications like *Microsoft Internet Information Services*. Also, *Linux* distributions have a tendency to grow in code size, which means more potential vulnerabilities. Users and administrators tend not to read documentation and often do not follow vendors' advisories. Consequently, the number of vulnerable hosts connected to the Internet will rise. This increases the chances for new worms to spread rapidly. So we will have to face them soon.

I can hear you saying; 'That guy didn't tell us anything new.' Correct; and have you applied your patches today? Now you know why vendors predict the future so well …

FEATURE 3

The Electronic Crime Scene

Edward Wilding
Maxima Group Plc, UK

The successful prosecution of a computer crime, fraud or other malpractice in the workplace is largely dependent upon the initial actions of those who have been charged with its investigation.

Preserving electronic evidence in the corporate environment is a painstaking discipline, equal to the management of evidence where murder or physical assault has occurred. It is common knowledge that the scene of crime can easily be contaminated; the correct handling of fingerprints, blood, hairs, fibres, toolmarks, shoe prints and the assault weapon itself are vital considerations. A common complaint from Scene of Crime Officers is that untrained personnel – usually the first to attend the scene – destroy evidence, or render it inadmissible, through ill-advised handling or other tampering.

As any corporate investigator will testify, the same applies in the workplace. This article itemises some basic factors in securing and processing the 'electronic scene of crime', where computer crime or misuse has occurred, is suspected, or where digital evidence is present.

Fruit from a Rotten Branch

Experience suggests that any electronic scene of crime, when processed by inexpert staff, will be compromised within 30 minutes of its discovery. Hasty, over-enthusiastic examination of computers has resulted in evidence being destroyed or rendered legally inadmissible.

A typical management response, in circumstances in which computers require examination, is to delegate the task to the in-house IT department, despite the fact that very few IT staff have any experience of evidence handling or computer forensic examination. Ultimately, any 'evidential' product resulting from an unqualified examination may be ruled inadmissible and will certainly be subjected to a vigorous legal challenge, as the following example illustrates.

A PC support desk technician executed a series of diagnostic programs on a company workstation. It became evident that the computer had been used to access pornography. The technician assumed that the designated user of the computer was in breach of company Internet policy. With a mixture of curiosity and misplaced investigative zeal, the technician proceeded to trawl through the entire contents of the Internet cache, which stored thousands of image files.

Management was informed of the discovery and the technician was instructed to print the offending images which were submitted as evidence of the user's alleged misconduct.

Oblivious to established forensic practice, the ensuing examination took place using the computer's native operating system and software. The technician even installed diagnostic software and printer drivers onto the evidential computer, created files and folders, and unwittingly generated hundreds of temporary files and printer spool records.

Hundreds of files were thus created *after* the computer had been impounded for evidential processing, and thousands of native files were altered by the technician's misguided actions (the last access dates had changed). This being the case, what other material changes had been introduced onto the computer? The evidence appeared to be irrevocably tainted and no firm conclusions could be drawn about the computer, the circumstances of its use, or its owner.

In this case, the evidence was tarnished because files were altered after the commencement of the investigation. The assertion that the evidence produced is 'fruit from a rotten branch' is a common defence argument raised in criminal cases. Forensic practitioners can cite dozens of similar cases where evidential integrity is compromised by well meaning but unqualified examination.

Computer Forensic Discipline

As soon as *Windows* (in any of its guises) is initiated, the data structure on the host disk is altered irrevocably. Data is cached, a large number of temporary files are created, some temporary files are purged, the swap file is altered, and a range of other more subtle changes occur. Consequently, underlying evidence may be overwritten. This point is not academic – swap files and residual data in slack space have proved an abundant source of critical evidence in a range of investigations. The use of a clean write-protected system diskette, an inherent part of best anti-virus practice, is also a vital tool as it prevents the operating system on the evidential disk from executing.

Computer forensic discipline developed from the overwhelming law enforcement requirement to examine seized evidential computers and data storage devices in a way that preserved the integrity of the original evidence.

The principles of established computer forensic 'best practice' are:

- Examination of the data should *never* be conducted on the original storage device, but should be undertaken on a copy of the data stored on the device.
- Ideally, the copy should be an accurate image of the entire data area. Typically, this image will be of a

computer's internal hard disk, but the principle of 'entirety' applies equally to evidential audit trails or logs, databases or any other relevant information.

- The backup method will be *non-invasive*, i.e. it will not alter the constitution of the original evidential data in any way. A single-bit change introduced into the original data area can render evidence inadmissible.
- The evidential computer's operating system should never execute; the computer must be clean booted from a system diskette or system CD.
- Ideally, two copies of the evidential data should be made (however, this is not mandatory under UK law).
- The copied data or forensic image should be tamper-proof, either by the use of cryptographic checksums or CRCs, or because the data is recorded to WORM drives or write-protected media.
- The backup process should generate an audit trail. The audit trail should record the system date and time. The forensic examiner should record the real date and time, so that any discrepancy between the two is apparent.
- The circumstances under which evidence is secured should be recorded in notes and written statements.

Electronic copies of files, folders, data, partitions, disks, programs, or any other storage agent, executable or digital document are admissible under both civil and criminal law in the UK, as they are in nearly every other jurisdiction. It is paramount, however, that the integrity of the copied data can be demonstrated. There is a sound practical reason for working on copies. Imagine trying to examine the formulations used in a large Excel spreadsheet, while simultaneously retaining the integrity of the original file, if no copy of it were permitted! Such examination is only possible by having access to a write-protected master file, from which unlimited inspection copies may be generated.

Mishandling of computer evidence is a common phenomenon. Equally problematic is the failure of systems to generate audit trails and logs that are sufficient to identify the perpetrators of computer misuse.

Dynamic workstation addressing, prevalent in many *NT* environments, can frustrate investigative efforts, because it confounds attempts to resolve each workstation's temporarily allocated IP address to a hard-coded MAC address, and thus frustrates efforts to identify an offending workstation. At the very least, the investigator will want to identify the computer used to perpetrate a crime, and the Userids active and associated with that machine. In an *NT* environment, this is an onerous task without DHCP logging.

Forensic Culture

Due to the very different perspectives involved, the culture of the forensic investigator is nearly always at odds with that of the IT professional. The following case study is instructive.

A hacker waged a personal war against a utilities company over a period of five months. The company operated a wide area network running the *Solaris* operating system. The hacker induced system crashes, caused random damage, and destroyed proprietary databases. On occasions, the hacker covered his tracks by destroying or disabling audit trails and event logging. The attacks escalated, became more confident, aggressive and destructive. An investigator was brought in to work alongside the system administrator, who was a Unix specialist with a commanding knowledge of the *Solaris* operating system and applications.

