ISSN 0956-9979

FEBRUARY 2002

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Three's company!** This month *VB* takes an in-depth look at three viruses. Gabor Kiss analyses the SWF/LFM-926 proof-of-concept virus on p.6. Starting on p.7, Costin Raiu investigates RST, an example of a *Linux* virus that infects local files also embedded with backdoor capabilities. Finally, Peters Ferrie and Ször explain why Badtrans.B became so widespread so suddenly at the end of last year, see p.8.

• **Lab work:** Andreas Marx has spent his fair share of time in test labs and knows only too well the familiar problem of preparing multiple computers in a test lab. He explains how the PCs in his lab were set up with *Windows*-based test environments, starting on p.11.

• **Let's talk about ME:** Twelve months after his last look at AV products for *Windows ME,* Matt Ham put them through their paces again. Find out how many of the 18 contenders earned a VB 100% award on p.15.

CONTENTS

COMMENT

> “ *Recent events suggest that one of the industry’s longest-standing stupidities will remain with us for some time to come.* ”

Generic AV Problems

There are, it seems, a few things you can count on from the AV industry – detection of viruses, regular update releases, the occasional contribution to an unnecessary media flap over some non-event with some non-entity of a virus, and so on. Since we pretty much take them for granted, we tend to ‘accept’ these events – even the rather negative ones – as par for the course. Fortunately, social and market pressure (and the effects of crying ‘wolf’) are gradually making themselves felt and, within a few years, we may yet have a fully-matured AV industry, where the unsavoury memories of hype and FUD are but a fading, albeit vaguely bitter, taste in the back of the industry’s collective throat.

Unfortunately though, recent events suggest that one of the industry’s longest-standing stupidities will remain with us for some time to come. Regular readers of *VB* over the past few years will be aware of my views of the grievous design ineptitude of those behind products that report whatever malware they find in a manner such as ‘*<filename>* is infected with the *<malware_name>* virus’. That some products are still hard-coded to produce such reports ten years and more since their developers introduced detection of malware other than viruses, is dumbfounding.

What recent events have prompted my gloomy outlook? In a nutshell, it was the mad dash to add ‘vulnerability exploit detection’ to virus (well, malware) scanners. Such exploit detection is not inherently a bad thing, but that is no reason to implement and deploy it badly. The rush to add exploit detection to virus scanners has obvious drivers. First, as many commentators predicted, malware writers have increased their interest in exploiting security vulnerabilities as a means of infiltrating their code into target machines. Second, the potential victims of malware have demonstrated that they are at least as resistant to patching their machines in a timely manner as they were to learning ‘thou shalt not double-click unexpected e-mail attachments’. A third factor may also be important – it seems that the Internet may be inhabited by corporate interests of a more dubious nature than many expected, with many ‘corporate’ sites trying to cash in by using common vulnerability exposures to promote themselves and/or their clients. But why do these factors motivate implementation of exploit detection in ‘virus’ scanners? Individually or together they do not, but the nature of much exploit detection is that it requires generic detection capabilities. Over the last couple of years several developers seem to have been quite successful at improving their products’ ‘generic detection’ capabilities without significantly worsening their false positive rates. As these developments were largely independent of a perceived need to add exploit detection to their products, it was rather serendipitous that another application for good generic detection capabilities should come along.

The vulnerability exploits that have been of most interest to virus scanner developers have been a small handful taking advantage of security flaws in scripting and/or ActiveX components in *Internet Explorer*. These vulnerabilities have been widely exploited by viruses and, rather surprisingly in the extent if not the brazenness of it, in simple Trojan scripts that alter one or more of *IE’s* start page, home page and default search page settings. It should also be noted that this latter type of exploit could be used as a delivery mechanism for much more troublesome payloads and, at least to me, it seems rather odd that we have not seen much of this kind of use of such exploits yet. Of course, once a couple of the ‘big name’ scanners started to detect ‘JS/Exploit’, ‘JS.Exception.Exploit’ and the like, the pressure increased on other developers. So what happened? Several vendors added some form of generic detection (‘sloppy string scanning’ sounds so crude!) of code attempting to exploit these commonly targeted vulnerabilities but ignored the fact that they were implementing detection of a whole new class of ‘malware’. Thus, even products that had finally beaten the ‘all that we detect shall be reported as a “virus”’ blues, have started reporting inanities such as ‘index.html contains the JS.Exception.Exploit virus’.

The more things change, the more they stay the same, eh?

Nick FitzGerald, Computer Virus Consulting Ltd., New Zealand

NEWS

Baltimore Shedding Content

UK-based secure content management software company *Clearswift Corporation* has offered $30 million for the *Content Technologies* subsidiary of *Baltimore Technologies Plc*. The subsidiary, which sells the *MIMEsweeper* family of content security products, was purchased by *Baltimore* for $992 million less than two years ago. Following a review in August 2001, *Baltimore* was forced to admit that it was 'operating two businesses with limited synergies between them'. *Baltimore* will now focus on its core business of supplying and marketing authentication and authorization solutions. The sale is subject to *Baltimore* shareholders' approval ■

Challenging Beliefs

The commonly-held belief that virus writers are young males in their late teens to early twenties has been challenged by PaX, a member of the UK virus-writing community, who told *Vnunet.com* that, although he is indeed male, he is now in his late thirties and 'happily married with three children' (and still writing viruses). PaX was speaking out against a report produced by the analyst firm *mi2g* and named three other virus writers – including Cheng Ing-Hau, the author of CIH – who he claims '*never* spread viruses'. PaX also told *Vnunet* 'The average age of a virus programmer is 28 and none that I know of have green hair or a love of drugs and heavy metal music.' ■

The Waiting's Over

Central Command has revealed the name of its new anti-virus solution (or should that be 'has finally stocked up its shelves with one to sell'?) – V*exira* (not to be confused with those little blue pills to help with problems of a more personal nature). For a limited period (until 10 June 2002), the company is offering a free 'upgrade' to all existing *Central Command* customers who purchased anti-virus protection on or after 9 November, 2000. *Vexira Antivirus* is available now from the *Central Command* Web site http://www.centralcommand.com/ ■

Peer-to-Peer Dangers

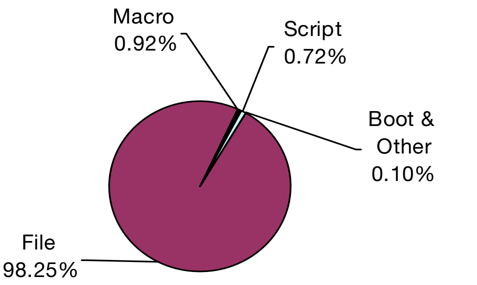
After Eric Chien warned us of malicious threats of peer-to-peer networking last month (see *VB*, January 2002, p.2), users trying to download *Napster* alternative *Audiogalaxy Satellite* from http://download.cnet.com got a little more than they had bargained for. The software's installer file was found to be infected with a W32/Nimda variant. While the infected file has now been replaced, the download page suggested that the file had been downloaded 28 million times since it was added in October 2001 ■

Prevalence Table – December 2001

Virus	Type	Incidents	Reports
Win32/BadTrans	File	13304	65.90%
Win32/Magistr	File	2297	11.38%
Win32/SirCam	File	1639	8.12%
Win32/Aliz	File	1049	5.20%
Win32/Goner	File	480	2.38%
Win32/Hybris	File	406	2.01%
Win32/Nimda	File	295	1.46%
Win32/MTX	File	95	0.47%
Win32/Klez	File	79	0.39%
Laroux	Macro	68	0.34%
Haptime	Script	47	0.23%
Kak	Script	40	0.20%
Tam	Script	40	0.20%
Win32/Ska	File	19	0.09%
VCX	Macro	17	0.08%
Win32/Kriz	File	16	0.08%
Win32/Shoho	File	16	0.08%
Win95/CIH	File	16	0.08%
Win32/Maldal	File	15	0.07%
Win32/Zoher	File	15	0.07%
Win32/Funlove	File	14	0.07%
Divi	Macro	12	0.06%
Marker	Macro	12	0.06%
LoveLetter	Script	11	0.05%
Win32/Bymer	File	11	0.05%
Win32/Gokar	File	11	0.05%
Others [1]		165	0.82%
Total		**20189**	**100%**

[1] The Prevalence Table includes a total of 165 reports across 50 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 …

Desktop Anti-Virus Testing Needs

In recent issues of *Virus Bulletin* I have seen a lot of discussion about anti-virus testing procedures. I am pleased to read that all sides in the discussion are aiming towards an improvement over current testing methods. Many of the suggestions made have been more complex in nature in terms of implementation, for example testing of anti-virus products over new virus collections and checking their heuristic capabilities etc. Most of the suggestions and opinions raised have been from members of independent anti-virus testing organizations.

Being an anti-virus developer myself, I would like to put forward some suggestions of my own, which I feel are more helpful from the user's point of view – certainly I think that users would love to have access to the results of such tests. My observations are based on the real-life needs of the end user and the problems they face when tackling any kind of virus outbreak.

The following are a few tests which I believe any testing organization could accommodate easily in their testing procedure and which are very important from the user's point of view.

1. On-access Scanning Test

- Try to access infected files on a remote machine from the test machine by browsing through Network Neighborhood.
- Try to access infected files on a remote machine from the test machine by mapping the shared folders to a local drive letter.
- Try to copy infected files on the test machine from a remote machine.
- Receive infected mails (this should include In the Wild worms that arrive by email) on the test machine.

2. Installation Test

Install the anti-virus product on a PC that has already been infected. The anti-virus product should at least detect whether there is any malicious code present on the system at installation and request that the command line scanner is run to remove the virus before going ahead with the installation.

3. Upgrade Test

Keep a PC protected with the anti-virus software, then infect the PC with a virus that is new to the anti-virus installed (and which is In the Wild or reported from many parts of the world). Check the AV vendor's Web site for the new virus update and download and update the software to detect and remove the virus. (Most of the AV vendors respond immediately to such virus outbreaks and update their Web sites with the new update/definition file for their customers.) Check how effectively the software handles such virus infection across the upgrade process.

4. Additional Components Testing

If the software being tested provides any additional features such as integrated protection for *MS Office* users or a plug-in for *Outlook* mail client etc., these features also need to be tested. Since, nowadays, most of the anti-virus products on the market provide this type of add-on or plug-in for most common email clients and *Office* suites, this is part of daily life for the common end user.

All of the scenarios described above represent real-life situations that a common end user will face daily and in all of the tests, the test set should comprise In the Wild viruses.

All of the testing procedures I have described should be carried out in addition to current test methods (which I have not repeated here), such as testing the scanner detection rates for viruses In the Wild and so on. In addition, anti-virus products should be tested for their heuristic scanning capabilities and other aspects as mentioned by Joe Wells in his article 'Pragmatic Anti-Virus Testing' (see *VB*, September 2001, p.12).

Sanjay Katkar
Cat Computer Services (P) Ltd.
India

Trouble Makers Applauded

I applaud Andreas Marx for his 'Trouble Makers' opinion piece (see *VB*, January 2002, p.14). My 'EIS exploits' project studied this threat from 1997 to 1999, and a major AV vendor tried to assassinate my character after I turned down a flagrant bribe to contain my research. That Marx could publish a paper on the topic – in *Virus Bulletin*, no less – shows we've come a long way in the last two and a half years.

Rob Rosenberger
Vmyths.com
USA

Good Statements, Wrong Conclusions

I strongly disagree with Peter Morley's opinion to remove the detection of (a few) old DOS viruses (see *VB*, December 2001, p.11).

