ISSN 0956-9979 SEPTEMBER 2001

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

- **Buffer Zone:** After the initial analyses of Code Red, it occurred to Bruce McCorkendale and Péter Ször that none detailed exactly how the buffer overflow worked. The two took matters into their own hands and investigated. Their analysis starts on p.4.

- **Keeping up with the Joneses:** As the AV industry heads towards the day when all vendors have become Application Service Providers, Joe Wells ponders the question ‘should AV testing be revised to keep up with the changes in the way the products are used?’ See p.12.

- **Slipping the NetWare:** One year exactly since the last *NetWare* Comparative Review, we look again at the same offerings, with a couple of additions, to see how things have moved on. See p.17.

CONTENTS

COMMENT

NEWS

VIRUS ANALYSES

FEATURE

OPINIONS

COMPARATIVE REVIEW

COMMENT

“ As we sat in our ivory towers of academe, SirCam came charging again and again ”

Report from the Front

It was a fairly relaxed summer. The students were away from campus. And while there was always work on my desk, I actually had time to dig down deeper in the pile than usual. Or so I thought …

Ever since 1988, when the University of Michigan’s anti-virus team was formed, I have received real and suspected virus samples. Most of these have been files forwarded from end users asking, ‘Is this file a virus?’ Occasionally, as with W32/Ska@m or JS/Kak@m, I would receive a plea for help, containing or followed/preceded by an infected attachment, but in each of these cases, a human made contact with me intentionally, rather than a virus itself seeking me out. With 13 years of service in the anti-virus wars, I’d expect to be in more than a few end user address books. I prided myself, when W97M/Melissa.A@mm and VBS/LoveLetter.A@mm and their ilk struck that, since I didn’t see a single natural copy of these viruses, perhaps people had heeded my call never to accept unsolicited attachments and to use properly updated anti-virus software.

On 18 July, I announced the regular anti-virus updates to the University community, and noted that the newly recognized viruses seemed not to contain anything particularly threatening. All was well for about a week … Then the walls started caving in. My mailboxes started seeing a huge increase in volume, almost all of which was coming in from outside. As we sat in our ivory towers of academe, W32/SirCam@mm came charging again and again, carrying a lance weighing at least 200 KB (and I’ve seen 5 MB) – and he was no knight in shining armour. (*For a full analysis, see Péter Ször and Peter Ferrie’s analysis, p.8 - Ed*.)

On 23 July when it first became clear that an outbreak was in progress, I quickly sent out an alert, put up http://www.umich.edu/~virus-busters/sircam.html, created Unix scripts and DOS BATch files to allow me to open the documents sent by SirCam, and wrote a standard template to email to victims. Then I spent several 18-hour days handling the onslaught: I’ve processed 200+ MB of files in 700+ infected messages from about 200 different victims, most of whom were not and never had been members of the University. The initial work to create utilities was worthwhile, and the high response rate from victims or their ISPs has been gratifying.

Colleagues at large organizations (understandably) put up filters at their gateways, and saw little of this directly. We took a different approach; I kept my personal portcullis open. Why? Because it is clear that many people outside of large organizations have been infected with SirCam, are spreading their private documents far and wide – and consuming lots of bandwidth. Often there are clues in the original documents that allow one better to determine the victim, even when the sending address is ‘forged’, so it’s possible to find the victim.

The attachments I’ve received have ranged from banal (e.g. a MIDI version of the doo-wop hit ‘Rama Lama Ding Dong’) to disastrous: a client list, including credit card numbers and expiry dates. This highlights the importance of contacting the victims rather than merely blocking – who knows how many copies of this file were sent, and to whom.

I suspect that this virus shuts itself down, to a degree, by filling the victim’s mailbox with messages SirCam sent to invalid addresses, and with replies like ‘What was this you sent me??’ – it takes five or fewer of these to use a MB, and some ISPs have small Inbox quotas. But more needs to be done than merely disabling a mailbox: the victim must be contacted and the virus destroyed.

How do we do this, in general? Surely I can’t do it alone, and not all of us will have the tools or expertise to handle the next outbreak – nor will the small ISP. It seems that the logical place to stem such attacks is at the service provider, but when such efforts fail, there needs to be a group of defenders who will let the virus in, track its origin, and then, like real knights, come to the rescue.

Oops; gotta run. Seven new SirCams just hit my mailbox.

Bruce P Burrell, University of Michigan, USA

NEWS

Leaves on the Line

The suspected author of the W32/Leaves worm, variants of which posed as a bogus *Microsoft* Security Bulletin, has been arrested following a joint operation between Scotland Yard and the FBI. The South London man will appear in court at the end of this month. If found guilty, he faces a maximum sentence of five years imprisonment ■

A Wolf in Sheep's Clothing

Playing a similar game to that of the aforementioned Leaves variants, Win32/Allgro-A is a mass-mailing worm which sends itself using MAPI functions and appears to be an anti-virus device. The worm arrives as an attachment (antivirus.exe) to an email whose subject is 'New Antivirus tool' and whose text claims that the attachment 'checks your system for viruses'. Once launched the worm copies itself into the Windows System directory, then makes changes in the Registry so that the worm runs on starting up *Windows*. The worm attempts to remove viruses such as W32/SirCam.A, W32/Pretty or W32/Badtrans.A from the system and, on 16 September, the worm displays the message 'System protected by I-Worm.Antivirus Copyright [c] 2001 by aLL3gRo.' Whether or not this worm was written with malicious intent, its method of propagation by masquerading as some form of security alert seems to be becoming increasingly relevant. We may never see an end to the days of disguising malware as some form of soft pornography, but the promise of a security update is proving to be almost equally alluring ■

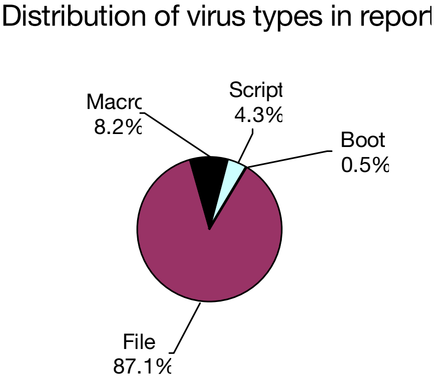
Red or Dead?