It soon became clear to the investigator that the inquiry would prove very difficult and frustrating. Unfortunately, the system administrator had not retained any of the log files or audit trails. On occasions, these had reportedly recorded the misuse of the root account, including the dates and times of the attacks, but this vital data was now irretrievable. Likewise, the administrator had made no diary entries as to when the incidents occurred. The absence of the audit trails and event logging denied the investigation key evidence. Moreover, the pattern of the attacks could not be analysed because the administrator had not recorded which resources were manipulated, targeted or destroyed.

The system administrator prioritised the security of the system and acted swiftly to block and disable the intruder. However, he failed to appreciate the investigative value of recording the attacks, nor did he appreciate that this data, collated over a period of weeks or months, would provide a strong portfolio of evidence that could be used to identify and prosecute the culprit. IT personnel often prioritise system security but fail to appreciate the legal, evidential and investigative ramifications of a security breach.

Conclusion

This is a brutally short article about an expansive topic. It is quite probable that many readers of this article will be required to produce computer evidence at some point in their careers. Consequently, I would urge all IT professionals (including virus hunters) to acquaint themselves with the rules of electronic evidence handling and the common pitfalls that bedevil an investigation.

Three misleading assumptions that regularly feature in computer crime cases:

1. The dates and times shown in audit trails or in electronic files are presumed to be an accurate reflection of the *real* date and time.

2. Userids and computer accounts are commonly confused with the actual people to whom the accounts are assigned, e.g. 'Smith's account was used to commit the crime; therefore, Smith committed the crime.'

3. Equipment is wrongfully associated with its owner, e.g. 'Jones' telephone was used to make the incriminating call, therefore, Jones made the incriminating call.'

FEATURE 4

Is it a Bird, is it a Plane?

Robert Vibert
Anti-Virus Information Exchange Network (AVIEN)

From a chance meeting of a few anti-virus specialists over cocktails at the opening drinks reception of VB 2000 in Orlando, Florida, the Anti-Virus Information Exchange Network (AVIEN) has grown into what Daryl Pecelj, Anti-virus Senoir Program Manager at *Microsoft Corporation*, characterizes as 'the world's top corporate virus fighters who collectively have more experience, knowledge and insight into the anti-virus arena than any other entity'.

With membership of the network representing currently more than 2.5 million computers protected by anti-virus software, and a global coverage that includes most countries in the world, AVIEN has had an amazingly swift rise – in less than a year – to a position of prominence in the world of malware defence.

As the moderator of AVIEN, I frequently hear the popular misconception that AVIEN 'belongs' to me and my company, *Segura Solutions*. I was pleased to be given the opportunity to set the record straight and explain AVIEN to the readers of *Virus Bulletin*.

Where Did it All Start, Really?

There were two paths that lead to the formation of AVIEN: discussions with Ken Bechtel (who, during the day, wields the AV shield for some 40,000 users at *Tyco Electronics*) about his vision for a Team Anti-Virus project (which included the certification of AV specialists), and my discussions with *Nortel* and other Canadian organizations about the possibility of organizing some sort of anti-virus user group in the Ottawa area.

The two ideas had common elements, the most important of which was the requirement for some means by which those organizations who employ anti-virus software could share their experiences and expertise.

Another driving force was the need expressed by corporate anti-virus specialists for access to more accurate and detailed information about viruses and other malware in a timely fashion. It was a common complaint from such individuals that viruses went by too many different names and that their descriptions varied widely between the different anti-virus vendor Web sites. It was also common to see inaccuracies in the virus descriptions, some of which were not corrected for many moons, and often some vital information was missing from those descriptions.

Ken Bechtel and I debated what was important for the end user and came to the conclusion that we should ask others in similar positions what they considered to be important. We met on our way to the VB 2000 conference and agreed to take advantage of that gathering to do some research.

The Cocktail Hour

My memory may be a little hazy, but I'm certain that the initial group of specialists who talked for hours on end during the cocktail reception in Orlando included Jeannette Jarvis and Dean Richardson of *Boeing*, Pete Sherwood and John Morris of *Nortel Networks*, Ken Bechtel and myself. The conversation was wide ranging, but common themes popped up, time and again. Ideas we discussed included the need to get away from the dependence on anti-virus vendors for all our information about viruses, the need to watch each other's backs and to share our wealth of information.

Some objectives for the group emerged:

- It should not be called a 'user group' (Pete Sherwood was adamant about this).
- Access should be restricted to include individuals working in larger organizations only (those responsible for a minimum of 1500 machines).
- No one from the vendor side of the equation should be allowed access to the group.
- We should all work together and keep each other in the loop.

Meeting the first requirement was fairly easy – all we had to do was come up with a name that conveyed the idea without involving the dreaded 'user' word. By the middle of the next day, I had defined a name that I thought was suitable, and run it by some members of the initial group.

Canvassing Opinion

As I met more people at the conference, I asked them what they thought of the concept of a network. The universal response was favourable, and an ever-growing collection of business cards accumulated in my pocket.

During my conference presentation on 'grumpy old men who make life difficult for AV specialists', I announced that a network was being formed and that I would act as contact point. By the end of the conference, the group of people who had given me their contact details was an impressive collection, to say the least.

On arrival back home in Canada, I sent an email to those who had expressed interest in the network and who seemed to meet our basic requirements for membership. Within a few days, and following a number of emails, we had come to an initial agreement about membership requirements, collective goals, and what some of the network's priorities should be. I set up a simple Web page displaying this information and fine-tuned the details to reflect what AVIEN was turning into.

Where is AVIEN Going?

Since those early days, AVIEN has grown considerably and members have established a very solid network.

'Membership in AVIEN has helped our company avoid several virus outbreaks by consistently reporting viruses sometimes hours before the AV vendors have even mentioned it. This allows us to take the proper precautions until the vendors release definitions to detect the virus. Being able to network with a talented group of professionals with the same goals has been a great resource for me. If I am deploying a new AV product or responding to a new viral threat and I have a problem I can turn to AVIEN for advice and support. I think within the next five years being a part of AVIEN will be as essential as anti-virus software itself!' says Travis Abrams, Network Technician at *Holland & Knight LLP*.

Early Warning System

One of the key projects undertaken by AVIEN has been the Early Warning System (EWS). Established in November 2000, it has been instrumental in assisting members a number of times already.

Andrew J. Lee, of the Virus Alerts Response Team at *Dorset County Council* (UK) has this to say about it: 'One of my first experiences of AVIEN was receiving an alert about a problem with a vendor's update file – apparently it was overwriting the boot sector of *NT3.51* servers. At the time we still had a fair number of these servers running as print servers. I was on leave the day the message arrived, and knew that I had set the update to be rolled out that morning. A quick phone call thankfully averted what would have been an absolute disaster. AVIEN paid for itself that morning. Ever since then our AVIEN membership has probably saved the company thousands of man hours and prevented other major disasters.

'It's also important to consider the benefits of peace of mind. It's very hard to judge scale when your vendor warns you of a new virus. You wonder if is it going to be major, or just a flash in the pan. In these situations the benefits of having access to so many other professionals in similar-sized companies to our own are certainly not to be underestimated.'