First, I think it's a good idea to make detection of viruses more generic, saving space in signature files. But it is *not* a speed problem – most DOS viruses infect only DOS files. Therefore, only DOS files need to be scanned for these types of virus. Today, this is nearly no overhead, because most current files are *Win32* or OLE files.

The main problem *is* that today's updates always include all previous virus definitions – some signatures are more than ten years old yet will still be included in every daily or weekly update.

Therefore all AV companies should concentrate on writing scanners that use small, incremental updates. This feature should be available not only for automatic updates, but for manual updating as well. A few programs already support incremental updates, but most of them are only a little (maybe one third to two thirds) smaller than the normal updates – and therefore I wouldn't class them as truly 'incremental updates'.

One idea would be to create a separate signature file including all of today's virus signatures and CRC sums, like all the old DOS stuff. After that, a new definition file can be started, which should be able not only to include new definitions, but also to remove, change or extend definitions in the old file. The 'large' file can be updated, maybe once a year, to reduce the size of the second definition file. Or, it could be renewed as part of regular (often quarterly) engine updates.

This would save terabytes of network traffic every month. For today's outbreaks we usually have to download large signature files, consuming immense traffic and this makes a few security Web sites unreachable. With such incremental updates, we would not have the problem of more and more download servers needed every month and this would save a lot of money (for both AV companies and their customers). Also, we could release updates more often, if necessary, without thinking about the additional costs caused by too much download traffic.

This was not the only problem I saw with Peter Morley's proposals, there is another: no AV company will decrease the number of viruses they claim to be able to detect. I can remember hearing about a collection of 14,000 new viruses generated by a virus construction kit Peter Morley obtained for inspection (see *VB*, November 1998, p.10 for details). Only a few hours of analysis and tests were needed to add detection of all of them.

However, nearly all AV companies have increased the number of known viruses by about 14,000 and not only by the 10 or 15 which would represent the number of really new virus signatures needed to detect all viruses. One company increased the number first and all others have followed suit very quickly …

Andreas Marx,
AV-Test.org, University of Magdeburg
Germany

Collection Support Calls

Since my article 'Cutting Off the Tail – Revisited' (see *VB*, December 2001, p.11) was written (October 2001), in which I advocated reduced detection of legacy DOS viruses, we have received a small number of support calls (six), from customers (mainly large ones), pointing out that we no longer detect viruses we used to detect. Two points emerge.

1. All of the calls were about 'old droppers' or viruses I had regarded as 'old rubbish'.
2. All of the calls were what I refer to as 'Collection Support Calls'. None of them was about an infection which had to be handled and cleaned up. All of them had been prompted by tests the customers had done on internal collections.

I believe that our customers make these calls because they want to help us correct an error, and because they they think the situation is that we never want to miss detection of anything we used to detect. And, since that *was* the situation three years ago, who can blame them?

However, this is no longer true for legacy DOS viruses, and since the concept of a 'Collection Support Call' is relatively new, I should like to expand on how I believe they should be handled.

1. Confirm that the call *is* a collection support call. If it is not, then any virus involved should be detected for at least a further three years, and we will make immediate arrangements to do so.
2. Check whether the customer really wants detection to be put back. The virus may well have caused chaos back in 1995, and have been a historical company crisis. If so, it is perfectly reasonable to request we do it. If he says 'Yes', we should do it, and no messing!
3. If the customer says 'No. It doesn't matter now', suggest that he considers separating out the legacy DOS viruses from his collection, and not using them for future detection testing.

Peter Morley
NAI Avert
UK

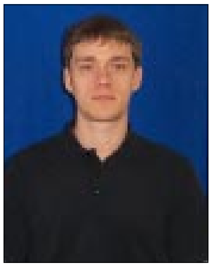
VB 2002 Call for Papers

Virus Bulletin is seeking submissions from those wishing to present papers at VB 2002 in New Orleans, USA, on 26 and 27 September 2002. The conference will host two concurrent streams of sessions, corporate and technical. Abstracts of approximately 200 words must reach the Editor of *Virus Bulletin* by Friday 22 February 2002. Submissions received after this date will not be considered. Please send abstracts (in ASCII or RTF format only) to editorial@virusbtn.com. Authors are advised in advance that the submission date for completed papers selected for the conference programme will be Friday 28 June 2002.

VIRUS ANALYSIS 1

SWF/LFM-926 – Flash in the Pan?

Gábor Kiss
Sophos Anti-Virus, UK



Another year, another proof-of-concept virus. It seems that whenever a development tool allows direct command shell access or file system function calls, sooner or later a bored virus writer will take the opportunity to create yet another proof-of-concept virus and take his/her well-deserved place in the league of time wasters.

SWF/LFM-926

The case of SWF/LFM-926 is no exception. It is a simple proof-of-concept virus that does nothing but replicate. In the following we will briefly discuss how this virus works.

Since this virus is a *Flash* file (SWF) infector, we start our discussion with a brief overview of the general structure of SWF files. SWF files are composed of a file header of variable length and one or more data blocks of variable length. Each such block has a block header that describes, among other things, the type and the size of the given block.

The blocks are processed by the *Flash* player in a linear fashion, starting at the beginning of the file. There are two main types of block: definition blocks, which define the basic graphical and sound components, and control blocks, which define how to handle these components during the playback. The virus occupies the first block in the file. As a result, it will be processed before the rest of the file is processed.

The Virus

The virus is in an action block (a control block), which is associated with the ActionGetURL action. *Flash* has its own script language called ActionScript. Although the virus uses an ActionScript command (FSCommand exec) to get command shell access, most of the virus is in shell (batch) script format in the SWF file and in this form it occupies 3237 bytes.

When the infected SWF file is played the script will execute cmd.exe with the appropriate command tail. It will display the fake message ‘Loading.Flash.Movie...’ then it pipes the virus body in debug script format to debug.exe in order to compile it into a com program. This conversion technique is not new. It can be found in certain other script viruses (mainly in batch scripts). And finally the virus runs the com file (v.com) created by debug.exe. The actual infection is carried out by this dropped com form of the virus. V.com contains the whole virus and is 926 bytes in length. It searches for uninfected SWF files in the current directory.

The virus uses the *Flash* version ID byte (the 4th byte) of the SWF file header as an infection marker. If an uninfected file is found, it checks whether the first block in the file is a SetBackgroundColour control block. If it is not, then the virus does not infect the file.

If the first block is found to be a SetBackgroundColour control block, the virus regenerates its script form by dynamically converting itself, then inserts a new block containing this script form between the SWF file header and the first block of the original file. This insertion works correctly only in the case of hosts smaller than 64 KB. If the host is larger than this limit, the virus corrupts the block structure of the host SWF file. The virus infects all uninfected SWF files in the current directory.

As can be seen, this virus is *Windows* version-specific. Because of the embedded cmd.exe reference (and other version-specific characteristics), the virus will not work on *Windows 9x*. Since the infection requires the execution of an *Intel* com program, non-*Intel* platforms (for example certain Web servers) are also immune to this virus. According to *Macromedia*, only the stand-alone *Flash* player is affected by SWF/LFM-926 and the virus will not be activated when the SWF file is played from within a Web browser.

Conclusion

Although this virus has no payload, the technique it demonstrates seems to be extendable to implement general Trojan or virus dropper SWF files. Hopefully this virus will prove to be just a ‘flash in the pan’ and the technique it demonstrates won’t be used by other viruses.

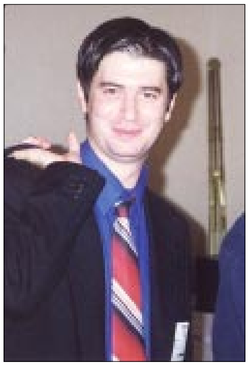
SWF/LFM-926

Aliases:	SWScript.LFM, LFM.926, ACTS.LFM.926.
Type:	*Macromedia Flash* file (SWF) infector.
Self-recognition:	Infection marker in the 4th byte of the Flash file header.
Payload:	None.
Removal:	Cut the first block out of the SWF file and fix the file header.

VIRUS ANALYSIS 2

ReST in Pieces

Costin Raiu
Kaspersky Lab, Romania



It may be the case that today the most common forms of *Linux* malware are worms, pieces of code that automatically relocate from one system to another, and rootkits, which hackers use in order to secure their access to a compromised system. Interestingly, however, some new forms of malware have appeared recently – *Linux* viruses that infect local files also embedded with backdoor capabilities. For instance, the so-called 'Remote Shell Trojan' (RST) which was mentioned initially on 5 September 2001 by the security team at *Qualys*.

Unfortunately, the people at *Qualys* did not provide the AV industry with a sample of this Trojan/virus, so an independent analysis was not possible until recently, when a newer version of RST was reported to *SecurityFocus*, who distributed it to us.

The Viral Part

The viral part of RST infects ELF executables by looking for the first PT_LOAD segment, increasing its size by 4096 bytes and inserting itself at the end of the code segment while relocating the rest of the data after the viral code. To achieve this, the virus patches the Segments and Section Header Tables, while making sure the segment with the virus code has the Read, Write and eXecute flags.

When an infected file is executed, it forks a copy of itself which will continue with the infection and backdoor processes. The main instance continues by passing the control to the original host. Next, the newly-spawned child process performs an anti-debugging check, to determine whether the current process is 'ptrace'-ed, (usually this means a debugger is being run on it). If this is the case, the child process will terminate execution immediately.

If the current process is not 'ptrace'-ed, the virus searches for all the files in the current directory, and attempts to infect them. After that, it will attempt to infect all the files in the '/bin' directory, which under normal conditions will work only if the infected program was run under an account with higher privileges.

The viral code does not show any attempt to exploit any *Linux* vulnerabilities in order to obtain higher access privileges in cases where the virus is run on a normal user account.

The Backdoor

Next, the virus attempts to run its backdoor part. To do this, it must create two new devices named '/dev/hdx2' and '/dev/hdx1' – of course, in order for this to work, the code requires 'root' access. If root access is not available and the creation fails, the child process terminates execution.

However, if the necessary access is available and the creation of the two new devices succeeds, the virus checks for the existence of a network interface named 'eth0', and attempts to set this into 'promiscuous' mode.

EGP

Also the virus attempts to create an 'Exterior Gateway Protocols' (EGP) raw socket, and puts it into listening mode over eth0. The same thing is performed for the usual PPP interface, 'ppp0'.

It's interesting to note that, unlike most other backdoors, which use either UDP or TCP to communicate with their clients, this one uses EGP, the foundation of almost all the current traffic in the Internet. (EGP itself, RFC 1265, is a rather old protocol, but EGP today includes the newer 'BGP', the 'Border Gateway Protocol' which is practically used all around the Net to route packets, make sure they arrive without errors, etc.)

The drawback of using this method of communication is, of course, that in order to sniff the EGP traffic, the backdoor requires root privileges, which are not always available.

When a special EGP IP packet arrives, the backdoor part of the virus will check whether the 23rd byte in the packet data is 0x11, then it will check for the presence of a specific password, 'DOM', as a three-byte string at offset 0x2a in the buffer.

If these two conditions are met, the backdoor will check for a 'command' byte, which is either 1 or 2. '1' spawns a standard '/bin/sh' shell which the attacker can control on the remote system.

Command byte '2' attempts to perform something of a strange action, which is to send the 'DOM' password back to the attacker through a UDP connection on port 4369 (0x1111). I consider this action to be somewhat strange, because I see no point in sending back the password – in order to control the backdoor, the attacker has to know the password in the first place. Maybe the purpose of this action is to check whether the remote system can initiate a UDP connection to the attacker (if it is not behind a firewall).