After cries of 'Time is running out for businesses to protect themselves against Code Red!' and 'Act now to prevent the meltdown of the Web!', along with warnings from the FBI, and general hysteria in the media, Code Red proved to be a bit of a damp squib. None of the catastrophic effects that had been predicted materialised, even with the advent of the 'even more destructive' Code Red II. It is thought that the reason the Web did not go into meltdown was that sufficient numbers of people heeded the various government warnings and applied the patch from *Microsoft*, closing the security hole in IIS 4.0 and 5.0. We can only assume that *Microsoft* was too busy preaching to practise, as the small matter of applying the patches to its own Hotmail servers was overlooked … Throughout the hysteria, those in the AV industry remained calm and level headed about the whole thing; David Perry of Trend Micro told the Associated Press: "I would suggest that because of Code Red, there's no reason to go out and buy mass quantities of beef jerky." VB wonders whether there is ever reason enough to go out and buy beef jerky ■

Prevalence Table – July 2001

Virus	Type	Incidents	Reports
Win32/SirCam	File	1973	46.7%
Win32/Magistr	File	701	16.6%
Win32/Hybris	File	504	11.9%
Win32/MTX	File	245	5.8%
Laroux	Macro	123	2.9%
Win32/BadTrans	File	66	1.6%
Kak	Script	55	1.3%
Divi	Macro	49	1.2%
Marker	Macro	48	1.1%
Win32/Funlove	File	47	1.1%
Haptime	Script	46	1.1%
VBSWG	Script	44	1.0%
Win32/QAZ	File	40	0.9%
LoveLetter	Script	28	0.7%
Win32/Choke	File	26	0.6%
VCX	Macro	23	0.5%
Tristate	Macro	14	0.3%
Win95/CIH	File	14	0.3%
Cap	Macro	13	0.3%
Solaris/Sadmind	File	11	0.3%
Win32/Navidad	File	11	0.3%
Class	Macro	9	0.2%
Ethan	Macro	9	0.2%
VMPCK	Macro	9	0.2%
Win32/BleBla	File	9	0.2%
Win32/Ska	File	9	0.2%
Others		101	2.4%
Total[1]		**4227**	**100%**

[1] The Prevalence Table includes a total of 101 reports across 43 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.



SirCam Round-Up

While media darling Code Red was turning into something of a non-event, the spread of SirCam was quietly reaching epidemic proportions.

One oddity of the SirCam invasion has been the way in which some attachments have arrived at *Virus Bulletin*. The advice given by *Virus Bulletin*'s technical staff to one user on W32/MTX infections had clearly been taken to heart at least partially – since they had bothered to save and convert it to .DOC format. Those, including *VB*, who then received that document as a SirCam-infected attachment might well have been grateful for the advice, but not so grateful for its unwelcome addition.

More intriguing still was the arrival of an outraged email in *VB*'s inbox, accusing *Virus Bulletin* of sending out infected documents. The giveaway in this case was that the document in question came, apparently, from a member of our staff no longer in residence – a certain Mr Fitzgerald. The true source of this attachment had previously been infected by W97M/Brenda, which, among other actions, changes the user mail address in the registry to 'Nick@virusbtn.com'. Since SirCam uses the information in the registry when constructing mail headers, the true source of the mail, and the virus, was totally obscured.

Higher profile victims of the worm have included the FBI and the Ukrainian Presidential administration, whose secret documents – including a timetable for President Kuchma's movements on Ukraine's 10th anniversary of independence – were identified when received by Ukrainian ForUm news Web site.

But, despite its malicious nature, SirCam has also inspired creativity in those whose lives it has touched: staff at *VB* were rendered speechless when 'Worm War Won', a moving poem, arrived in an email from Sybari. A work of literary genius? Well, let's just say the Poet Laureate need not lose any sleep ▮

Sittin' on the Beaches

Adobe Acrobat, another application formerly considered to be 'safe', has fallen to the machinations of the virus writers. This time not as a virus but as another vector to get to the desktop. The triplets VBS.PeachyPDF.{A,B,C} are, as their names suggest, Visual Basic Scripts rather than PDF viruses. More accurately they are, respectively, an embedded VBS file, VBE file and WSF file. The code in each differs in respect to references to the file's size.

The virus is not currently in the Wild, and is unlikely to become widespread since, in order to function properly, the virus requires not only *Adobe Acrobat Reader* but a full version of *Adobe Acrobat 4.0* or above – significantly less common among PC users. This, combined with the warning box that comes up, should mean that this method of attack is little more than a passing curiosity ▮

VIRUS ANALYSIS 1

Code Red Buffer Overflow

Bruce McCorkendale and Péter Ször
Symantec Corporation

[Having encountered conflicting information from a variety of sources about the Code Red (aka W32/Bady.worm) buffer overflow technique, Bruce McCorkendale and Péter Ször decided to look into it themselves. Here, as a follow-up to Costin Raiu's analysis in last month's issue, they present their findings - Ed.]

Analyses of the Code Red worms to date have either skipped over the details of the buffer overflow, or they have given incorrect details. Noticing this, we were inspired to dig into the buffer overflow to uncover the details.

Setup

The IIS web server receives GET /default.ida? followed by 224 characters, URL encoding for 22 Unicode characters (44 bytes), an invalid Unicode encoding of %u00=a, HTTP 1.0 headers and a request body.