Other AVIEN members feel the same way: 'AVIEN was the most sensible IT security purchase we made in 2000. For a mere $99.00 we avoided thousands of dollars in malware clean-up costs. Thanks to the warning, we started blocking the *.VB_ extension on the Friday before Anna K. hit. Talk about some good advice!' says Joe Broyles, Computer Maintenance Supervisor at *York County School Division*.

Growing Needs

As AVIEN became more widely known, membership requests began to arrive from many individuals who did not meet the criterion of managing at least 1500 machines. In response to this, subscriptions to the EWS were made available to all interested parties. Like all decisions taken within the Network, AVIEN members debated the pros and cons and voted on this matter.

At the same time, the main mailing list was becoming too cumbersome for some to handle, with multiple topic threads and discussions that were sometimes off-topic to the point of being considered a major distraction. With the approval of the members, an annual fee was instituted to cover the Network's administration costs. *Segura Solutions* assumed the role of administrator and procured the host site for the AVIEN Web site (http://www.avien.org/), and now processes the billing of new members, administers the mailing lists and puts out the empty milk bottles.

A common question about AVIEN is what the EWS alerts look like. The key here is not how they appear, but all the relevant information that they generate. In the words of Robert R. Giberson, Senior Analyst, Global IS Security, *Aventis Pharma*, 'Over the past few months I have been alerted to several virus outbreaks hours prior to an official 'vendor' alert. Not only has the information surrounding the virus itself been provided, but trend information and suggestions from other professionals 'in the trenches' on how to defend further against its infection are often just moments away.

'This is a free exchange of non-biased, factual information from some of the leading professionals in our industry. This information has been applied in the selection and implementation of additional technologies to strengthen our network resources, and build stronger defences by catering to our existing solutions.'

The EWS has garnered a justifiable reputation as being the first out of the gate with information on fast-spreading viruses and worms. All the major malware outbreaks since November 2000 have been spotted by an EWS member and warnings issued at least three hours in advance of vendor warning systems.

What the Future Holds

For the first time since anti-virus software has become an integral part of the security profile of every large organization, there is a forum in which users can discuss this topic freely, without worrying that vendors are lurking nearby, ready to pounce on them with offers of the latest and greatest solution.

AVIEN, already a strong voice, will continue to grow into a presence that the AV vendors must both respect and consider in the development of anti-virus solutions. As Robert Giberson says, 'We are their clients, and now we have a megaphone.'

INSIGHT

Up to the Challenge

Christine Orshesky
i-secure Corporation, USA

I was born in Fort Bragg, North Carolina and, although I was raised in Pennsylvania, I never quite got the South out of my blood – I guess I am a Southerner at heart, with my love of gardens and sweetened iced tea. I grew up in a very rural area, where my nearest neighbour was half a mile away. I attended Octorara Area Schools, on a campus surrounded by farmers' fields.

Then, for a change of scenery, this country girl set off to university at Temple University in the heart of Northern Philadelphia – the City of Brotherly Love. I studied Computer Science, with a specialisation in Criminal Justice. I concentrated on Artificial Intelligence programming courses. While studying, I participated in an internship with the Philadelphia Municipal Court to help prepare an online bail interview system that would use AI to determine bail guidelines. During college I was President, and later Treasurer, of the Temple Student Chapter of the Association for Computing Machinery, where I became interested in computer crime and law enforcement applications.

Enforcing the Law

My first employment after university was with the Federal Bureau of Investigation at their headquarters in Washington, DC. I wanted to be a detective until I learned that they had to use guns and that you couldn't become a detective until you had served your time on a beat. I wasn't convinced that I could handle that, so the next best thing was working for law enforcement (all the excitement – and tedium – without the bullets).

My first true virus response experience was with Michelangelo. Although I had cleaned up a few floppies and files infected with things like Green Caterpillar and Stoned previously, this was the first time I really got involved in the situation and used more than the anti-virus products of the day to resolve the problem.

One of the offices I worked with was suffering from repeated infection of their systems and floppies with Michelangelo. Although it was not clear (prior to its trigger date) how effective Michelangelo would be in carrying out its payload, the media gave the virus sufficient coverage to make many people, including myself, wary of taking any chances. I suspected that there were floppy disks in the office that had not been scanned because, for some reason or another, employees had been reluctant to turn them over.

Although I was a rather junior government employee at the time, my sense of urgency and unwillingness to take chances with this office drove me to take some actions that seemed quite drastic – at least in those days. I quarantined the office, called in extra computer technicians, called in the facility locksmith to open all filing cabinets and desk drawers, and a physical security officer to monitor the entrances. We systematically went to each cubicle (very Dilbertesque), scanning each workstation and confiscating all the floppies for scanning. In all, we confiscated about 1000 floppies, only about 300 of which were infected. Infected floppies were cleaned, when possible, and all floppies were returned. No one was disciplined and no one lost any data, games, screensavers, or other materials on those floppies or systems.

This experience instilled in me very early on in my information security career that it takes more than a software tool to solve a security problem. It takes an understanding of the business needs, the people, and the operational environment to manage an incident – anything less is only a bandage.

I worked at the FBI for over four years, performing many different duties including the development and maintenance of large extensive macros in *WordPerfect* to generate automated government forms, which were really just a combination of recorded keystrokes and printer commands – a far cry from the macro viruses of today. I also worked with their technical support centre and assisted with the development and testing of a standardized desktop platform. As I made the transition to the information security department, I returned to my Artificial Intelligence schooling and worked on an intrusion detection project using anomaly detection and rule-based components – a precursor for the commercial products on the market today. Before I left the FBI, I became the Quality Assurance Project Manager for their mainframe application development and integrated the quality assurance principles into their system development lifecycle.

While I enjoyed much about my time at the FBI, and valued the acquaintances and friendships I made there, after four years I felt it was time for a change and some new challenges.

Challenges

One of my most challenging roles was at the Pentagon (not the one in Abingdon, England, but the one in the large building just outside Washington, DC), where I worked as Virus Response Manager. There were lots of systems, lots of people and lots of politics – power and turf issues to be more precise. I worked for an organization that supplied the computing backbone for many of the military services housed in the Pentagon – each of which had their own regulations, rules, standards, philosophies and agendas. To say I became quickly schooled in diplomacy and compromise is probably an understatement. Yet, through my tenure there, I managed somehow to remain military-illiterate and, to this day, I cannot reliably tell one rank from another, particularly between services. (I found Sir and Ma'am kept me out of trouble most of the time.)

It was the challenges of the position at the Pentagon, and the lessons I learned there, that gave me the courage and passion to take my next step – I founded a small company to address anti-virus issues as I saw them. I wanted to stress the importance of education and the use of more than anti-virus products in handling virus-related issues. I wanted to help people manage virus-related situations instead of being managed by them. I specialize in the policies, processes, procedures, products and education of incident management (not just incident response), with a focus on malware-related incidents. It seems a perfect fit, given my experiences and my beliefs in a holistic (please, excuse this now overused term) approach.