Points of Interest

An interesting point to note is that, when attempting to set the network interfaces into promiscuous mode, the virus may try to report the infected machine (back to its author?) by connecting on HTTP to 207.66.155.21 (which points currently to ns1.xoasis.com) and requesting the '/~telcom69/gov.php' script. At the time of writing, this points to a nice notice stating that the respective account was terminated due to 'virus abuse'.

Also of note is the existence of two strings which can be seen inside the virus code, but which are not used anywhere in the code. They are 'snortdos' and 'tory'. I assume that these are either the author(s)' nicknames, or some piece of unreferenced data which was used in a code that has been removed from the virus, or which has yet to be implemented.

Conclusions

First, it's disappointing to realize that not everyone seems to understand the importance of cooperation between companies working in the security field. Despite the fact that many important steps have been taken in this direction over the last year, there is still a long way to go.

Secondly, even if the RST virus lacks the rapid propagation techniques of a networking worm, the ideas it demonstrates (such as using more complex and less common protocols for its backdoor parts, or the simultaneous combination of viral and backdoor capabilities) can only suggest that malware is becoming more and more complex, and that virus writers are making major efforts to create forms of malware that are harder to notice or which reach higher spread factors.

Finally, since virus authors often share their knowledge or steal the tricks they see in new viruses, we can expect to see more forms of malware using these techniques in the future. I wouldn't be surprised to see even more complex backdoors on *Win32* systems, bundled with sniffers, using raw sockets to communicate on custom protocols, and maybe automatically exploiting various vulnerabilities such as the recent *Windows XP* UPnP bug to replicate.

Linux/RST.b

Type: *Linux* ELF-infecting virus which contains a backdoor.

Payload: Attempts to report the infection by accessing the script http://ns1.xoasis.com/~telcom69/gov.php. Provides an attacker with an open shell.

Removal: Use an anti-virus to identify the infected files, clean them, or restore them from backup.

VIRUS ANALYSIS 3

Bad Transfer

Peter Ferrie and Péter Ször
Symantec Security Response, USA

In December 2001 *Symantec* received its one millionth customer sample submission. We waited for the big moment, but the millionth submission arrived sooner than we expected. This is because a new worm, Win32/Badtrans.B@mm, was making its way quickly around the Internet. The worm was released in late October 2001 and during December we received 30,000 submissions of this worm alone. This is an extremely high number for a single month and at least twice as many as we have experienced before.

So what happened to Badtrans? Why did it become so widespread all of a sudden? The original variant was in the wild from April 2001 and did not attract much attention, even though it was reported to the WildList.

First of all there are a number of new features in this worm that would easily cause it to be considered a new variant. The '.B' letter was fairly arbitrary, given that there were at least three variants of the worm known at the time. Badtrans uses techniques picked up from the Nimda virus, and it is clear that these techniques contributed highly to its success.

Configuration Bits

This worm arrives as an email with one of several attachment names and a combination of two appended extensions. The attachment contains the worm code and an appended block of configuration data which controls its behaviour. Clearly the author wanted to change the behaviour of the worm without recompiling the code. Thus the configuration data at the end of worm works much like a .ini file. The worm's author probably had a patch tool to create different behaviours during testing.

The worm also has the ability to replace existing copies of itself with newer versions. Badtrans.B is written in Visual C++ and packed with UPX, which is a common runtime compressor. With such a widespread worm, we might have expected to have seen new variants by now, however the configuration data contains file offsets which change if the file is altered in any way (such as repacking) and such alterations will prevent the worm from running correctly.

The configuration data contains various things, such as the name of the registry key, the registry value and data to use, the names of the files to create (for the worm itself, the key logger, and the data files), and the texts that will cause the key logging to begin. Additionally, there are control bits that are checked by the worm code, and a unique identifying value for controlling the overall execution.

The control bits control the logging, the encryption, the directories in which files are created, and what is stolen (keystrokes and/or cached passwords). Thus many things can change based on these values.

The unique value is used as a parameter to verify the requests to run a new copy and delete the old one. When the worm is executed, first it will search for and terminate all other running copies of itself. Then it will append the unique value to the word ‘Restart_’ and run again with this parameter. If this parameter has been specified already, then the worm will run itself yet again, but with an additional parameter which is the unique value appended to the word ‘Kill_’. The purpose of the Kill command is to delete the file that was used to launch the worm initially.

Auto Launcher

The worm uses the malformed MIME exploit to execute automatically. The emails are HTML format combined with a malformed MIME header that causes *Microsoft Outlook* to execute the attachment immediately and without prompting. More on the exploit, including the necessary patches to protect the system against such an attack can be found at http://www.microsoft.com/technet/security/bulletin/MS01-020.asp.

When Badtrans.B is first executed, it copies itself to %System% or %Windows%, depending on the control bits, using the filename contained in the configuration data (currently ‘kernel32.exe’). Then it registers itself as a service process (*Windows 9x/ME* only) to hide its presence from the Task List in *Windows*. It creates the key logging files in %System%, whose names are specified in the configuration data (currently ‘kdll.dll’ and ‘cp_25389.nls’).

The log file is encrypted with a simple algorithm, whose keys come from a string in the configuration data (currently ‘uckyjw@hotmail.com’). Andreas Marx at *AV-Test.org* developed a *Windows* application to decrypt the encrypted log file. This may be of use to anybody who wants to know what has been logged and possibly distributed to the hacker.

The DLL is loaded and several functions are accessed dynamically from it. The code of the DLL is stored in the resource section of the worm’s code. That explains the choice of UPX, a packer that does not pack resource sections. There are some buggy packers that pack resources and therefore cause problems for certain applications that use *Windows* resource APIs.

%Windows% and %System% values vary from system to system. The worm locates the \Windows folder (by default this is C:\Windows or C:\Winnt) or the \System folder (by default this is C:\Windows\System or C:\Winnt\System32) and copies itself to that location.

Password Stealing

A timer is used to examine the currently open window once every second and to check for a window title that contains particular characters. Currently, these are ‘LOG’, ‘PAS’, ‘REM’, ‘CON’, ‘TER’ and ‘NET’. These texts form the start of the words LOGon, PASsword, REMote, CONnection, TERminal and NETwork, respectively. There are also Russian versions of the same words in the list.

If any of these words are found, then the key logging is enabled for 60 seconds. When the logging delay expires (currently every 30 seconds), the log file and the cached passwords are sent to one of several addresses (some of which are currently not operational, some contain obscene words and are not listed here), using one of several SMTP servers. The addresses are:

ZVDOHYIK@yahoo.com
DTCELACB@yahoo.com
WPADJQ12@yahoo.com
I1MCH2TH@yahoo.com
udtzqccc@yahoo.com
YJPFJTGZ@excite.com
JGQZCD@excite.com
OZUNYLRL@excite.com
XHZJ3@excite.com
rmxqpey@latemodels.com
muwripa@fairesuivre.com
cxkawog@krovatka.net
ssdn@myrealbox.com
bgnd2@canada.com
smr@eurosport.com
tsnlqd@excite.com
eccles@ballsy.net
fjshd@rambler.ru
S_Mentis@mail-x-change.com.

The SMTP servers are:

mx2.mail.yahoo.com
mail.ifrance.com
fs.cpio.com
mail.monkeybrains.net
usa-com.mr.outblaze.com
mta.excite.com
mail5.rambler.ru
mail.canada.com
smtp.myrealbox.com
mail.ukr.net
mail-fwd.rapidsite.net
imap.front.ru
inbound.latemodels.com.criticalpath.net
inbound.ballsy.net.criticalpath.net.

The email addresses as well as the server names, are encrypted in the worm’s code. Since there are Russian server names and Russian words in the password stealing routine it is reasonable to assume that this worm has a Russian origin.

After 20 seconds, the worm shuts down if the appropriate control bit is set.

Sending Mail

If RAS support is present on the computer, then the worm waits for an active RAS connection. When such a connection is made, with a 33 percent chance, the worm searches for email addresses in *.ht* and *.asp files in the Personal and Internet Explorer Cache folders. If it finds addresses in these files, then it sends mail to those addresses using the victim’s SMTP server. The attachment name will be one of the following:

Pics	images
README	New_Napster_Site
news_doc	HAMSTER
YOU_are_FAT!	Stuff
SETUP	Card
Me_nude	Sorry_about_yesterday
info	docs
Humor	fun

In all cases, MAPI will also be used to find mail to which the worm will reply. The subject will be ‘Re:’. In that case, the attachment name will be one of the following:

PICS	IMAGES
README	New_Napster_Site
NEWS_DOC	HAMSTER
YOU_ARE_FAT!	SEARCHURL
SETUP	CARD
ME_NUDE	Sorry_about_yesterday
S3MSONG	DOCS
HUMOR	FUN

The worm appends two extensions. The first should be one of the following: .mp3, .zip, .doc, but because of a bug, .zip is never chosen. The second extension that is appended to the file name is .pif or .scr. The resulting file name would be something like, for example, CARD.doc.pif or ME_NUDE.mp3.scr and so on.

If SMTP information can be found on the computer, then it will be used for the From: field. Otherwise, the From: field should be one of the following:

“Mary L. Adams” mary@c-com.net

“Monika Prado” monika@telia.com

“Support” support@cyberramp.net

“Admin” admin@gte.net

“Administrator” <administrator@border.net>

“JESSICA BENAVIDES” <jessica@aol.com>

“Joanna” <joanna@mail.utexas.edu>

“Mon S” <spiderroll@hotmail.com>

“Linda” <lgonzal@hotmail.com>

“Andy” <andy@hweb-media.com>

“Kelly Andersen” Gravity49@aol.com

“Tina” <tina0828@yahoo.com>

“Rita Tulliani” <powerpuff@videotron.ca>

“JUDY” <JUJUB271@AOL.COM>

“Anna” <aizzo@home.com>.

However, due to a bug, only every second name in the list can be chosen. In order to prevent multiple emails to the same person, Badtrans.B writes email addresses to the %System%\Protocol.dll file. Additionally, the underscore (_) character is prepended to the sender’s email address, which interferes with replying to infected mails to warn the sender of infection (for example, user@website.com becomes _user@website.com).

Before sending email, the worm will look in the registry for the name of a DNS server. If one cannot be found, then it will use a default server whose IP address is stored in the worm code.

The DNS server is used to verify that the domain specified in an email address is truly valid. This idea exists in Nimda, too, however in the case of Badtrans.B the result is ignored and email is sent even if the domain cannot be verified as valid.

After sending the mail, the worm adds itself to the registry key specified in the configuration block, using the specified value and data (currently, these are ‘HKLM\Microsoft\Windows\CurrentVersion\RunOnce’, ‘Kernel32’, and ‘kernel32.exe’). This causes the worm to run the next time *Windows* is started. These values can differ based on the control bits mentioned previously.

Conclusion

Evidently the use of exploits in computer viruses is becoming increasingly common. Additionally, many malicious hackers are creating mass-mailing worms by combining backdoors and other password-stealing applications to make them more successful. Just like Sircam and W32.HLLW.GOP@mm, Badtrans started up as a password-stealing Trojan and became more successful once the MIME encoding was added to the SMTP mass mailing that was already present.

Hackers are becoming increasingly interested in your personal information! We should all be sure to keep aware of recent exploits and apply all security patches to protect ourselves from a lot of trouble. Apparently the Nimda virus spread this message well enough, yet Badtrans.B became successful with a very similar strategy.