For the original Code Red worm, the 224 characters are N; for the most recent version of the worm, they are X. In all cases, the URL encoded characters are the same (they look like %uXXXX, where X is a hex digit – they have been published in previous analyses). The request body is different for each variant.

IIS keeps the body of the request in a heap buffer (probably the one it read into after processing the content-length header indicating the size to follow). Note that, despite the fact that a GET request is not allowed to have a request body, IIS dutifully reads it according to the header's instructions.

Buffer Overflow Details

While processing the 224 characters in the GET request, functions in IDQ.DLL overwrite the stack at least twice – once when expanding all characters to Unicode, then again when decoding the URL escaped characters. The original *eEye* exploit demonstration probably uses one of these previous overwrites, but we have not seen this. *eEye* claims that they are overwriting the return address, which suggests that control is transferred to their shellcode when a RET instruction is executed. However, the overwrite that results in the transfer of control to the worm body happens when IDQ.DLL calls DecodeURLEscapes() in QUERY.DLL.

The caller of the DecodeURLEscapes() function is supposed to specify a length of the data buffer to decode in wide characters, but instead specifies a number of bytes. As a result, DecodeURLEscapes() thinks it has twice as much room as it actually has, so it ends up overwriting the stack. Some of the decoded Unicode characters specified in URL encoding end up overwriting a frame-based exception block. Even after the stack has been overwritten, processing continues until a routine is called in MSVCRT.DLL. This routine notices that something is wrong and throws an exception.

Exceptions are thrown by calling the KERNEL32.DLL routine RaiseException(). RaiseException() ends up transferring control to KiUserExceptionDispatcher() in NTDLL.DLL. When KiUserExceptionDispatcher() is invoked, EBX is pointing to the exception frame that was overwritten. (That EBX is pointing here is a side-effect of the immediately previous processing.) The exception frame is composed of four DWORDs, the second of which is the address of the exception handler for the represented frame.

The URL encoding whose expansion overwrote this frame starts with the third occurrence of %u9090 in the URL encoding, and is as follows:

%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3.

This decodes as the four DWORDs: 0x68589090, 0x7801CBD3, 0x90909090 and 0x00C38190. The address of the exception handler is set to 0x7801CBD3 (second DWORD), and KiUserExceptionDispatcher() calls there with EBX pointing at the first DWORD via CALL ECX.

The instruction CALL EBX is at address 0x7801CBD3 in MSVCRT.DLL. When KiUserExceptionDispatcher() invokes the exception handler, it calls to the CALL EBX, which, in turn, transfers control to the first byte of the overwritten exception block. When interpreted as code, these instructions find and then transfer control to the main worm code, which is in a request buffer in the heap.

The author of this exploit needed the decoded Unicode bytes to function both as the frame-based exception block containing a pointer to the 'exception handler' at 0x7801CBD3, and as runable code. The first DWORD of the exception block is filled with four bytes of instructions arranged so that they are harmless, but also place the 0x7801CBD3 at the second DWORD boundary of the exception block. The nop, nop, pop eax, push 7801CBD3h accomplish this task easily.

Having gained execution control on the stack (and avoiding a crash while running the 'exception block'), the code finds and executes the main worm code.

This code knows that there is a pointer (call it pHeapInfo) on the stack 0x300 bytes from EBX's current value. At pHeapInfo+0x78, there is a pointer (call it pRequestBuff) to a heap buffer containing the GET request's body, which contains the main worm code. With these two key pieces of information, the code transfers control to the worm body in the heap buffer. The worm code does its work, but never returns – the thread has been hijacked (along with the request buffer owned by the thread).

Conclusion

This technique of usurping exception handling is complicated (and crafting it must have been difficult). The brief period between the release of the *eEye* description of the original exploit and the appearance of the first Code Red worm leads us to believe that this technique is somewhat generic. Perhaps the exception handling technique has been known to a few buffer overflow enthusiasts for some time, and this particular overflow presented the perfect opportunity to use it.

Having exception frames on the stack makes them extremely vulnerable to overflows. This is an oversight in the current OS implementations, but *Windows XP* provides a new 'Vectored Exception Handling' feature that could allow exception frame data to be kept on the heap (however, current compilers only use stack-based exception frames).