Most of the time, I work on getting business, performing tasks, developing training materials, and speaking as often as possible. I feel very passionately about education, particularly where information security is concerned, and I enjoy the opportunities that teaching provides to meet new people, new challenges, and new ideas. And, hey, I like to talk, so I might as well put it to some useful purpose.

The Future

I suspect the future of the AV industry is rather good, so long as it continues to evolve with the threats. It seems we will always need something to protect our computers and the data they contain from malicious or accidental damage, compromise, or modification.

The future for virus writers looks fairly healthy if you look solely at the production of malicious code and the opportunities for profit – we are doing more with less, from a technological perspective, and there is more prolific use of those technologies. The laws and the tools to combat the threats are evolving as well, though, so it may not be as easy for virus writers to release viruses without personal repercussions. However, I am not convinced that pursuit and punitive endeavours will have a completely deterrent effect. We still have homicides, we still have burglary, and we still have a variety of other offences even though we have had laws and punishments against these offences for centuries. I think it takes education, specifically in ethics and appropriate use of technology, to combat this problem.

I think we have made great strides in many ways, but we are still lacking an overall appreciation for the scope of the problem. The fact that so many malware threats rely on virtually the same ploys to spread, and the fact that they do spread, says a great deal for the number of lessons we have ignored or the ways we could do things better. People continue to open email attachments sent by others both known and unknown to them. I keep returning to the fundamental educational issues: if people understood the threats and their role in combating those threats, I suspect we might be a little better off. Instead, as an industry, AV projects the view that the user doesn't need to understand and in some cases is unable to understand – and thus the saga continues.

In terms of AV methods, I prefer 'defence in depth' strategies, where protection is installed at as many entry points as possible, and more than one anti-virus product is used, so all of your eggs are not in the one proverbial basket. As you might suspect, I also strongly support user awareness, for all levels of users from executives to clerks.

I dislike products and solutions that prevent me from protecting my environment in the way I want to and from obtaining the information I need to manage an incident. There is no single answer for every network and every organization. I am put off by product vendors and anti-virus developers/researchers telling me what I need when they do not understand my business needs and may never have handled a virus incident in an operational environment.

Home Life

About four years ago, I met the then editor of *Virus Bulletin*, Ian Whalley, and through coincidence or engineering we went on our first date in Washington, DC during Clinton's second inauguration – a bitterly cold weekend in mid-January. We have been inseparable since – well except for the large ocean and many miles that remained between us for over three years. I happily earned many frequent flyer miles travelling to England for visits, merrily enduring the grilling by passport control when I stated I was going to Abingdon for pleasure, and felt a bit like a jet setter – frequently popping over for long weekends. Ah, those days, while full of many pleasant memories, are gratefully past us, and we now share a home in 'Up State' New York with our two cats, Boots and Danni, aka 'Splidge and Splodge'.

I would like, someday, to put this career to a different use – possibly pursue a law degree so I can translate between those who make the laws regarding technical issues and those who have to enforce or comply with those laws. I also have a dream for my 'retirement' in which I would be back in a rural setting, out in the middle of nowhere, maybe hosting a bed and breakfast or maybe just enjoying the passage of time, but I expect that will be many years in the future.

CONFERENCE REPORT

Viva Las Vegas

Matt Ham

At first, Las Vegas seemed an odd choice of location for a conference, let alone two in short succession, and arriving in the town for the Black Hat Briefings and DEFCON 9 gave me little reason to change my mind.

Las Vegas appears to be devoted exclusively to relieving visitors of their hard-earned cash. It is also a city far from the beaten track, even by US standards, making it seem almost as if the conference organisers intended to discourage attendance. Unlikely though it might seem, this is not too far from the truth. Of the two conferences, DEFCON is the older and started as a small meeting of hackers who were, indeed, meant to be discouraged by the location – as a form of 'natural selection'. The two conferences share their creator and organiser in Jeff Moss, and thus shared Las Vegas as a location on consecutive days.

What's it All About?

History lessons apart, what are the target audiences for these conferences, and what relevance do the two have to the anti-virus industry, since they seem to have much stronger links to the general security field?

Security holes are becoming increasingly relevant on *Windows* systems, as traditional routes of propagation are gradually made more difficult to use. For example, *Office* macros were once wide open to abuse, but these opportunities are diminishing with each new *Office* suite. Where once all that was needed was to write a macro, now the additional matter of circumventing the inbuilt security must be taken into consideration by virus writers.

The increasing number of common virus host platforms is also one that makes security more relevant, since these platforms are mostly unix-based and have good provision for operating system security. On a well configured *Solaris* box the only worm that could thrive would be one such as Solaris.Sadmind – that is one which uses unintentional bug- or exploit-related security holes in its propagation.

Black Hat Briefings

The Black Hat Briefings are first for inspection. The Briefings target an audience based firmly in the computer security professional arena, and were held in the splendour of Caesar's Palace (hotel).

Information presented ranged from general overviews of, for example, WAP and SMS security holes, to the much more esoteric executable editing for the removal of potential buffer overflow attacks.

The subject matter of talks could, generally, be assumed to be Unix-derived systems in those cases where the operating system was not specified, although *SQL Server*, *Windows* and *Lotus Domino* were included among the list of those platforms inspected.

Only one session was devoted wholeheartedly and exclusively to *Virus Bulletin* territory: 'The Future of Internet Worms' by José Nazario of *Crimelabs*. As mentioned, Unix-based operating systems were the most common subject matter at the conference and this presentation was no exception.

In contrast with the more *Windows* worm-dominated real world, this led to something of a different view of the future potential of worms. The major difference was in the consideration that worm hosts are not as easy to come by on a Unix-based system as in a *Windows* environment: with machines on which security is more likely to be implemented, and where the sheer volume of potential hosts is significantly smaller, there is a great need for subtlety by worms in a Unix environment.

This subtlety was proposed to add steps to the usual *Windows* worm activity, which can currently be summed up as finding other hosts (regardless of whether they are infectable), attempting to infect and in most cases ignoring the results. The more complex potential future worms discussed borrowed heavily from current DDoS technology, both having an additional phase in which possible hosts are detected and selected, and remaining in contact with other affected machines after propagation has been successful.

Another fundamental difference was in the countermeasures we might expect these worms to encounter – the emphasis being on avoiding Intrusion Detection Systems (IDS), rather than direct scanning for worm files. As a relatively high and variable volume of traffic is likely to achieve the easiest circumvention of traffic-based IDS, mail and newsnet were seen as good channels by which worm code could enter a machine. Worms such as W32Hybris can be regarded as the vanguard of this theoretical new range of worms.

Meeting and Greeting

The remaining presentations were variable in their direct relevance to anti-virus specialists, though many had a more indirect anti-viral aspect. Such matters as how security holes might be taken advantage of by viruses or worms in the future, or how security measures could be used to limit the effects of such malware frequently entered the discussion. As always, there were ample opportunities to meet other delegates and gossip about personal experiences during the scheduled breaks. During one of these breaks I was privileged to meet perhaps one of the most unlucky worm victims.