Win32/Badtrans.B mm

Alias:	I-worm.BadtransII, W32/Badtrans.B mm.
Type:	SMTP mass mailer that uses malformed MIME exploit to execute itself automatically if *Microsoft Outlook* is not patched.
Size:	29,020 bytes.
Removal:	Stop and delete worm process, fix registry values.

FEATURE

Test Lab Installations

Andreas Marx
AV-Test.org, University of Magdeburg, Germany

It's a familiar problem: the preparation of more than one computer in a test lab. PCs may have different hardware configurations, but there are several with exactly the same hardware. At *AV-Test.org*, for example, we have five identical P-III-800, five Athlon 1.3 GHz and two P-III-800 multiprocessor systems.

Of course, it should be easy to use different OS versions (like *Win 98* or *2000*) on one PC, but nobody wants to install all *Windows* versions on all computers, with long driver and software installation sessions.

PC Simulator

An interesting idea would be to use *VMware* (which can be downloaded from http://www.vmware.com/) in its most current version (3.0). *VMware* is a full PC simulator – you can run *Windows XP* from a *Linux* system, for example.

The other great advantage of *VMware* is that, with the exception of the CPU type, it simulates the hardware completely – for example the Ethernet adapter or PCI bus drivers. The current state of the simulated PC can be frozen in 'Suspend' mode at any time, during installation or normal work, and restored later with only a few seconds delay, making it faster than starting up a PC.

The simulated PC memory of this 'Suspend' mode will be written to disk and can be inspected using a file viewer. But the main advantage is that it is possible to copy the image files easily from one PC to another (with different hardware, processor and so on) and the simulated OS will not find any changes. This means that the system only has to be installed once and can be used as often as required on different hardware.

However, *VMware* is still a simulator and therefore slower than a normal PC. It is useful when answering customer support calls for inspecting problems using different customer operating systems, and it can be useful for small tests or even controlled virus replication. But for time-consuming tests, like larger virus scan sessions, *VMware* is too slow.

The Goals

Therefore, we still face the problem of different hardware configurations. The goal should be to have small installations, but which include all the necessary files of all the operating systems we need and which can be restored in less than one minute to test complex issues from a clean, consistent system. In this situation only the essential OS or *Office* components will be installed – viruses don't need a grammar or spell checker, for example.

At the same time, we don't want to boot from a floppy disk: not only is it much slower, but more important is the risk of 'forgotten' boot virus-infected disks after a test. Therefore, the boot sequence will be modified after all images are running satisfactorily.

On all of the different hardware the OS has to be installed at least once to avoid instability issues and driver problems. (For professional systems like *Win 2000*, tools exist to prevent these problems, but the instructions for their use would fill more than an entire issue of *Virus Bulletin*.)

Using a drive image software package such as *Ghost* the remaining PCs can be installed easily, changing only small parts like the computer name or the IP address (at *AV-Test.org* we do not use DHCP).

In our case, we need to install all operating systems on three PCs, because we have three different hardware configurations. However, we used all of the PCs for the installation session – three to install the German version, three for the English versions, two for documentation and the rest to install other language versions. It's quite easy to install these different language versions, since the screen messages mean exactly the same in the different languages.

Preparation

First, we ensure that all PCs with the same hardware have the same BIOS version with identical options. For example, we always disable APCI and other power save modes. We note all the changes we have made in the documentation so that the settings can be restored easily, if needed.

The next step is a run of fdisk of *Win 98* in the correct language version. All of our PCs use FAT16-only drives, because *Win NT* does not support FAT32 and *Win 98* cannot read NTFS partitions without additional tools and so on. However, all operating systems can use FAT16 with a limitation of 2 GB per partition. This does not present a problem, because we can add more of them – our system uses three partitions: one for the main OS installation, one for a few drivers we don't want to install from disks, plus tools for partition imaging as well as the swap file, and the last partition is reserved for image files of all operating systems.

Next, we format drive C: using a *Win 98* boot disk and the '/S' switch to ensure it's bootable. *Win 9x*-based systems will overwrite the boot loader using their own system, but *NT*-based operating systems will add a simple boot manager menu – we use this to be able to restore images easily without the need for floppy boot disks. We can start a DOS session and run *Ghost* after that.

If drives D: and E: are formatted, we continue to copy all necessary drivers to drive D: (as for the network adapter), as well as the drive image software, in this case *Ghost*. We also copy a file manager such as *Norton Commander for DOS* or the *Volkov Commander* onto the drive, so we cannot get lost in the command line completely.

As the first step, we will make an image file of the nearly empty, but bootable C: partition – we will use this for all later OS installations. After that, we can start to install the preferred OS (usually from a bootable installation CD).

Once the installation is completed and *Windows* shows a 'Welcome' screen, we start DOS (using the boot menu or a bootable floppy) and save this first installation to an image file. This is useful if we need to test something with a very clean installation.

Installation of Drivers and Programs

Next, we install all drivers and programs – this includes Service Packs for *NT*-based systems. However, for *Win 9x*-based systems we don't install much, because these systems tend to exist fully unpatched in the real world.

We do not install a newer version of *IE*, nor do we install new versions of *DirectX*, the *Media Player* and so on – mainly to save space. The only exception is *NT*, where we install *IE 4.01 SP1* which comes with the Option Pack and several other programs require it.

In addition, we install a few applications that we use very frequently, such as a small screenshot program, a file manager (I find I cannot live without *Windows Commander*, see http://www.wincmd.com/), *Acrobat Reader* for the inspection of documentation and *Winamp*.

Also we install a generic text-only printer (included in all *Windows* versions) so that we can save log files of scanners, if they do not provide a 'Save as' function. All applications can be accessed either by an icon on the desktop or using a hot key.

We make a note of all the changes and installations we have made as we go along, and at regular intervals we create new image files. We can use these image files if we find that a new driver causes problems, if we have changed the wrong registry keys or deleted too many files in the following steps. Alternatively, the image files could be used simply to test programs that do not run under our special test installations, although we have never encountered this situation.

Reducing images

Of course, we have at least 2 GB we can use on drive C:, but the smaller the installation, the smaller the image file and it will also take less time to be created or restored. Now it's time to reduce the size of the image file.

First, we can use the maximum compression for the image tool, for *Ghost* this is 'ghost -z9'. However, the size of a standard image file is still 110 MB for *Win 98*, up to 472 MB for *XP Pro* and most people stop here.

The first step should be to disable the Hilbernate mode: if enabled, a very large, hidden system file called hilberfil.sys can be found in the root of the C:\ directory. The system will save the current content of the memory to this file and all image programs will save the file, which is very time-consuming.

The next step is to change the swap file location – good image software will not store the swap file, but it's easier if we don't need to worry about the swap file.

As a third step, we investigate the files that, hopefully, we will not need any more on the systems, such as txt and readme files, bitmaps, wav, pnf, temp and log files, as well as all of *Windows* help files. If needed, we can still copy them in the correct folder from a network folder.

Windows 98

For *Windows 98* we install a boot menu by changing the msdos.sys file and placing the 'BootMenu=1' under the '[Options]' section so we can start the command line if needed to save or restore the image file. For this, a 'path' variable should be set in the autoexec.bat file to access the utilities directly.

Also, we delete all program starts in the 'Run' registry key under 'HKLM\Software\Microsoft\CurrentVersion\Run' except Systray.Exe – this means that the system starts much faster and no useless tasks will be started on every boot. We do the same for the 'RunOnce' key.

Windows ME

Under *Windows ME* the process is almost identical, but the 'PCHealth' task should not be deleted in the 'Run' key. Also it is not possible to start DOS directly, but we can use a simple trick to prevent the need for a boot disk. The program wininit.exe will be started at every boot if a file wininit.ini exists in the *Windows* installation directory.

If we want to *Ghost* a PC, we simply have to copy the image program to %windir%\wininit.exe (overwriting the existing one) and we have to create the ini file. After that, we can reboot the machine and *Ghost* will start automatically. We have created a simple batch file for this, and to reboot we use the command 'RunDll32.exe Shell32.dll,SHExitWindowsEx 0x2'.

Windows ME also creates a folder called '_Restore' and saves a lot of data here. In most cases, we do not need the data stored in this folder if we create a clean image. We can only delete the data under plain DOS, but it will save a lot of space. *Windows* will recreate the folder during the next boot-up and PC-Health can still be used. %windir%\Options\*.* can be deleted, too.

Windows NT

Windows NT (both *Server* and *Workstation*) is much smaller than *98* or *ME*. However, there is still some room for improvement. For example, we found all Help files twice in our installations – under 'system32' as well as in the correct 'Help' folder.

After the reinstallation of *SP 6a* and the *Security Fix Rollup Package* (*SRP*), we delete all folders like '$NTUninstall$' (a backup of all modified data by Service Packs or Hotfixes will be saved here, but since we do not wish to uninstall them, we can delete it). It is also useful to clear the event log, not to save space, but to give us a better overview of what has happened if we start tests.

Windows 2000

Under *Windows 2000* the size of the image can be reduced significantly. First, all drivers will be stored in the '%windir%\Drive Cache\i386' folder in a large cab file called driver.cab as well as one or two smaller cab files installed by a Service Pack. We should not simply delete these, but it is a good idea to store them on a (read-only) network drive.

All test systems can share these files – we only have to change a registry key which can be found by searching for 'DriverCachePath'. After a reboot, we can delete the 56 MB files without any negative effects. Everything still works as it should, and *Windows* does not prompt for an installation CD if we want to install additional hardware.

In the next step, we should look at a folder called '%windir%\ServicePackFiles\i386', which contains copies of all the Service Pack files that have been installed. After copying all files to a network drive, we should change the Registry key for 'ServicePackSourcePath', reboot, and delete about 160 MB.

Windows 2000 has a protection against the replacement of important system files. For this, a copy of the protected files will be stored at '%windir%\system32\dllcache'. We can delete the content of this folder too. If a program tries to replace a system file *Windows* will prompt for an installation CD and will not simply restore the original. I think it is much more useful to be alerted in this way if something goes wrong with a program. As with *NT*, it is useful to clear the event log to give us a clear overview of what is going on.

Windows XP

For *Windows XP* (both *Home* and *Pro*) the steps are nearly identical to those for *Windows 2000*. However, no Service Pack exists for *XP* at the moment and in our test installation only a few cat files can be found in the Dllcache, because *Windows* only uses this feature if the installation partition is larger than about 3 GB. This is also a good time to activate *XP* – it will remain activated if the image is installed on the other test computers.

After we have installed all *Windows* versions, we can install all the *Office* versions we need to test programs or replicate viruses. However, we only use *Windows 98* to install *Office*, because it's both small enough and the fastest system to start.

Once we have finished all installations, we run a disk-fragmentation utility, if one is available. This does not save any space, but it does make the test systems more unified.

We can transfer all images to the other PCs, which have only an empty, but prepared HDD at the moment ('fdisk' has been run etc.). To do this, we copy an image of *Windows 98* (for all different configurations) to a bootable CD and restore the correct one. After this, we can start *Windows 98*, copy the rest of the images from a network drive to the local disk and *Ghost* them one after another.

To finalize all images, we change the label of the C: partition, the name of the computer and its IP address. But we also clean the 'last used documents' cache, copy our standard configuration file for WinCmd to the disk and create an image.

After all the steps, we have a *Win 98* image of 77 MB size and a Ghost time of less than 30 seconds. *Win ME* is 102 MB, *Win 2000 Pro* is about 141 MB and takes only one minute restore time.