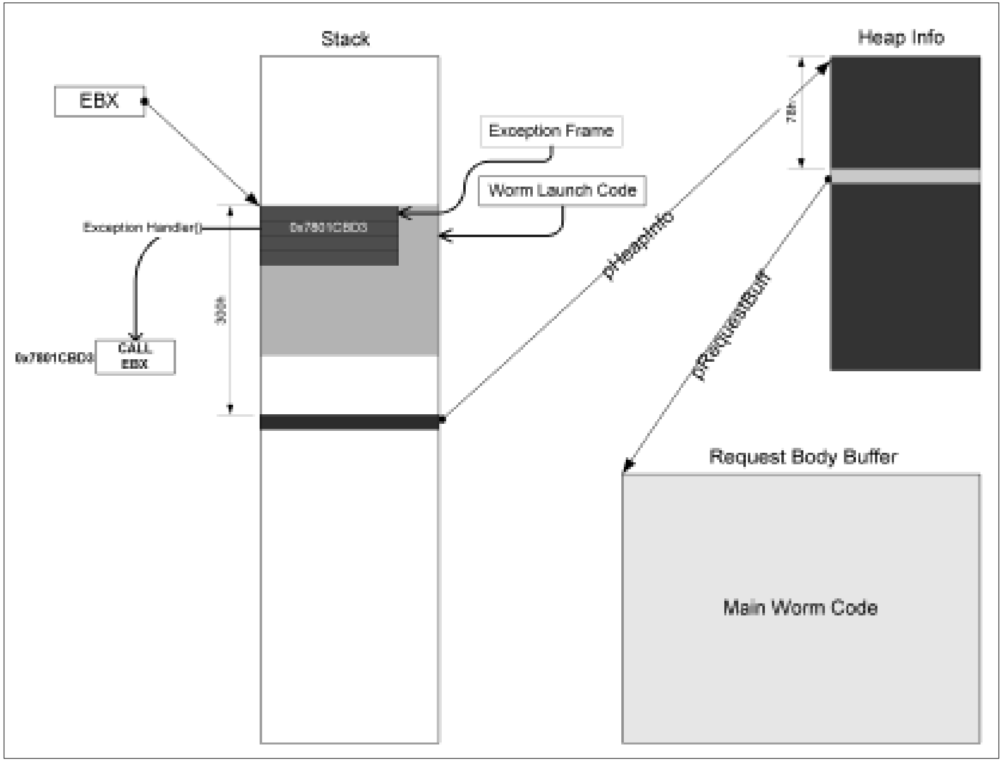


Figure 1: Stack, Heap, and Frame Layout.

VIRUS ANALYSIS 2

Streaming Infections

Costin Raiu
Kaspersky Lab, Romania

About a year ago, two Czech virus writers put some of their time into the making of the first computer virus to take advantage of the NTFS facility called 'streams'. Win2K/Stream.3628 infects PE files by replacing their body with a copy of itself, while saving the original content of the file in an additional stream within the NTFS structure of the file. However, this technique did not become popular, and no similar viruses were written …until recently.

Potok?

'Potok' is a Russian word, meaning 'stream', but what interests us most is that 'Potok' is also the name of a recent mass-mailing VBS virus, which happens to use NTFS streams.

Written by someone who goes by the nickname 'Lord Nikon', the virus is not extremely complex – there are only about 253 lines in the main VBS source, of which a good deal are comments. Most of the comments are part of a 'development diary' of the author, which starts on 21st July. The last entry in the log is dated 30th July, on which date the author mailed his creation to a selection of AV developers around the world. Apparently, the virus author used code pieces from the VBS samples included in an MSDN article called 'A Programmer's Perspective on NTFS 2000' which, among other things, contains some hints on how to recognize NTFS volumes from VBS, and how to operate with streams.

Arrival

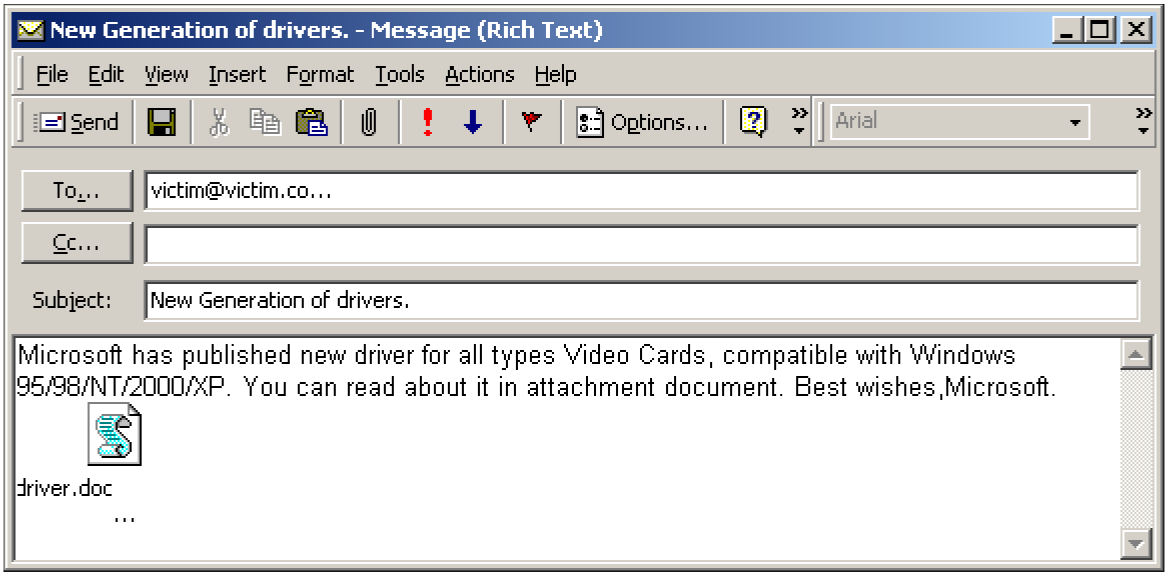
Like most self-mailing VBS viruses, VBS/Potok arrives in the form of an email message which has the virus code attached. The attached filename in infected emails is 'driver.doc{46 spaces}.vbs'.

When run, the first thing the 'driver.doc[...].vbs' file does is to make a copy of itself in the *Windows* directory. This copy will be used further along the mass-mailing step. Then, if not already present, it will create an empty file named 'odbc.ini' in the same folder, and it will check whether the drive hosting the virus copy is an NTFS-formatted volume. If the drive is not NTFS-formatted, the virus simply exits at this point. However, if the drive is NTFS, it will proceed to dump four other scripts in four associated streams of the 'odbc.ini' file, named 'main', 'mail', 'user' and 'group', each of which holds various pieces of the virus code.

After that, the virus creates another file in the 'system32' subdirectory of the operating system root. The file, named 'go.vbs', contains code which is designed to reverse the virus source splitting in the above-mentioned four streams.

After creating 'go.vbs', the main virus component waits ten seconds, then proceeds to execute it. So, 'go.vbs' carries the infection process further. It creates (yet!) another file, named 'notepad.vbs', in the 'system32\ras' directory and fills it with the contents of the 'main', 'mail', 'user' and 'group' streams of the 'odbc.ini' virus holder. Next, it waits another ten seconds, and passes the control to 'notepad.vbs'.