The administrator in question had on-access scanning on workstations, scheduled updates every night and a good level of staff awareness as to what might be a suspicious file – what could possibly go wrong?

First, the staggered update failed to prevent a time-out on a signature update, and a worm slipped through which would otherwise have been detected by the update. The workstation in question belonged to the company's graphic designer, who was waiting for artwork from the printing company. The printing company was infected by the undetected worm, the subject being one of the innumerable 'pictures for you' types. The result of such a combination need not be related.

DEFCON 9

After the air-conditioned consumer's paradise that was Caesar's Palace, the setting for DEFCON was somewhat low key, though this came as something of a relief after two days of toga-clad hotel workers. As mentioned previously, DEFCON started its life as a purely hacker meeting event, and has undergone a good deal of evolution in the meantime.

The audience is not targeted as such (in fact the conference is not advertised at all), but does fall into several categories: the professional seeking to know his enemy, the established hacker meeting his friends and the newer hacker trying to make those friends. In practice, those in the professional category often fall into one of the other two categories as well since, unlike virus writers, hackers are often the best people to hire for protecting against other hackers. (Quite which category Richard Ford, ex-*VB* editor and DEFCON attendee, falls into I will leave as an exercise for the reader.)

Presentations

Again, the general tone of the presentations varied between the technically simple to the obscure, though many were devoted to related topics such as the legal and political implications of hacking.

Of note was the frequency with which disclaimers and warnings preceded the main body of the speeches. As an aside, one interesting fact learned from both of the conferences is that lawyers, and those who have defended themselves successfully in court, make very good speakers, even with potentially dry subject matter.

As to the relevance of subject matter addressed at DEFCON, the presentations were, in general, slightly less relevant to anti-virus matters than the Black Hat Briefings, though this was accounted for by the greater number of non-technical themes addressed.

One of the more promising malware-related speakers did not appear, leaving 'An Open Source, International, Attenuated Computer Virus', by Dr Cyrus Peikari of *VirusMD Corporation* as the presentation of most interest to *Virus Bulletin* readers. The presentation began as a demonstration as to how, in nature, biological viruses are not combated as effectively by disinfection or barrier methods as they are by immunisation, and Dr Peikari sought to transfer the analogy fully to computer viruses.

There was a distinct realisation that such notables as Vesselin Bontchev have addressed 'good' computer viruses in the past and their objections were rather briefly handled. Audience opinion was notably mixed as to how well Dr Peikari's analogy worked and what methods could be employed which users would find acceptable.

Speeches aside, DEFCON does have a large social aspect, which is much more party- than work-oriented. Most organised events involved alcohol – and quickly became less organised. By the very nature of the audience, there were many cliques and many attendees who preferred to remain silent about their more impressive activities. This, and the greater number of attendees, tended to make networking more of a hit-and-miss affair than at the Black Hat Briefings.

Summary

Both conferences were, overall, interesting – but how relevant are they to anti-virus specialists? If current trends continue and viruses become both more security-aware and more diverse in their platforms, the answer is perhaps that these conferences are not so relevant today, but will become increasingly relevant in the future.

Of the two, the Black Hat Briefings are more applicable to the AV specialist and provide better hard copy information (though both the low cost and convenience of attending DEFCON might well warrant attendance at both conferences, especially given the excitement afforded by the arrest of one of the speakers at DEFCON this year).

This year at least two anti-virus developers were represented at one or other of the conferences and AV presence may well increase in the future.

Conference information:

Black Hat Briefings: More information, including details of upcoming events, is available at the Web site http://www.blackhat.com/.

DEFCON: Information can be found at the Web site http://www.defcon.org/ or by sending an email to questions@defcon.org.

PRODUCT REVIEW

F-Prot Antivirus 3.10 for Win95/98/ME

Matt Ham

It was notable in the last DOS comparative (see *VB*, July 2001) that three of the products under review were based on the *F-Prot* engine. This engine is produced by a developer which is, by current standards, of very modest size, and located almost at the end of the earth in Iceland. Despite this, not only the developers of the products reviewed last month, *Command* and *F-Secure*, but also the lesser known *perComp* use the *F-Prot* engine in their scanners. The question then springs to mind as to what the engine's developers themselves can make of the product's capabilities. This review commences with every reason to expect good things. There is also a second question to be answered: why should this product be chosen over those that are based on the same engine? This is somewhat more subjective and will only be covered in passing, to allow the reader to make their own judgement.

The Package

The product was supplied in electronic format, a total zipped package of just over 5 MB. Unzipping and running the setup program launches the ever-popular InstallShield, the first choice being the option to accept or decline the License Agreement, a document which holds no surprises.

The next option is the destination for the program install, and the choice of a Typical, Compact or Custom install. The choice of installation type is the usual InstallShield fare, though in this case 'Typical' and 'Compact' give no indication of what will be installed and, in fact, display exactly the same immediate behaviour.

Custom install presents the option to select the installation of the OnDemand Scanner, Integrity Checker, DOS Scanner, Scheduler, Updater and RealTime Protector. Of these, only the OnDemand Scanner is a mandatory selection. Later information screens show that the Typical install selects all of these modules for installation, while the Compact install provides only the OnDemand and DOS Scanners. For the purposes of this review, the Typical install was chosen.

Following the selection of the program group for the installation, a summary screen of selected options is displayed, which includes the details of which exact modules will be installed. Installation can now proceed, with a final choice after this as to whether to integrate the scanner within the *Windows* shell and whether to spawn an immediate scan. The former option was selected for review purposes.

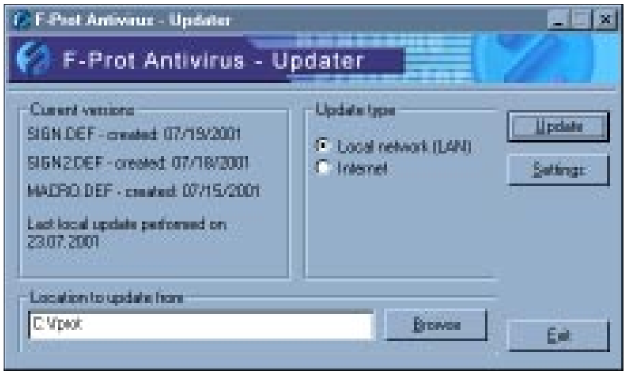
Full installation of the product leads to the installation of the following as items in the start menu: Scan, OnDemand Scanner, Integrity Checker, Scheduler, Updater, Uninstall and RealTime Protector. All of these, with the exception of Scan, will be considered later. Scan is a subfolder which offers shortcuts to preset scan jobs of CD-ROM, floppy drives and hard drives. By now it will be obvious that nothing but the program files have been considered, and this is because these were the only files provided in the package. Help files as a replacement for documentation are available as part of the software.