We don't use the GUI version of *Ghost*, but we have written a simple program that displays all the available image files and which starts *Ghost* using the command line switch '-clone,mode=pload,src=file.gho:1,dst=1:1 -rb -sure', where 'file.gho' is the image file that will be written to the first partition of the first hard drive. The '-sure' switch prevents an additional 'are you sure?' question and '-rb' will reboot the computer after the image has been restored successfully.

Testing on Other Platforms

Of course, these are only our test systems for simple *Windows*-based product tests. For Server tests, we have additional hard disks where only one system is installed using NTFS partitions. We can exchange the hard disks easily, because they are all located in a mobile hard disk rack.

The same applies to other platforms like *Netware*, *Linux* or *FreeBSD*. We always try to reduce the size first and at least two FAT partitions can be found on the disks: one to be able to boot DOS without the need of an extra disk and one to store the *Ghost* images.

We needed about two complete weeks and three people to set up our lab with all *Windows*-based test environments, including the different *Office* installations. However, now this has been done, we find that it is a very efficient way to work, especially since it saves a lot of waiting time before a test can be started.

OPINION

Quo Vadis Anti-Virus?

Rainer Fahs
EICAR, Belgium

Every now and then reports in computer or security magazines predict the end of the virus problem. Most of these claim that AV technology has a grip on the problem now. Usually such reports are supported with some statistics about the last virus epidemic outbreak, which resulted only in some damage, while the AV industry was quick enough to update their data files in order to stop the infector spreading further. Even the widespread outbreaks of CodeRed and SirCam – which received a lot of press attention worldwide, but rather little excitement among professional AV researchers – did not change this attitude.

Does this mean we can sit back? Can we rely on technology that concentrates more on post-infection action than on prevention? Is it really enough to find and identify malicious code on one's computer, assess the damage, repair it and go back to business?

Is virus-writing 'cool', and letting viruses loose just a nuisance that society ought to tolerate? If not, what mechanisms does society have to prevent this? Are technical mechanisms sufficient or do we need laws – or a combination of both? And if we need laws, how do we bring them into effect on the Internet?

The Problem

The real problem is a multi-faceted one, which needs to be tackled with a more holistic approach, encompassing:

- Technology and technique
- Ethics, morals, laws, and education
- Awareness, warning, and recovery
- Organization and trust.

In light of the new mass-mailing viruses, it has become quite obvious that keeping your AV product up to date is no longer sufficient protection. My computer was infected by Badtrans.B shortly *before* I was advised to update my AV product with the relevant new signature files.

A fact that does need careful consideration is that the time lapse between the first appearance of a new piece of malware and the availability of an effective defence mechanism (new signature files) is too long. Not only that, but there are few co-ordinated efforts available to warn users effectively prior to infection.

The users have started to react – networks connecting users and experts (not necessarily from AV vendors) are growing rapidly around the world. The Anti-Virus Information Exchange Network (AVIEN), a large user community, was founded a little over a year ago in the USA, and the EICAR WG II in Germany has established a method of quick information exchange.

Is this an uprising of the users largely ignored by the marketing-driven AV vendors during the recent past? Many vendors are still selling products that fight the problem at the last bastion of a large battlefield – the end-users' PCs – rather than at multiple layers within networks, not to mention Internet solutions. Where are the new technological solutions? Where is the incentive amongst the gurus of the AV scene? Are they just too busy with updating their signature files or engines, leaving no time for real research?

Where are the academic researchers at the universities around the world? On *VForum*, 24 December 2001, Prof Klaus Brunnstein of the University of Hamburg, wrote: '… I believe that basic research is heavily needed to analyse those paths which enterprises are less likely to follow as they may not promise Return-on-Investment (honi soit qui mal y pense :-)'. I guess that we have indubitable agreement on that statement. However, where are the initiatives from the universities?

Progress

In the areas of ethics, morals, laws, and education there has been some noticeable progress in the recent past. The Convention on Cybercrime signed by 26 Member States of the Council of Europe and four other non-member countries (Canada, Japan, South Africa and the USA) on 23 November, 2001 in Budapest has been implemented and will now be incorporated into national laws.

The Convention will give us equivalent laws throughout the European Union and defines the formerly controversial subjects of discussion such as: Illegal Access, Illegal Interception, Data Interference, System Interference, Misuse of Devices, Forgery and Fraud.

Certainly this is laudable progress in one area. However there are still issues remaining and the EICAR Task Force on the Cyber Crime Convention will continue to work on those. Currently EICAR is involved in two separate areas. One is the Belgian government's Early Warning and Information System (EWIS), and the second is the European Commission's Information Technology Systems Research Program.

Task Force

A special Task Force was set up to investigate EICAR's potential future involvement in similar programmes. The first challenge was to determine whether we would be able to submit a formal proposal in order to win an EC project under the Information Society Technologies (IST) Programme.

At a meeting of EICAR members on 29 December 2001 in Munich, all options for such a proposal were discussed in great detail. Unfortunately, the meeting concluded that it is impossible either for EICAR to build the necessary legal entity or to allocate the experts required to undergo such commitment. The board members of EICAR decided not to submit a formal proposal. However, the decision was made to actively support and contribute to the parallel activity of EWIS within the EC.

In order to be able to identify issues and initiate and co-ordinate actions and initiatives, a framework was drafted which we have called: ‘Cyber Defence Alliance’ (CDA). This framework identifies some ambitious objectives such as:

- Global co-operation with anti-virus and security organizations
- Support of the EC Convention on Cyber Crime
- Support of the EC RTD Information Security Technology (IST) Programme
- Warning/verification/reporting
- Standardized and automated reporting
- Central database of malicious code
- Unified naming convention
- Criteria/requirements for AV expertise
- Certification and licensing
- Education and awareness
- Support of research which stipulates enhancements in defence mechanism (EICAR AVEP/Survey).

In support of global co-operation, EICAR and AVAR (the Association of Anti-Virus Asia Researchers) have agreed to work closely together and, in December 2001, the Chairmen of the two organizations signed a Statement of Mutual Recognition and Cooperation (see *VB*, January 2002, p.8).

The members of EICAR noted the initiative of Frank Felzmann and Guenther Musstopf for a unified naming convention (see *VB*, January 2002, p.12) and we encourage all serious AV researchers to support that initiative. However, success for such an ambitious objective is only possible if legacy issues and mental boundaries can be overcome. We have to realize that the problem is neither one within national boundaries nor one that can be solved within artificial marketing boundaries.

EICAR will continue to establish co-operations and to encourage activities in support of the Cyber Defence Alliance. We know that there are some hard nuts to be cracked, but with the support of all the fine experts around the world, we are confident that we will be able to make progress.

COMPARATIVE REVIEW

It’s ME Again!

Matt Ham

The last *Windows ME* comparative review – in the February 2001 issue of *Virus Bulletin* – saw VB 100% awards earned by six products from a field of 17. One year on, 18 products are in the line-up. The newcomers to the roster are *Command AntiVirus*, *F-Secure Anti-Virus* and *Trend PC-cillin*, while a product from *Network Associates* is the most noteworthy of the absentees from this test. (On this occasion *NAI* failed to supply a product by the cut-off date for submissions, though we are assured that *NAI*’s offerings will continue to grace the pages of future comparative reviews.)

Windows ME is the closest any *Virus Bulletin* product review comes to the home-user market, and certainly a couple of the products submitted for testing were more consumer- than business-oriented. Possibly more surprising is the number of products which were submitted not just for the *ME* review but which have been seen in an identical form on other *Windows* platforms – ‘*Product-Scan for Windows 95/98/ME/NT/2K/XP*’ is hardly a name that trips off the tongue, but it is not too dissimilar to some of the actual product titles.

In terms of new features, both *Kaspersky Anti-Virus* and *GeCAD*’s *RAV* presented new GUIs in these tests, with the *Kaspersky* product having a new scanning engine in addition. As for the problems that beset products in the previous *ME* review, these were twofold – problems with boot-sector detection and instability due to the production of massive log files in memory.

Test Procedures

Since there are still some questions as to exactly what earns a product a VB 100% award, there follows a short explanation. This is mostly unchanged from a year ago, with some slight modifications and clarifications.

In order to achieve a VB 100% award a product must detect in its default settings all viruses on the top half of the WildList of the month prior to its testing. ‘Default settings’ refers to such selectable items as sensitivity of detection, scanned extensions and the use of heuristics. Settings not related to detection may be changed to facilitate the production of realistic results. Full detection must be demonstrated in both on-access and on-demand scanning.

For on-demand testing, results are as first choice taken by parsing of log files, with the setting of ‘report only’ selected. Network and CD scanning has been noted to introduce sporadic errors into the test results and thus this is done on a copy of the test sets on a local hard drive.

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
Aladdin eSafe Desktop	0	100.00%	2	99.83%	99.84%	34	99.13%	71	92.50%	31	98.73%
Alwil AVAST32	0	100.00%	2	99.87%	99.88%	21	99.53%	52	95.59%	49	97.46%
CA InoculateIT	0	100.00%	7	98.21%	98.30%	0	100.00%	1	99.94%	8	99.31%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	12	99.35%	0	100.00%
Command AntiVirus	0	100.00%	2	99.40%	99.43%	0	100.00%	43	97.50%	2	99.95%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.94%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	8	99.81%
F-Secure Anti-Virus	0	100.00%	4	99.71%	99.72%	0	100.00%	43	97.50%	23	99.66%
GDATA AntiVirusKit	0	100.00%	4	99.79%	99.80%	19	99.60%	43	97.50%	1	99.98%
GeCAD RAV	0	100.00%	1	99.91%	99.92%	0	100.00%	52	97.56%	23	99.15%
Grisoft AVG	0	100.00%	1	99.96%	99.96%	26	99.40%	181	87.85%	56	98.00%
Kaspersky Lab KAV	0	100.00%	4	99.79%	99.80%	19	99.60%	0	100.00%	1	99.98%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	29	98.65%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	13	99.66%	60	95.48%	13	99.50%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.81%
Trend PC-cillin	0	100.00%	0	100.00%	100.00%	1	99.99%	234	93.86%	7	99.83%
VirusBuster VirusBuster	0	100.00%	0	100.00%	100.00%	0	100.00%	69	93.33%	8	99.81%

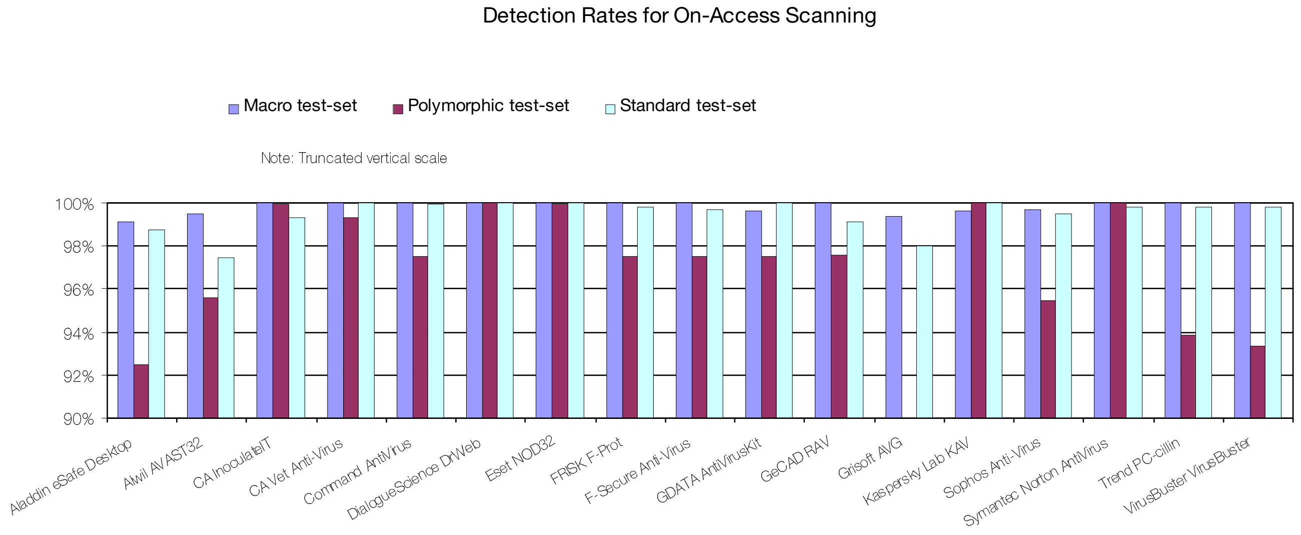
However, it has been the case in many products of late that log files are either useless for *VB* results or that the taking of log files causes the scanner to crash after a certain size is reached. In such cases the preferred method is to run a scan selecting delete as the option, followed by another choosing quarantine and another scan to check that no further files are being detected as viral. Those files remaining are regarded as misses.