Now, if you think that 'notepad.vbs' creates a file again, and dumps various things into it, sleeps for ten seconds,


New Generation of drivers. - Message (Rich Text)
File Edit View Insert Format Tools Actions Help
Send
Options...
Arial

To... victim@victim.co...

Cc...

Subject: New Generation of drivers.

Microsoft has published new driver for all types Video Cards, compatible with Windows 95/98/NT/2000/XP. You can read about it in attachment document. Best wishes,Microsoft.

driver.doc
...

Figure 1: VBS/Potok.A infected email.

then executes it, you are wrong. In fact, ‘notepad.vbs’ is the last file in this process, and it holds the virus replication code as well as some other code (explained below). The replication code will instantiate an ‘Outlook.Application’ object, in much the same way as the LoveLetter virus does. For the first 50 entries in the first *Outlook* address list it will create emails with a message supposedly from *Microsoft*, the subject ‘New Generation of drivers’, and with the ‘driver.doc[...].vbs’ file attached.

Next, the virus attempts to backdoor the local machine. For this, it tries to create an account named ‘Lord_Nikon’ with the password ‘password’, which will only work if the account running the virus has administrative privileges. In addition it will attempt to add the ‘Lord_Nikon’ account to ‘Administrators’ as well as the Russian-equivalent of the ‘Administrators’ user group. As an interesting detail, the virus attempts to create the ‘Lord_Nikon’ account, as well as executing the other account-related operations, first through the use of two objects, named ‘WinNT://server’ and ‘WinNT://server/group’. On my test machines this operation failed, but the virus contains a second method by which to accomplish this task, based on the external ‘net.exe’ application. It runs two commands, ‘net user Lord_Nikon password /add’ and ‘net localgroup Administratotrs Lord_Nikon /add’, the first of which works quite well , therefore ‘backdoor-ing’ the system with the ‘Lord_Nikon’ account. However, as you can see, the author included a typo (‘Administratotrs’) in the second command, therefore the ‘Lord_Nikon’ account will not be added in the Administrators group.

After that, ‘notepad.vbs’ terminates execution, and along with it, the virus execution cycle is finished. Since the virus keeps no track of the mails it sends, executing it again on the same machine will prompt it to attempt to send itself via email one more time, as well as attempting to perform all the other ‘backdoor-ing’ operations.

Given that the two ‘net…’ commands open two Win32 console windows, it’s less likely that the virus will go unnoticed, and the fact that it doesn’t delete ‘go.vbs’ or ‘notepad.vbs’ also decreases its chances of remaining undetected by AV scanners.

An interesting point is that most of the virus dropping/recombination routines are massively split into pieces using the VB operator ‘_’. This must have been used as an anti-heuristic measure but, unfortunately for ‘Lord Nikon’, many AV products have no problem in detecting it heuristically, regardless of the line splitting.

Detection/Removal

Hopefully, by now, most AV products will have been updated to search within the NTFS streams associated with a file. In this particular case, there is no other chance to detect if the ‘odbc.ini’ file has been touched by the virus, and cleaning it also requires the product to wipe the ‘main’, ‘mail’, ‘user’ and ‘group’ streams. Besides that, detection and removal of the ‘go.vbs’ and ‘notepad.vbs’ files is trivial. One should also keep in mind that a machine infected by the virus might also have an extra ‘Lord_Nikon’ user account, so checking user accounts is a must after a compromise by ‘Potok’.

The Virus Author’s Diary

As mentioned in the introduction, the virus contains a couple of comments from the author, apparently written during the time he was developing the code. My thanks go to Dmitry Gryaznov from *NAI*, who took some time to translate the comments and provided further information regarding some of the details from the ‘diary’. ‘Lord Nikon’ claims that he started coding the first version of the virus around July 16th, and finished it on July 30th. During those two weeks, he seems to have worked through the nights, or at quite early hours, if we are to believe some of the entries in the log. I’d say, given the amount of time the author seems to have needed to finish the virus, he/she was probably a novice in VBS programming, with very little previous experience.

Conclusions

‘Potok’ is not necessarily a breakthrough in virus development, but it shows an interesting path which may lead to more complicated things in the future. At the time of writing, the vast majority of computer users are not running NT/2K systems, and some of those actually running such machines still have FAT/FAT32 drives. Therefore, I believe that, right now, a virus which depends on NTFS streams to replicate has little opportunity to reach the level of spread of a virus like LoveLetter or VBSWG.J. However, this kind of technique can be used as an addition to the main operations of viral code to complicate detection and disinfection.

I’d say that, if an anti-virus product doesn’t support NTFS streams scanning and disinfection, it’s not such a big problem, at least not yet. But, with more viruses attempting to use them, and with the forthcoming *Windows XP*, an NT-based OS which supports NTFS and NTFS streams, it’s as well to keep an eye on the viral developments in this area, just in case.

VBS/Potok

Aliases:	VBS/Stream.
Type:	Mass mailing VBS virus.
Payload:	Backdoors the system by creating a user named ‘Lord_Nikon’.
Removal:	Remove ‘system32\go.vbs’, ‘system32\ras\notepad.vbs’, and the streams ‘main’, ‘mail’, ‘user’ and ‘group’ from the ‘odbc.ini’ file from the Windows directory. Also, delete the ‘Lord_Nikon’ user account.

VIRUS ANALYSIS 3

SirCamstantial Evidence

Peter Ferrie and Péter Ször
Symantec Corporation, USA

Although SirCam made a name for itself sending out random files and personal documents from infected PCs, not all of the information that spread with Win32/SirCam was spread by the worm itself. Almost as soon as updated descriptions of SirCam were posted to Web sites, selected texts from these descriptions appeared on other sites, complete with identical spelling errors and inaccuracies.