Web Support and Updates

The *F-Prot* Web site (http://www.frisk.is/) is quite small, but as a result it is easy to navigate. It also underwent a complete revamp during the process of this review. The support area was the most disappointing part of this revamp. It contains FAQs, Technotes and information related to retailers and sales, but this had not been updated past the 3.09a version of the software at the time of writing. However, the lack of current information was acknowledged on the Web site as being in the process of being rectified, and users were directed to use the support email address for immediate information.

The download area was much more edifying, giving access not only to update files for the *F-Prot* product range, but also trial versions of the *Windows* software or the free private user version of the DOS scanner (as reviewed in *VB*, July 2001). The virus information area was a big improvement over previous incarnations, with information on such topical issues as W32/SirCam and Code Red. There were also links to the online virus libraries of *F-Secure*, *Command* and *perComp*, which have more comprehensive malware listings. This area was listed as 'under construction', which might explain the existence of a confusing link to http://www.complex.is/, which seemed to be nothing other than an exact mirror of the http://www.frisk.is/ site.

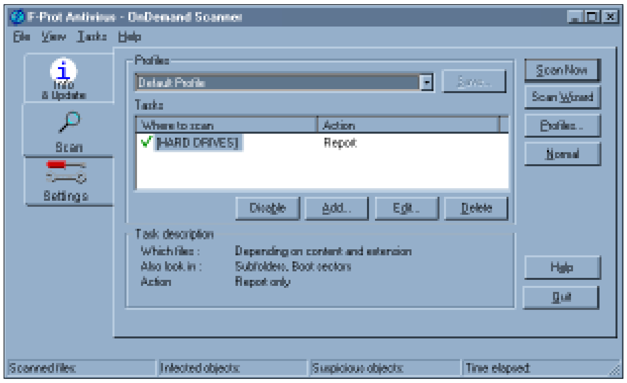
In addition to the direct file download available from the site, the Updater can be used to keep the virus definition



Current versions
SIGN.DEF - created: 07/19/2001
SIGN2.DEF - created: 07/18/2001
MACRO.DEF - created: 07/15/2001
Last local update performed on 23.07.2001

Update type
● Local network (LAN)
○ Internet

Location to update from
C:\fprot



Default Profile

Tasks

Where to scan	Action
✓ [HARD DRIVES]	Report

Task description

Which files :	Depending on content and extension
Also look in :	Subfolders, Boot sectors
Action	Report only

Scanned files: | Infected objects: | Suspicious objects: | Time elapsed:

files up to date. The standard commercial licence for *F-Prot Antivirus* includes a one-year duration of electronic virus definition updates, though updates on CD are available quarterly for an additional charge of US$100. Initially, the automatic update facility for the program did not seem to work particularly well (a matter which is covered in more depth later in the review).

Features

Back to the program itself, or rather its components which are spread between the five applications installed to the start menu: the OnDemand Scanner, Integrity Checker, Scheduler, Updater and RealTime Protector.

OnDemand Scanner

The OnDemand Scanner opens with a page of great interest to poor folk such as me, who are all too familiar with the claim 'so many thousands of viruses are detected by product X'. In the case of *F-Prot Antivirus* the detected files are broken down into sensible groupings, with exact figures (as much as these numbers can ever be considered exact) in each category. The categories include such esoteric names as PalmOS, and the catch-all of Batch/Others which inspire evil thoughts as to the contents of updated *VB* test sets. On a more useful and interesting point for the average user, this area also details when the last updates were made to virus definition files for *F-Prot Antivirus*. From this page the Updater, RealTime Protector, Scheduler and Integrity Checker can be launched.

As a slightly unusual option, the OnDemand Scanner can be selected from its own introduction screen. This screen is, in fact, one tab of three which make up the interface of the OnDemand Scanner. The first tab screen to be seen is labelled Info and Update. Selecting OnDemand Scanner from that tab changes the tab view to Scan. The remaining tab controls Settings.

The Scan tab in its simplest form has an area where objects or folders can be manually entered or browsed to and a single click allows scanning of the selected area. To this basic functionality are added Scan Wizard, Options and Advanced buttons. Options controls which objects should be scanned as far as archives, compressed files and boot sectors are concerned, as well as which extensions should be scanned. The finesse of the scan process is controllable, with the options being by extension, by content and extension, or dumb. The default scans boot sectors and inside subdirectories, uses content to determine when to scan, and uses an extension list rather than all files. Further options dictate what action should be taken on virus discovery, with the standard setting being a pure report, while the standard prompts for action, disinfect, quarantine, delete, rename and move are also supported.

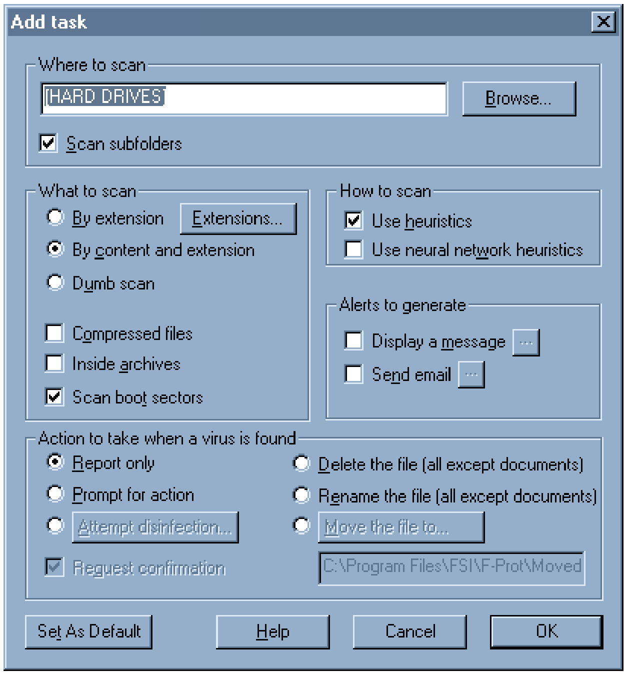
It is of note that document files are specifically excluded from the delete and rename options, even when these are selected. This reflects that these files are substantially more reliably disinfected. As is often the case with such features, it becomes obvious that there are even more possible tweaks which might make life simpler. In cases where a virus is known to make alterations to documents, further discrimination so as to allow documents only to be quarantined, or even better quarantined according to virus found would be useful. With the associated problems and complexity involved in implementation, however, this does not seem a likely course of action by any anti-virus developer.

The Advanced option completely changes the interface, adding a Profiles button and a range of buttons allowing jobs in these profiles to be enabled or disabled, added, edited and/or deleted. The scan jobs are the building blocks of Profile, which is a little different from the usual parlance of anti-virus software.