For on-access testing, a selection of tools is used which seek recursively through the test sets, opening each file in turn. Scanners are set to block access on opening an infected file and a tool generates a log of those files opened.

For products which scan on file write rather than open a different method is used. Under operating systems where such a function is available natively, the test set is copied using a command which allows the blocking of individual copy operations. In this test the XCOPY command was used for this purpose. *Aladdin eSafe Desktop* uses a slightly different method of decision-making to determine when a file should be scanned. In order to simulate this activity a custom tool is used for this product.

Some products show unrepeatable misses during on-access testing which are attributable to the massive flow of infected files through the scanning engine. For these products on-access testing may be performed with deletion in the same manner as described for the on-demand tests.

For false positive detection the scanners are required to produce no false positives on the OLE and clean test sets. Many products declare files to be ‘suspicious’. This is not considered to be a false positive but is registered in the table of results by the numbers enclosed in square brackets. A test of archive scanning speed is performed for purposes of testing the rate of scanning only. In this test most products scanned inside archives by default. Those products where it was necessary to activate archive scanning manually were tested with archive scanning off for the non-archived test sets. These products were *Sophos Anti-Virus*, *Alwil AVAST32*, *GeCAD RAV* and *F-Prot Antivirus*.



The Test Sets

The test sets were aligned to the December WildList, giving a sizeable gap between this update and the August WildList used in the November 2001 test of *Windows NT* products. The nature of these changes is a good reflection of the way the virus threat is progressing at the moment.

Changes to the set consisted of 39 leaving and 31 entering the list, but the proportions of virus type showed a greater difference. Leaving the test set were 28 macro viruses with a scattering of script, boot, DOS and 32-bit *Windows* viruses making up the remaining 11. Of those viruses entering the list for the first time the figures were almost reversed, with four new macro viruses and the catch-all group of ‘the rest’ accounting for 27 new samples. Of these, 21 were 32-bit *Windows* files which certainly seem to be the current fad among virus writers.

Looking at these figures, what might be expected as a result? With the majority of the new 32-bit viruses being more exactly defined as worms, and of these mostly non-polymorphic, it might be anticipated that these samples would be easy to detect. On the other hand, the new extensions used by some of these samples are a guaranteed way of inspiring otherwise competent scanners to miss detection.

Aladdin eSafe Desktop 3.0.33

ItW Overall	99.84%	Macro	99.16%
ItW Overall (o/a)	99.84%	Standard	98.81%
ItW File	99.83%	Polymorphic	92.47%

The behaviour of *eSafe Desktop* is, at first glance, an example of the problems associated with the new extensions used by ItW viruses. The offending samples here were W32/Nimda.A with .ASP and .HTM extensions. However, since all files in the test set were scanned on demand, this does not seem to be an extension-related pair of missed samples. Other than this, detection on demand was good, with polymorphic detection being an area in which *eSafe Desktop* is still showing improvement in detection rates.

The boot-sector testing was perfect both on demand and on access, which brings us back to the problems with boot-sector detection in the last *ME* test. On that occasion several products had major problems with this area of detection – on this occasion, all products were able to detect all samples both on access and on demand. Not only this, but the user-friendliness has also improved in many cases, resulting in a great sigh of thanks from the reviewer.

On-access tests of *eSafe Desktop* showed almost identical results to the on-demand tests and thus speed tests are the next area of interest. Here there was one oddity, standing out in the scanning of the clean test set, which showed a propensity to slow down and appeared to halt on several of the files in the test set. At first it was assumed to have crashed, but eventually the test concluded. Unfortunately this test produced three false positives – the same number as produced a year ago. The remaining speed tests showed a similarity with past results: the scanning of executable files is not very fast, but OLE files are performed at the faster edge of average for this test.

Alwil AVAST32 3.0.419.0

ItW Overall	99.86%	Macro	99.55%
ItW Overall (o/a)	99.88%	Standard	98.46%
ItW File	99.85%	Polymorphic	95.59%

AVAST32 was another product to suffer at the hands of a false positive – though in this case a single one. Scanning speeds were particularly fast on the OLE clean set, and pretty good on the executable portion of this test too.

The feeling of ‘almost but not quite’ continued in the detection tests. Once more W32/Nimda.A was missed in the ItW set – here it was missed in the .EML form both on

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
Aladdin eSafe Desktop	0	100.00%	2	99.83%	99.84%	37	99.16%	74	92.47%	29	98.81%
Alwil AVAST32	0	100.00%	2	99.85%	99.86%	18	99.55%	52	95.59%	29	98.46%
CA InoculateIT	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.94%	0	100.00%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	12	99.35%	0	100.00%
Command AntiVirus	0	100.00%	2	99.85%	99.86%	0	100.00%	43	97.50%	2	99.95%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.94%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	8	99.81%
F-Secure Anti-Virus	0	100.00%	3	99.81%	99.82%	8	99.80%	43	97.50%	22	99.69%
GDATA AntiVirusKit	0	100.00%	1	99.91%	99.92%	0	100.00%	43	97.50%	1	99.98%
GeCAD RAV	0	100.00%	0	100.00%	100.00%	0	100.00%	13	99.83%	21	99.20%
Grisoft AVG	0	100.00%	0	100.00%	100.00%	23	99.42%	181	87.85%	56	98.00%
Kaspersky Lab KAV	0	100.00%	1	99.91%	99.92%	0	100.00%	43	97.50%	1	99.98%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	561	92.97%	29	98.65%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	13	99.66%	60	95.48%	13	99.50%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	3	99.81%
Trend PC-cillin	0	100.00%	0	100.00%	100.00%	1	99.99%	234	93.86%	7	99.83%
VirusBuster VirusBuster	0	100.00%	0	100.00%	100.00%	0	100.00%	69	93.33%	8	99.81%

access and on demand. Other samples missed in the ItW set were a single sample each of O97M/Tristate.C on access and a sample of W32/SirCam.A on demand. The missing of W32/SirCam.A is the most concerning, since the others are likely due to extension- or format-related problems, while the missing of W32/SirCam.A is more likely to be due to a defective identity in the virus database.

Other misses were by and large confined to the polymorphics, whether in the polymorphic test set or in the macro or standard test sets. An apology must be made concerning a comment about *AVAST32* in the last *Windows NT* test – contrary to a statement in that review, the product *does* allow browsing for scan targets.

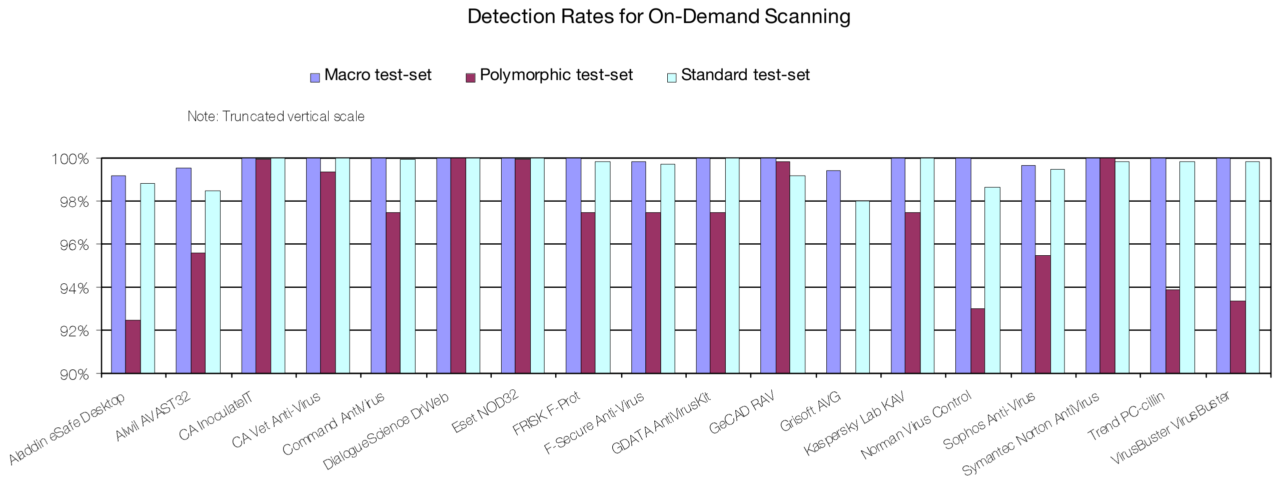
CA InoculateIT 6.0.85

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	98.30%	Standard	100.00%
ItW File	100.00%	Polymorphic	99.94%

The greatest surprise on receiving *InoculateIT* on this occasion was that it was not accompanied by a lengthy patch list – reducing the number of downloads and installation procedures required quite considerably. Though this is good from a reviewer's point of view, this lack of patching may be an irritation to the developers.

The results of on-demand scanning were the usual excellent level demonstrated by *InoculateIT* in recent tests – just one sample of W32/Zmist.D was missed from the entire test set. The matter of on-access scanning was different, however, with a further 16 misses occurring, all of them in .HTM and .HTA extensioned samples.

Such a sudden change is almost certainly related to a configuration issue rather than an inability to detect. This notwithstanding since some of these files were within the ItW set *InoculateIT* misses out on a VB 100% award.
On the other hand, false positives remained absent and speed tests showed there to be no worries for *CA* on that front either.



CA Vet Anti-Virus 10.4.4.1.1740

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	99.35%

Remaining with *Computer Associates*, *CA Vet* is the next to be scrutinized. *Vet* managed once more to produce consistent results over the on-access and on-demand tests and missed only one sample of ACG.A and 11 of W32/Zmist.D. When it first appeared, W32/Zmist.A was heralded as a major challenge to detect – and changes in the variants from A to D were implemented mainly to increase this difficulty. That so many products are offering partial or full detection of this virus is a good sign.

Returning to *Vet* and with no false-positives the first VB 100% award of this comparative can be forwarded to *CA*. A single worry from a reviewer's point of view is that *Computer Associates* now seem to be marketing their products under three rather than two badges – the *CA Vet*, *InoculateIT* and *eTrust* lines of product. If this expansion continues it will not be long before a comparative review is populated by a majority of *CA* products.