Evidently the emerging complexity of new 32-bit worms is proving a tough challenge for every one of us in this business: if ExploreZip was boring and difficult to analyse, SirCam was a major pain. SirCam's author tried to make sure that the analysis would not be straightforward. The worm is written in a high-level language, but all the string constants (including its email message) are encrypted in such a way that it took a little while to decrypt completely (at least for some of us).

Start Your Engines

Win32/SirCam usually arrives as an attachment to an email. This attachment is special, because it contains not only SirCam itself, but an additional file (attached to the end of SirCam), which has been 'stolen' from the Personal or Desktop directory of the sender's computer.

When this attachment is run, SirCam will detach the stolen file and display it. The way in which the file is displayed depends on its suffix. If the suffix is .doc, SirCam will attempt to run *WinWord*. If this fails, then *WordPad* will be used instead. If the suffix is .xls, SirCam will run *Excel*. If the suffix is .zip, SirCam will run *WinZip*. If the suffix matches none of these, SirCam will run rundll32. Even in the event that no suitable application can be found to display the file, SirCam will install itself in the system. There is the additional risk that the stolen file might contain confidential information, or even macro viruses, in the case of *WinWord* and *Excel* documents, which SirCam will help to spread further.

SirCam begins installation by attempting to copy itself into the Recycle Bin. It is assumed that this is called 'Recycled', and that it is located on the drive that contains *Windows* (the hard-coded directory name is the one thing that prevents SirCam from functioning correctly in *Windows NT/2000/XP*, in which the Recycle Bin is named 'Recycler').

Once SirCam has placed itself in the Recycle Bin, where it is hidden from the view of programs such as *Explorer*, SirCam will copy itself to the System directory, using the name 'SCam32.exe'. A new value, Driver32, is placed in the RunServices key in the registry, which refers to the SCam32.exe file. Thus, the worm will run whenever *Windows* is booted.

Additionally, SirCam.exe installs itself as the application that handles requests to run other .exe files, by changing the exefile Open key (HKCR\exefile\shell\open\command) in the registry. In this way, SirCam gains control whenever an application is run. This is not a new technique. In fact, the PrettyPark worm was one of the first viruses to utilize this technique, more than two years ago.

Not content with such control, SirCam will also watch for requests to run applications in the Desktop directory (referred to by …\Explorer\Shell Folders\Desktop in the registry). When such a request is made, SirCam will prepend itself to the specified file, before running the application! Thus, even if the registry is restored and the files are removed from the Recycle Bin, infected files could remain in the Desktop directory.

Spread the Word

After installation is complete, SirCam will search the local network for computers which allow unrestricted access. SirCam will copy itself to the Recycled directory on each unprotected computer that is found and append a line to the Autoexec.bat file. The line will run the SirCam file from the Recycle Bin whenever the computer is booted. Then SirCam will rename rundll32.exe to run32.exe in the Windows directory on the remote computer, and create another copy of SirCam in its place. Neither the copying of the SirCam files to remote computers nor the emailing to other users occurs in *Windows NT/2000/XP*, however each of the other effects can be observed.

Randamn

The date-activated trigger is checked at this point, however two factors prevent it from working. The least significant of these factors is the dependency on the date format used by the computer, which SirCam requires to be dd/mm/yy (as opposed to mm/dd/yy, for example). However, the more significant factor is that the trigger contains a random component, but the random number generator is never initialized, resulting in there being no chance of producing the required condition.

Unfortunately, there are two other ways in which the payload can be activated. One is by renaming one of the three files, SirC32.exe, SCam32.exe, or rundll32.exe, to another name and running that file. The other is to run an attachment whose stolen file contains the characters 'FA2' not followed immediately by the characters 'sc'. The payload deletes all files in all directories on the drive that contains *Windows*.

The missing randomiser initialization prevents SirCam from copying itself to the Windows directory as ScMx32.exe, and copying itself to the Startup directory (referred to by …\Explorer\Shell Folders\Startup in the registry) as Microsoft Internet Office.exe. It also prevents SirCam from creating, on October 16, a file that fills the remaining disk space.

I'm Sending You a Letter

When SirCam is run for the first time, it will change *Internet Explorer's* Download directory (referred to by HKCU\Software\Microsoft\Internet Explorer\Download Directory in the registry) to point to the Desktop directory, in order to maximize the use of the prepending routine mentioned earlier.

During the second execution, SirCam will gather email addresses into files stored in the System directory. SirCam searches for email addresses in *Internet Explorer's* Cache directory (referred to by HKCU\Software\Microsoft\WindowsCurrentVersion\Explorer\Shell Folders\Cache in the registry), the user's Personal directory (referred to by HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Personal in the registry), and the directory that contains the *Windows* Address Books (referred to by HKCU\Software\Microsoft\WAB\WAB4\Wab File Name in the registry), in files whose name begins with 'sho', 'get', or 'hot', or whose suffix is 'htm' or 'wab'.

Thus, SirCam creates a file called scy1.dll, which contains the addresses from %cache%\sho* files, sch1.dll contains the addresses from %cache%\get* and %cache%\hot* files, sci1.dll contains the addresses from %cache%*.htm files, sct1.dll contains the addresses from %personal%*.htm files, and scw1.dll contains the addresses found in *.wab files.

If the Address Book registry key is not found, SirCam will search for WAB files in the System directory instead. After creating the lists of email addresses, SirCam will search for files to attach to the emails that it will send. The list that is created consists of the name of every .doc, .xls, and .zip file in the user's Personal and Desktop directory and is called scd.dll. An apparent oversight on the part of SirCam's author prevents the inclusion of .exe files in the list.