Given these options, how does the Scan Wizard fit into the grand scheme? Not surprisingly, this is used to create the scan jobs. This commences in much the same order as discussed above, with a choice of area to be scanned and then dealing with extensions to be scanned, method of scan and in fact all but the response options detailed above. The third page of selections adds some new ground, however, as either 'heuristics' or 'neural network heuristics' may be selected for use. These appear to be independent features and can be selected or deselected in any combination. At this point there is no description of how the two heuristic methods differ, so the decision as to which to use is left mainly to guesswork.

The response to infection section is next, coinciding exactly with the response options detailed above. There are options available as to whether to display messages and/or send emails if an infection is detected. At this point the Wizard process is complete and can be used immediately to scan or used to scan later. The scan may also be saved as a job under a specified name so that it may be selected later.

This order of operations describes the creation of a job under the Advanced interface. The Normal interface dispenses with the heuristics and mail/messaging options and does not allow the configuration to be saved into a job or profile. Jobs can also be created directly from the Advanced interface, all choices being available from one complex page of choices. Oddly enough, neural network heuristics are not mentioned in help, though they are



Add task

Where to scan

[HARD DRIVES]

☑ Scan subfolders

What to scan

○ By extension

● By content and extension

○ Dumb scan

☐ Compressed files

☐ Inside archives

☑ Scan boot sectors

How to scan

☑ Use heuristics

☐ Use neural network heuristics

Alerts to generate

☐ Display a message

☐ Send email

Action to take when a virus is found

● Report only

○ Prompt for action

○ Attempt disinfection...

☑ Request confirmation

○ Delete the file (all except documents)

○ Rename the file (all except documents)

○ Move the file to...

C:\Program Files\FSI\F-Prot\Moved

available as an option. One minor irritation is that the profiles are not available from the scan subfolder of the start menu, which would be a welcome feature, even more so if a subset could be added to this location rather than all profiles.

This dispenses with the second of the three OnDemand Scanner tabs and the Settings tab remains. This is the location of various controls which would not find a happy home elsewhere. Here, your preferred language can be selected (though English was the only language available in the copy reviewed), sounds can be activated in virus detection and email server and address settings can be adjusted for use in the messaging function when viruses are detected.

The Settings tab is also the area in which the report file is controlled, options for which include location, reporting all objects and whether the file is overwritten or appended. The log file may be viewed from here, though as it is in pure text format this is really a luxury not a necessity.

In addition to the three tabs there are also File, View and Help drop-down menus. Help offers standard *Windows*-style help functions, which in this case are usually images of the appropriate tab or dialog with hyperlinks to more detailed information on the area selected.

There are some oddities available if a full help index is selected. For example, selecting the topic ‘Eicar test file’ brings up a page which has no picture but an exhortation to click on this non-existent picture. Other than oddities of this type, help was generally useful and well written.

The View menu generally allows movement between the three tabs, though when in the Scan tab it adds a few new functions. Advanced and Normal mode may be swapped between and the Report and Last scan results may be viewed if so required. The Scan tab also adds a new menu, Tasks, which allows another method of task manipulation.

Most varied of all the menus is the File menu. On the Info & Update tab the File menu offers an update option, while on the Settings tab it is used to exit the program. Meanwhile, on the Scan tab this menu can be used to activate scans, invoke the Scan Wizard, save or manage profiles or exit the program.

The Others

The OnDemand Scanner covers by far the bulk of the software options, though the other parts are in some cases more important. The functions of a Scheduler, Uninstall and Integrity Checker hold no great surprises for most readers. The two remaining programs of interest are thus the RealTime Protector and the Updater.

The Protector, the on-access scanning component, is remarkable in that it comes with relatively few options. Areas protected can be set to all files, documents or ‘known file types’, which presumably refers to the OnDemand extension list. The default action is to deny access to files, though this can be deactivated, and floppies may be checked on access and/or shutdown. The remaining option is whether to show the icon on the taskbar which, in the absence of a help function, leaves a few questions about exactly what behaviour is expected of the on-access scanner where, for example, archives are concerned.

The Updater has been touched upon earlier in this review, where it was stated that there were initial problems with the update process. This turned out to be a minor problem with attempting to update without restarting the machine in the interim. Once the machine had been restarted updating was invisible and automatic. Of more interest was the reaction by *Frisk Software International*. The problem was dealt with quickly, sensible questions about my actions were asked, and a patched version of the program was available within 24 hours. In a manner designed to endear itself to me the technical support staff of the company were definitely faster acting than the salesmen – always a good sign.

Scanning and Speed Tests

The detection rate of the 3.09a version of this program was tested only last month, so it seemed likely that there would be no great surprises, though the difference in platform had potential for some disastrous effects as has been seen in the past with other products. Equally interesting, however, was the mystery of the on-access scanner, and quite what this uses as its source for configuration. Tests here were performed on a *Windows ME* machine.

The scanner was tested by copying and unzipping large archives of infected files from CD to hard drive and within folders on the hard drive. The on-access scanner does not have any controls related to the scanning either of archives or compressed files, so any change in behaviour might show that the on-access scanner inherits its settings from the on-demand scanner. The file copying was performed within *Windows* with the default OnDemand scanner profile first set not to scan within archives and compressed files. This was then compared to the behaviour with archive and file scanning on. As a control the file was also scanned from the on-demand scanner in both cases.

The results were that, although on-demand scanning showed detection within a .ZIP archive when archive scanning was activated, the on-access scanner did not demonstrate any ability to detect within the same archive. This suggests that the on-access scanner uses its own settings, independent of the on-demand scanner, and is thus not as highly configurable as it might be.

What is more, the archive could be extracted in a DOS box, through pkunzip.exe, without any sign of infection being noted. The same process performed by WinZip, on the other hand, detected the contents as viral as the files were unzipped. Further tests copying and performing file opens on infected objects within a DOS box confirmed that scanning was not active under this environment – at least for those operations performed.

As for the standard *VB* test sets, those used were identical with the July 2001 DOS comparative. A preliminary scan of the set quickly showed one useful feature in that the scan parameters are summarised on the GUI whenever a scan job is selected or active, thus it easy to see exactly which scan was being performed. The first scan detected all files in the *VB* test set – labelling 125 of these as suspicious, these presumably being detected through heuristics.

This presumption was checked by running the same scan without the benefit of heuristics. As might be expected, the number of exact detections remained the same. Oddly enough, though, this test did result in the presence of some ‘suspicious objects’, which were examined to check whether a pattern could be discerned. The ‘suspicious’ tag, it turns out, is not reserved simply for heuristics but has other targets. These include those programs determined to be destructive or a security risk, an example being the .EXE portion of JS/Unicle. In a slightly more debatable fashion, W32/Blebla and W32/Msinit both fell into this category too. Using the neural network heuristics gave a result identical to that without the use of heuristics, so the mystery of its inclusion deepens.

The dumb and extensions-only scan settings were tested for detection next. The result of the extensions-only scan showed one thing by being identical to the default scan – that the extension list is chosen well. The dumb scan, scanning all the files, was expected to result in exactly the same detection results and did so.