Command AntiVirus 4.64.0

ItW Overall	99.86%	Macro	100.00%
ItW Overall (o/a)	99.43%	Standard	99.95%
ItW File	99.85%	Polymorphic	97.50%

After the *Computer Associates* run of products it is time to start on those powered by the *F-Prot* engine, starting with the offering from *Command*. The first area of note comes with the speed of scanning, which is certainly fast and came with no false positives to detract from the performance.

Detection was equivalent for the on-access and on-demand tests, with the majority of misses coming from the samples of W32/Zmist.D once more. However, it is the remaining misses which are more of concern. The .ASP form of W32/Nimda.A remained undetected, as did the sample of W32/Redesi.C. As these are both In the Wild for the test sets used this is sufficient to deny *Command* a VB 100% award on this occasion. As far as mitigating circumstances are concerned there is one comment to be made, in that this product was submitted considerably earlier than any others reviewed at this time.

DialogueScience DrWeb 4.27

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

To deal with the bad matters first, in its traditional manner *DrWeb* gave rise to 15 suspicious files but no false-positives. The speed at which this scanning is performed is good, however, and thus all is well (if not perfect) on this front.

Detection rates to be happy with are also becoming a tradition for *DrWeb* and this was no exception. Once more all files in both the on-access and on-demand test sets were detected correctly and this gains *DrWeb* another VB 100% award for the efforts of *DialogueScience*.

Eset NOD32 1.144

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	99.94%

The testing of *NOD32* started with some odd glitches during the on-access tests – which suffered from the intermittent and non-reproducible misses on access as described in the test procedures information. Performing several scans resulted in different misses each time, usually within or just after the polymorphic test sets. The speed at which these files were processed, by far the fastest of the products

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files	
	Time (s)	Throughput (MB/s)	FPs [susp]	Time(s)	Throughput (MB/s)	FPs [susp]	Time (s)	Throughput (MB/s)	Time(s)	Throughput (MB/s)
Aladdin eSafe Desktop	1305.0	419.1	3	26.0	3051.3		471.0	338.5	36.0	2072.4
Alwil AVAST32	394.0	1388.2	1	16.0	4958.4		69.0	2310.4	37.0	2016.4
CA InoculateIT	367.0	1490.3		18.0	4407.4		107.0	1489.9	21.0	3552.7
CA Vet Anti-Virus	217.0	2520.4		12.0	6611.1		108.0	5064.2	21.0	3777.8
Command AntiVirus	132.0	4143.4		11.0	7212.2		94.0	1695.9	14.0	5329.1
DialogueScience DrWeb	353.0	1549.4	[15]	13.0	6102.6		145.0	1099.4	25.0	2984.3
Eset NOD32	77.0	7103.0		15.0	5288.9		16.0	9963.5	2.0	37303.7
FRISK F-Prot	236.0	2317.5		21.0	3777.8		109.0	1462.5	14.0	5329.1
F-Secure Anti-Virus	2406.0	227.3		56.0	1416.7		1502.0	106.1	458.0	162.9
GDATA AntiVirusKit	249.0	2196.5		36.0	2203.7		135.0	1180.9	47.0	1587.4
GeCAD RAV	663.0	824.9	[1]	37.0	2144.2		278.0	573.4	18.0	4144.9
Grisoft AVG	350.0	1562.7	4 [2]	22.0	3606.1		122.0	1306.7	20.0	3730.4
Kaspersky Lab KAV	254.0	2153.3		33.0	2404.1		159.0	1002.6	47.0	1587.4
Norman Virus Control	2782.0	196.6		16.0	4958.4		294.0	542.2	20.0	3730.4
Sophos Anti-Virus	199.0	2748.4		29.0	2735.6		148.0	1077.1	21.0	3552.7
Symantec Norton AntiVirus	236.0	2317.5		31.0	2559.2		100.0	1594.2	21.0	3552.7
Trend PC-cillin	245.0	2232.4		23.0	3449.3		116.0	1374.3	32.0	2331.5
VirusBuster VirusBuster	324.0	1688.1		31.0	2559.2	[1]	199.0	801.1	31.0	2406.7

on test, seemed a possible cause for these misses (files were, presumably, not being scanned since *NOD32* had built up a large backlog of files waiting to be scanned). Sure enough, by introducing a delay between the file accesses used to trigger detection, the phenomenon was markedly reduced.

With this problem possibly explained, the matter of detection rates could be examined more thoroughly and in the end only one sample of W32/Zmist.D was missed on access and on demand.

No false positives in addition to this performance results in another VB 100% award for *Eset* –which is lucky indeed, for the company states on its CD packaging that *NOD32* has never yet missed a file In the Wild in *VB* tests.

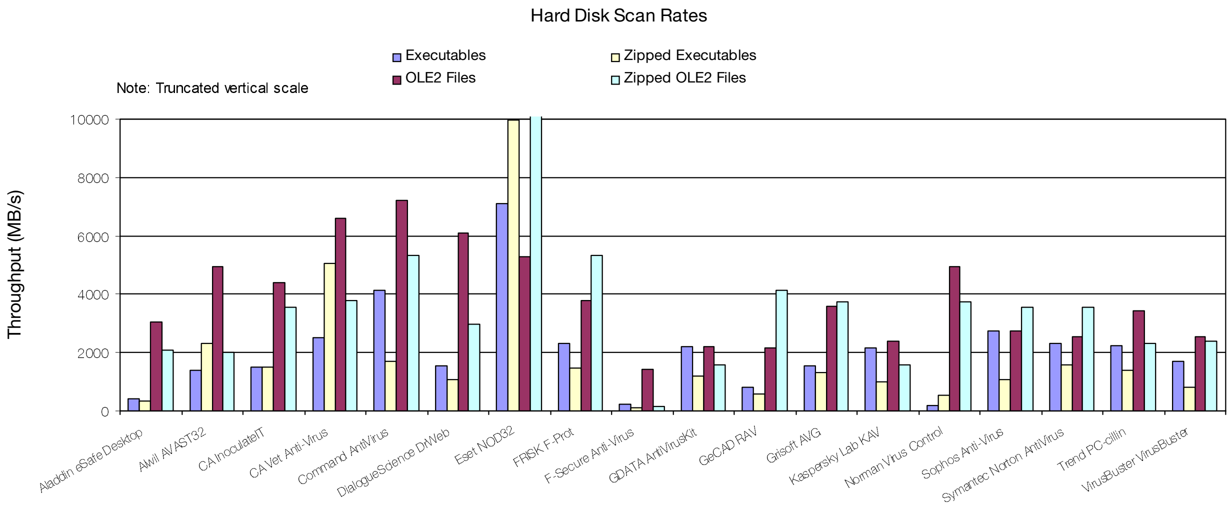
FRISK F-Prot 3.11b

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.81%
ItW File	100.00%	Polymorphic	97.50%

With the same engine as *Command AntiVirus*, but a more recent set of virus information, would the *FRISK* engine fare better in its originator's product than in a third-party product? Oddly enough, in terms of raw speed, it seems that the *Command* product has the edge, leaving detection as the other possible point of differentiation between the products.

F-Prot's detection was indeed different, though not in the way which might be expected. Sure enough, both W32/Nimda.A and W32/Redesi.C were fully detected by *F-Prot*, which is sufficient to entitle the product to a VB 100% award. More mysteriously, though, the samples of W32/Tuareg.B detected by *Command*'s product were missed by *F-Prot*.

The only explanation that springs to mind is that this is related to different time-out or heuristic settings between the two implementations, which might lead to differences when faced with complex polymorphics such as W32/Tuareg.B. Without insider knowledge however, this all remains speculation.



F-Secure Anti-Virus 5.30.7262

ItW Overall	99.82%	Macro	99.80%
ItW Overall (o/a)	99.72%	Standard	99.69%
ItW File	99.81%	Polymorphic	97.50%

With such intriguing differences between the other two *F-Prot*-based products, the third was approached with interest. Results here did little to clarify matters. First, *F-Secure*'s detection rate was lower than either of the others, due mainly to the default non-scanning of a variety of extensions both on access and on demand. The ItW samples of W32/Nimda.A and W32/Redesi.C were all detected, as were the W32/Tuareg.B samples. With such a combination of results, little in the way of a conclusion springs to mind.

Of more relevance, the selection of unscanned extensions included the .BAT and .LNK extensions used by W32/SirCam.A, and as this is in the Wild, no VB 100% award goes to *F-Secure*. Also unhappily for *F-Secure*, their product is vastly slower than the other two *F-Prot*-based products, especially on polymorphic viruses and notably so in the clean test sets.

GDATA AntiVirusKit 10.1.0.0

ItW Overall	99.92%	Macro	100.00%
ItW Overall (o/a)	99.80%	Standard	99.98%
ItW File	99.91%	Polymorphic	97.50%

AntiVirusKit (*AVK*) is another product which shares an engine, this time with the *Kaspersky* products. With a new engine installed by *Kaspersky* in their own product, it remained to be seen how *AVK* would fare. The answer was that no problems related to the engine could be noted, with both speed and detection looking good. Good, however, was not enough to gain a VB 100% award for *AVK*, since the .ASP form of W32/Nimda.A remained undetected. Other misses were entirely relegated to the samples of W32/Zmist.D.

One rather odd feature relates to the on-access scanning of boot sectors. This testing was not available in the last *ME* comparative in which *AVK* was inspected and the solution offered is somewhat imperfect aesthetically. The on-access boot-sector scanner operates by launching the on-demand scanner when boot sectors are accessed and found infected. This works, but seems rather more clumsy than the usual dedicated messaging system.

GeCAD RAV 8.5.80

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	99.92%	Standard	99.20%
ItW File	100.00%	Polymorphic	99.83%

The first of a pair of products to have benefited from a facelift, the new-look *RAV* is both aesthetically and from an ease-of-use point of view, superior to the old. The engine remains the same, but with such major interface changes obvious will there be changes in functionality? The last review saw problems which had all vanished on this occasion, so no lack of improvement can be cited on either detection or usability.

For *RAV*, only one small problem remained – the .EML form of W32/Nimda.A which was undetected on access. There were other misses in the polymorphic sets for a small number of Cryptor and all W32/Zmist.D and a number of standard files, but overall detection has significantly improved since the test this time last year. More impressive still is the change in false positives, down from two false positives and 47 suspicious files to a mere one suspicious file on this occasion. All in all a good result, with just one disheartening miss for the developers to curse.

Grisoft AVG 6.0.313.174

ItW Overall	100.00%	Macro	99.42%
ItW Overall (o/a)	99.96%	Standard	98.00%
ItW File	100.00%	Polymorphic	87.85%

From a new-looking product to one which has seen no outward change for over two years now. Even more impressive is that *AVG* can still be installed using the same CD that shipped all that time ago. The updates, of course, are more than just virus information and contain a replacement for probably most of the internal parts of the applications, yet this is still an impressive longevity for an anti-virus product CD.

To start with the bad, *AVG* managed to throw up four false positives and two suspicious files on the clean test set, thus denying the product any chance of a VB 100% award. This was done in a very respectable time, however. Detection was marred on access only for the In the Wild set – where the extensionless sample of O97M/Tristate.C was missed. This and the misses of all .MDB files in the test set are likely to be extension- rather than content-based lapses in detection. With regard to detection in the other test sets, *AVG* is somewhat weak with regards to the various polymorphic *Win32* viruses in the test set, though shows no serious weaknesses elsewhere.