On the third and subsequent runs, and if an active connection to the Internet exists, SirCam will retrieve the information required to send email using SMTP. Sending mail using SMTP avoids relying on an email program such as *Outlook*. The SMTP information consists of the current user's email address (HKCU\Software\Microsoft\Internet Account Manager\Default Mail Account\Accounts\SMTP Email Address in the registry), the address of the email server (HKCU\Software\Microsoft\Internet Account Manager\Default Mail Account\Accounts\SMTP Server in the registry) and the user's display name (HKCU\Software\Microsoft\Internet Account Manager\Default Mail Account\Accounts\SMTP Display Name in the registry).

If, for some reason, this information does not exist, SirCam will use prodigy.net.mx as the email server, and the user's logon name as the email address and display name. Then SirCam will attempt to connect to an email server. First, it will try the user's own email server (or prodigy.net.mx). If this fails, SirCam will attempt to connect to the email server of the person who sent the infected email. This is possible because SirCam carries within it the email information of the previously infected person. If this connection fails, then SirCam will attempt to connect to goeke.net, then enlace.net, then doubleclick.com.mx.

Compositions

If one of the connections to an email server is successful, an email is constructed in the following way: if the language used on the current user's computer is Spanish, SirCam will send email in Spanish, otherwise it will use English.

The email body consists of three lines. The first line of the email body is always 'Hola como estas?' in Spanish, and 'Hi! How are you?' in English; the third line is always 'Nos vemos pronto, gracias.' in Spanish, and 'See you later. Thanks' in English. The second line is chosen from the following list, in Spanish:

- 'Te mando este archivo para que me des tu punto de vista'
- 'Espero me puedas ayudar con el archivo que te mando'
- 'Espero te guste este archivo que te mando'
- 'Este es el archivo con la informacion que me pediste'

and, in English:

- 'I send you this file in order to have your advice'
- 'I hope you can help me with this file that I send'
- 'I hope you like the file that I sendo *[sic]* you'
- 'This is the file with the information that you ask for'

However, since the randomiser is not initialized, the choice is reduced to the first line alone, until October 16, or until SirCam has been run at least 8000 times, at which point the last line can be chosen, too.

As long as an active connection to the Internet exists, SirCam will send email to every address in each of the email lists that it created. It will send an email three times to each address in the scw1.dll list, then once each to all the other addresses, in the order: scy1.dll, sch1.dll, shi1.dll, and sht1.dll, before starting again with scw1.dll.

SirCam keeps the current mailing position in the registry, so if the connection is broken and restored later, SirCam can continue to send mail as though it were never interrupted.

Interestingly, SirCam ensures that the current user never receives an email from SirCam. In the case that the recipient is the current user, SirCam will send the mail instead to email address otrorollo@esmas.com.

I'm Sending You a File …

For each email it sends, SirCam will randomly select a file from the scd.dll list, prepend itself to that file, attach an additional extension, chosen randomly from 'pif', 'lnk', 'bat', or 'com', and send the email. The lack of the randomiser initialisation has no impact on the emailing routine. If an Internet connection exists for long enough, eventually every recipient will receive multiple copies of every file in the list, and among those copies all four of the random extensions will be represented. To avoid overloading email servers, SirCam remains idle for one minute between sending each email.

In some ways, SirCam's success has had much to do with luck: the emails SirCam constructs are unintentionally malformed such that it appears, to some email scanning products, that the mail contains no attachment. This has allowed the worm to slip past some gateway scanners, though this is far from the sole reason for SirCam's widespread distribution.

Conclusion

Evidently SMTP propagation is the hot topic of the year. Even the first Win32 mass-mailer, Parvo (see *VB*, January 1999) used an SMTP engine. However, most of the worms that have utilized SMTP mailing so far have got a few things wrong. Thanks to the implementation mistakes and bugs, it was a little while before SMTP worms could take their real place. Most of the previous worms have lacked some important detail in their spreading mechanism. For instance, Magistr often sends clean files or files that will not run on the recipients' computers because of some missing DLLs. As VBS creations are controlled with proactive technologies, so virus writers turn their attention to the creation of more dangerous binary worms. One thing is for sure: there is more to come!

W32/SirCam.worm

Aliases:	W32.Sircam.Worm@mm, Win32/SirCam@mm, Backdoor.SirCam.
Type:	Win32 SMTP mass-mailer worm, prepender.
Payload:	Propagates confidential files, attempts to delete all files on disk, attempts to eat up free space on disk.
Removal:	Fix registry and modified files, delete standalone worm copies, restore infected ones from backups.

FEATURE

Apple Blight

Paul Baccas
Sophos Plc, UK

The Apple Macintosh has the dubious honour of having been the target of the first recorded computer virus. Yet you might be forgiven for believing that the Macintosh no longer exists, as both the virus-writing community and the anti-virus vendors have, of late, largely ignored this operating system.

The situation is not helped by an attitude of complacency among Mac users. A typical response from a Mac user on the subject of computer viruses is: 'Viruses do not happen on my computer!' In fact, they *should* be asking, 'How will this virus affect my Mac?' And it is not just the Mac users who seem to be ignorant of the risk of viral attacks; Mac developers seem to suffer from the delusion that viruses are something that only happen to other operating systems.

The past 12 months have seen the vulnerabilities of the Macintosh operating system highlighted by both new and old threats. AplS.Simpsons.A represented a new threat, whose possibility was known to researchers, though its arrival was still somewhat unexpected. The old threat was that of the macro virus, and arrived in the form of W97M.Melissa.W (see *VB*, February 2001, p.3).

The Simpsons in the Big Apple

Why is New York called the Big Apple? Perhaps because it is full of worms.