For checking the impact of settings upon the speed of scanning the infected set is not a very likely real world scenario, so the *VB* clean set was used. As a baseline the clean set was scanned using the default settings for the scanner. This was easy to revert to as, without explicitly setting default values, only the scan jobs themselves were saved between scanning sessions.

The default scan took a total of 220 seconds, in comparison with a dumb scan at 216 seconds and an extensions-only scan at 214 seconds. Since the extensions in the clean set are all infectable types this was not really expected to show any great variation.

As a less scientific but more ‘real’ method of judging overheads from these options, the same scans were performed on the C: drive of the test machine, giving figures of 23 seconds as a baseline, compared with 20 seconds for by extension and 43 seconds for a dumb scan. This is more like what might be expected, with simple use of an extension list requiring fewer resources than judging file types, while scanning even uninfectable objects will simply waste time.

More relevant were the tests including the heuristics, traditionally a cause of engine slowdown. Deactivating heuristics took the time for a scan of the clean set down from 220 seconds to 200 seconds, while activating neural network heuristics took the time up to 222 seconds. Activating the two varieties of heuristic simultaneously left the time at 220 seconds – a result that is odd indeed.

Conclusion

With only one minor technical problem, addressed quickly by dint of *Frisk* being the developer, no real criticism can be directed at this product within *Windows* (the problems in the DOS box having been fixed since this review.) The primary weakness, as is common in electronically distributed software, is that of documentation which is missing entirely in hard copy format. This is a problem which is mitigated by the good inline help, but it remains sketchy in some areas. As for decidedly good features, there are many; the technical support and intuitiveness of the software are both very good and detection rates go from strength to strength.

Technical Details

Product: *F-Prot Antivirus 3.10 for Win95/98/ME*

Developer: *Frisk Software International*, P.O. Box 7180, IS-127 Reykjavik, Iceland; Tel +354 561 7273; Fax +354 561 7274; WWW http://www.frisk.is/; email sales@frisk.is.

Price: For commercial use $2 per installation for up to 2500 computers, subject to a minimum cost of $40.
For larger orders, educational and private use rates see http://www.frisk.is/f-prot/products.

Test Environment: 750 MHz AMD Duron workstations with 128 MB RAM, 8 and 4 GB hard disk, CD-ROM, LS120 and 3.5-inch floppy, running *Windows ME*. Pentium II laptop with 48 MB RAM 1 GB hard disk, CD-ROM and 3.5-inch floppy, running *Windows 98*. Celeron workstation with 256 MB RAM, 30 GB hard disk, CD-ROM and 3.5-inch floppy, running *Windows 95*.

Virus Test Sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/DOS/2001/05test_sets.html.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park,
Abingdon, Oxfordshire, OX14 3YP, England

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

US subscriptions only:

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

Tel (781) 2139066, Fax (781) 2139067

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

END NOTES AND NEWS

i-Security 2001 takes place on 6 and 7 September 2001 at the Putra World Trade Centre, Kuala Lumpur, Malaysia. For further information tel +60 3 21696228 or send an email to sfrc@tm.net.my.

COSAC, the 8th International Computer Security Symposium takes place 9–13 September 2001 at the Hotel Dunloe Castle, Killarny, Co. Kerry, Ireland. For full programme information and details of how to register visit http://www.cosac.net/.

ISSE 2001 Information Security Solutions Europe takes place 26–28 September 2001 at the QEII Conference Centre, London. Leading technologists, heads of industry and legal professionals will present the most recent security concerns and solutions in over 70 sessions. For further information email isse@eema.org, tel +44 1386 793028 or visit the Web site http://www.eema.org/isse/.

Reserve your place now for the 11th International Virus Bulletin Conference & Exhibition (VB2001) on 27 and 28 September 2001 at the Hilton Prague. To benefit from the special *VB* subscriber rates contact Bernadette Disborough; tel +44 1235 544034 or visit the *Virus Bulletin* Web site http://www.virusbtn.com/vb2001/ for a booking form and more details.

Information Security World Africa 2001 will be held 3–5 October 2001 in Johannesburg, South Africa. For further information visit the Web site http://www.terrapin.co.za/event/E839/.

COMPSEC 2001 will take place from 17–19 October 2001 at the Queen Elizabeth Conference Centre, London, UK. For more details about the 18th world conference on computer security, audit and control, visit the Web site http://www.compsec2001.com/ or contact Tracy Collier: tel +44 1865 843297; email t.collier@elsevier.co.uk.

Internet Security runs from 23–25 October 2001 at ExCel, London, UK. For more details contact Andy Kiwankua: tel +44 20 8232 1600 ext. 246, email andy.kiwanuka@pentoneurope.com, or visit the Web site http://www.internetsecurity2001.com/.

The Black Hat Briefings and Training Europe take place in Amsterdam this autumn. Training runs from 19–20 November and Briefings from 21–22 November. For more information visit http://www.blackhat.com/.

The 4th Anti-Virus Asia Researchers (AVAR) Conference takes place on 4 and 5 December 2001 at the New World Renaissance Hotel, Hong Kong. For full details about the conference see the Web site http://www.aavar.org/.

Australian anti-virus company ***Leprechaun Software* has integrated the scan engine developed by *VirusBuster Ltd*** into its products. By coincidence, *Leprechaun*'s leading product is named *VirusBUSTER*.

***McAfee*'s *VirusScan Wireless* product will provide anti-virus support** for *Nokia*'s Internet-connected wireless devices, the 9210 and 9290 Communicators. *McAfee*'s 'micro engine' will examine files and applications for viruses and malicious code locally on the 9210 and 9290. The Communicators' Internet connection capability will enable the automatic update of virus definition files. For more information visit the Web site http://www.nai.com/.

Kaspersky Labs* has released the latest version of *Kaspersky Anti-Virus* for Unix/*Linux operating systems. The new version allows installation of a centralised AV defence for file servers and application servers operating on *OpenBSD* (version 2.8) and *Solaris 8* (for Intel processors) systems, as well as for exim e-mail gateways. For more details see http://www.kaspersky.com/.

***Sybari Software* has released *Antigen 6.0* for *Domino* servers**, designed specifically to meet the anti-virus and security needs of *Domino*/*Notes* administrators. *Antigen 6.0* can be downloaded from the *Sybari* Web site http://www.sybari.com/.

Global services company *EDS* is to re-sell *F-Secure* products and provide integration, security management and support services to *F-Secure* customers worldwide. In addition, *F-Secure* will provide security content for *EDS*' Cyber Security Institute, a computer security curriculum to arm IT professionals and consumers with skills to battle against hackers, security breaches and viruses. See http://www.f-secure.com/.

Trend Micro* has entered into an agreement with UK WASP *myWasp to provide content security for *myWasp*'s Roaming wireless application service. This alliance will enable *myWasp* to utilise *Trend Micro*'s InterScan VirusWall technology in its wireless applications solutions. For more see http://www.trendmicro.com/.