Kaspersky Lab KAV 4.0.1.54

ItW Overall	99.92%	Macro	100.00%
ItW Overall (o/a)	99.80%	Standard	99.98%
ItW File	99.91%	Polymorphic	97.50%

Kaspersky Anti-Virus is the second product to have undergone a great change in appearance recently – and in this case too, the change is all for the better. It may lie purely in the realms of aesthetic subjectivism, but the product did feel nice to use. The comparative test is not about such affairs, though, so we move on to the more objective ratings.

The *Kaspersky* product performed well, though with a slightly lower rate of data throughput than the *AVK* product which also houses a *Kaspersky* engine. On matters of detection, all W32/Zmist.D samples were missed both on access and on demand, though the other missed samples were of more interest if fewer in number. On demand, the .ASP sample of W32/Nimda.A was missed as the sole remaining file. On access this was also missed but in addition the .PPT and .POT samples of O97M/Tristate.C were undetected. It is clear that this is an extension scanning decision, nevertheless these misses deny *Kaspersky Anti-Virus* a VB 100% award.

Norman Virus Control 5.2

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	98.65%
ItW File	100.00%	Polymorphic	92.97%

Norman Virus Control continues to frustrate with its lack of any log-producing facility. Quite why this should be the case is a mystery, though this is easy to circumvent by undocumented means. Also perplexing is the continued matter of the slow scan rates on the executable portion of the clean test set, a problem which does not occur if the same files are archived and then scanned. On the scanning of compressed executables and on raw and compressed OLE files the throughput is at much more respectable levels.

Despite these troubles there were no false positives – leaving the detection rates as the arbiter of the VB 100% award. Oddly enough, detection rates for some areas seem to have plummeted since the last reviews of *Norman Virus Control*, with the polymorphics Uruguay.4 and Sepultura:MtE-Small being missed where before they were detected. Thankfully for *Norman*, however, these misses were not present in the ItW test sets, where full detection merits another VB 100% award.

Sophos Anti-Virus 3.53

ItW Overall	100.00%	Macro	99.66%
ItW Overall (o/a)	100.00%	Standard	99.50%
ItW File	100.00%	Polymorphic	95.48%

On this occasion, *Sophos AntiVirus* proved quite an entertaining product to test. To deal with the dull but worthy matters first, the clean test sets were scanned and produced no false positives in the process while producing good throughput rates. On-demand testing was also much as expected. *SAV* missed those files it usually misses (files where detection is only supported under a full mode of scanning), but did detect some files which it was hitherto incapable of.

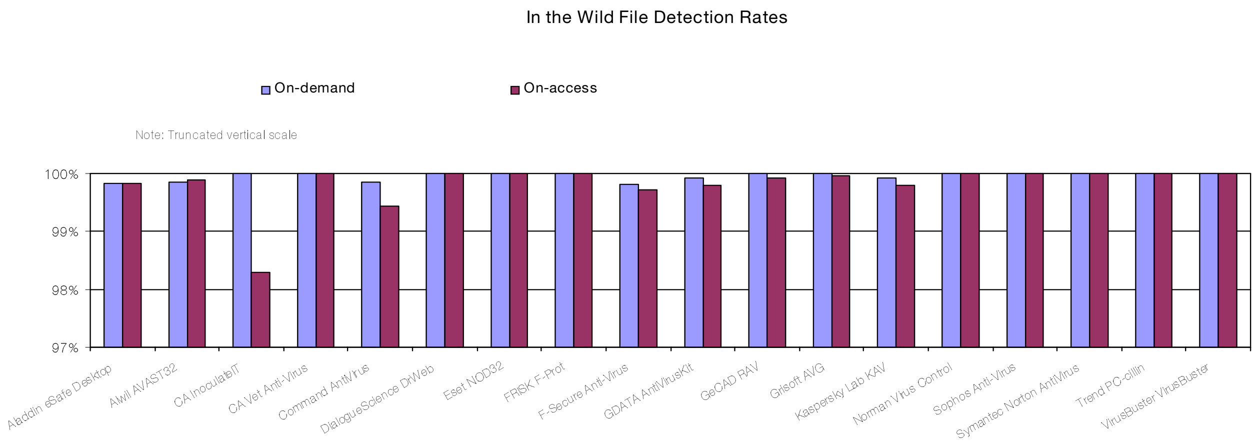
Momentary problems lay in the on-access test, where the samples of W32/Maldal.C and the .HTM sample of W32/Haptime.D were undetected in addition to those files undetected on demand, but further testing showed this problem to be non-reproducible. Due to the transient nature of the problem it was not enough to deny *SAV* a VB 100% award on this occasion.

Symantec Norton AntiVirus 2002 8.00.58

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.81%
ItW File	100.00%	Polymorphic	100.00%

Norton AntiVirus 2002 is the *Norton* home-user offering, and thus was a novel experience in terms of testing when compared with the usual corporate fare. Unfortunately, the freshness of the experience was marred by a lack of features which are expected in the corporate environment.

Lack of logging required that on-demand testing was performed by the deletion of infected files, while the testing of the on-access scanner required some undocumented



tweaks. However, detection was such that *NAV* is eligible for a VB 100% award.

To its credit it should be stated that *NAV* detected all of the files in the test set but Goldbug, had no false positives and was quite speedy on the clean set tests. It was also very easy to use – though this was at the expense of flexibility and configuration.

Trend PC-cillin 2000 7.61.0.1437.195

ItW Overall	100.00%	Macro	99.99%
ItW Overall (o/a)	100.00%	Standard	99.83%
ItW File	100.00%	Polymorphic	93.86%

As a potential home-user product, *PC-cillin* represents the other side of the coin – having most of the features seen in its server-based counterparts, but being more daunting to behold than some home users might be able to accept.

Starting with the clean set tests, these were all completed without false positives and in a speed slightly better than the average. (It is notable that, with a few exceptions which stand out suitably, the scanning speeds seen in the comparatives are becoming more and more clustered about a central point, thus the preponderance of ‘about average’ comments made when referring to scanning speeds.)

Moving on to detection rates, *PC-cillin* performed much as it has done recently on other platforms. The misses consisted of a single *Excel* polymorphic sample, with the remainder being spread amongst the executable polymorphics.

The polymorphics are thus an area where *Trend*’s scanner does have room for improvement. Those polymorphics which are In the Wild, however, were perfectly detected, suggesting that this is an area where research is applied to threats rather than in a blanket manner. This detection is also quite sufficient for *PC-cillin* to gain a VB 100% award.

VirusBuster VirusBuster 3.08

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.81%
ItW File	100.00%	Polymorphic	93.33%

Last in the line-up comes *VirusBuster*. One point that springs to mind to rant about is the small default size of the log which, at 50 KB, is barely enough for the general information passed to it. However, the performance of *VirusBuster* was good.

Clean set scanning rates were in that popular ‘average’ position, with only one suspicious file found. This did have some claim to originality since it was in the OLE set, while almost all other false-positives occur in the executable portion of the test sets. Not so uncommon are those viruses where *VirusBuster* missed detection. W32/Zmist.D, ACG.B, W95/SK8044 and W95/SK7972 have all been missed by a number of products. Despite these polymorphic misses however, there were no misses in the macro and ItW test sets and thus *VirusBuster* brings the review to a close with a VB 100% award.

Conclusion

In conclusion this review was almost too simple – nearly all problems encountered were overcome easily and the products themselves were universally friendly. I hope the same is true for *VB*’s first *Linux* Comparative, which is due in two months.

Technical Details

Test environment: Two 750 MHz AMD Duron workstations with 64 MB RAM, 8 and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, all running *Windows ME*. The workstations and test sets were rebuilt from image back-ups after each test.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/WinME/2001/12test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

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.

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, 6 Kimball Lane, Suite 400, Lynnfield, MA 01940, USA

Tel (781) 9731266, Fax (781) 9731267

END NOTES AND NEWS

The VI Ibero American Seminar on Security Information and Communications Technologies runs in Havana, 18–24 February 2002. Topics covered will include anti-virus software, network security, Web security and network remote diagnostics. Contact José Bidot: email jbidot@seg.inf.cu.

CeBIT 2002 runs in Hannover, Germany, from 13–20 March 2002. As a result of rearranging the individual topic clusters, easier navigation and a more compact, convenient layout is promised for visitors to CeBIT 2002. For details including the preliminary list of exhibitors, or to order tickets online, see http://www.cebit.de/.

Cost-Effective Risk Management for Information Security takes place at the Café Royal, London, 19–20 March 2002. Blue chip corporate case studies and expert organizations will examine the strategic issues surrounding cost-effective risk management for information security. For more details tel: +44 207 368 9300 or visit http://www.iqpc.co.uk/GB-1759/ediary/.

The 2nd Security Audit & Control of Information Systems Conference and Expo (SACIS) will be held 19–20 March 2002 in Istanbul, Turkey. Topics will include Internet/Intranet security, computer crime, denial of service attacks, forensic investigation, intrusion detection and email security. For more details email svs@svs.com.tr or visit the Web site http://www.smartvalley.net/sacis/.

Information Security in the Age of Terrorism takes place 25–26 March 2002 in Washington, D.C. Hear from a stellar faculty about the latest threats to information security and how to combat those threats. For more information visit http://www.frallc.com/, or email sldowt@aol.com.

Information Security World Asia 2002 will be held 16–18 April, 2002 in Singapore. The show will include an exhibition, and a number of interactive workshops. For further information visit the Web site http://www.isec-worldwide.com/isec_asia2002/.

Infosecurity Europe 2002 will run from 23–25 April 2002 at London's Grand Hall, Olympia. Over 40 free seminar sessions will run over the three days, explaining some of the key security issues facing organizations today. For more details visit the Web site at http://www.infosec.co.uk/.

The Southwest CyberTerrorism Summit, to be held 4 May, 2002 in Dallas, TX, USA, will feature presentations from both hackers and industry security experts. Topics include wireless hacking, cyber-attacks, information warfare, privacy, computer viruses, industrial espionage and identity theft. For more information visit the Web site http://www.DallasCon.com/.

The 11th Annual EICAR Conference & 3rd European Anti-Malware Forum takes place 8–11 June, 2002 in Berlin, Germany. For more details of the event see the EICAR Web site http://www.eicar.org/.

Information Security World Australasia 2002 will be held 19–21 August, 2002 in Sydney, Australia. The conference and exhibition represent the region's largest dedicated IT security show. For full details see http://www.informationsecurityworld.com/.

***Network Associates Inc.* is to transfer its stock listing to the New York Stock Exchange**. Says *NAI* Chairman and CEO George Samenuk, 'Moving to the New York Stock Exchange will increase *Network Associates*' visibility with a wider base of investors in domestic and international markets.' The company will begin trading on NYSE from 12 February, 2002. For more information see http://www.nai.com/.

***F-Secure* has issued a hotfix** for the bug in its *F-Secure Anti-Virus* version 5.30 software that can cause system crashes in *Windows*. To read the advisory see http://www.f-secure.com/support/top-issues/.

***Kaspersky Labs* and *Ernst & Young* are to begin a collaboration** for the development and delivery of prepared information security solutions to end users in Russia and other countries within Commonwealth of Independent States. See http://www.kaspersky.com/.

Sales of *Symantec*'s anti-virus software have contributed to better than expected third quarter results for the company, with sales of anti-virus software up 53 percent from the same quarter last year. *Symantec Corp.* posted third quarter revenue of $290.2 million, compared with $241.8 million for the quarter last year. Chairman and CEO of the company John W. Thompson attributed the better than expected results in part to a general heightened awareness for security. For more details see http://www.symantec.com/.