The AplS.Simpsons.A worm was not only the first piece of malware to use AppleScript, but also the first piece of Mac malware to implement mass-mailing capabilities. Like its cousins on the Wintel platform (VBS/VBSWG etc.), the worm makes use of the automation abilities of mail programs. In this regard it has more in common with *Office* and Script infectors on the Wintel platform than with other native Mac infectors. The use of automation is now a fundamental part of the macro and Script virus area.

When speaking about *Outlook 2001* at the *MacWorld Expo* earlier this year, one *Microsoft* employee is reported to have said, '… in response to my query about how will the Mac version deal with Melissa, and other VBA-created virii, the response from *Microsoft* was, "Easy, we don't support Visual Basic in the product, but we are going to try and have an excellent AppleScript implementation."' (*MacTech*, February 2001.)

The belief that automation is a good thing seems to prevail among software developers, and most Macintosh products are in some way controllable via AppleScript. When I was asked, in January 2001, about the possibility of the occurrence of a Loveletter-type worm on the Macintosh my response was, ‘I am now of the opinion that it would be possible, on a Mac, for a virus/worm to send emails.’

I reached this conclusion simply by viewing the Scriptable Items list that is provided by AppleScript in several mail programs, combined with the number of freely available AppleScripts on the Web for autoresponse to emails. Typing ‘AppleScript’ into the search engine on the Apple Web site revealed a plethora of applications that are scriptable and that a) are mail programs, and/or b) already have virus problems on the Wintel platforms.

Though there have been no confirmed reports of AplS.Simpsons.A worm in the Wild, the ease with which it must have been written, compared with the complexity of the file format, is a worrying feature. There is still debate as to whether researchers actually received samples of two worms or one. The author of AplS.Simpsons.A worm wrote it in such a way that it relied on the presence of several conditions on the OS for it to work. However, I am confident that more could be achieved by this type of malware.

Mac(ro)s

In May 1998, six months after the last major look at the world of Macs and Macros (see *Proceedings of the Seventh VB Conference*, 1997), John Norstad retired *Disinfectant*, the free anti-virus utility for the Mac, saying, ‘I made this decision not because of the new Autostart-9805 worm, but rather because of the widespread and dangerous *Microsoft* macro problem.’ Now, over three years later, with the number of native Mac viruses around 50 and the number of macro viruses in the thousands, Norstad’s decision appears to be justified.

This was in the days of *Word 6.x* and *Excel 7.x* for the Mac and when *Office 97* was becoming standard issue on *Windows*. At the time, the macro virus threat was beginning to top the prevalence tables and was a relatively straightforward problem. Since then, the *Office* products have become more complex, the macro languages more powerful and macro capability has been added into more applications.

Office Renovations

In the applications arena *Office* has undergone some major renovations over the past three years and on the horizon there are more to come. The first was *Office 98* for the Macintosh, which was introduced as a replacement for older versions of *Office* on the Mac. *Office 98* is essentially *Office 97* with a few minor differences. Then came *Office 2000*, a new version of *Office* for WinTel machines with VBA5 replaced by VBA6 and some more comprehensive security features. Next, *Office 2001 for Mac* was released. This is essentially *Office 2000 for Mac* with a few changes as would be expected in a product with a slightly longer development time. Finally, the most recent offerings in the *Office* family are *Office XP* and the Beta of *Office for OS X*.

Virus Developments

In the virus arena there have been less radical, but equally significant changes.

The upgrade in the macro language in *PowerPoint* to VBA5 meant it became possible to write macro viruses for *PowerPoint*. The realisation by virus writers of the COM nature of the *Office* application and VBA has provided a significant threat, from cross-application viruses to mass-mailers. Along with this the emergence of multi-partite *Office* infectors has worrying implications. Combined with the rise of anti-heuristic techniques and encryption, this has meant more complex and convoluted code.

Working with Different Offices

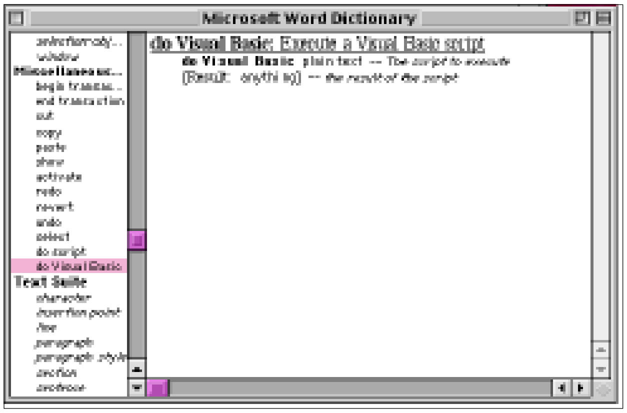
The main difference between the versions of *Office* is the operating systems on which they run. One of the biggest differences is that, on Intel-based processors they use the Little Endian format, while on the others the norm is Big Endian.

There is also the problem of the 4k sector size introduced into the OLE standard in Win2k (see *VB*, October 1999 p.8). Though the OLE2 definitions are a standard, this does not preclude the occasional unforeseen changes in *Word 2001 for Mac*, where a compiler change changed a DWORD into a WORD, giving W97M/Melissa.W. These differences mean that OLE2 handling code must be both robust and accurate.

Another difference concerns how the macro code is stored. Macros written in VBA5/6 are stored in several places in the OLE2 files – there is the compiled code (aka p-code), the execode and the compressed code (though the execode does not always exist). The execode and compiled code are specific to the application from which they are made, for example, a corrupted file with just p-code or execode that replicates under *Office 97* will not work under *Office 98*. When an *Office 97* file is opened under *Office 98* the p-code and execode are recompiled to their *Office 98* forms.

This means that X97M/Jini.A1, which only has *Office 97* execode, does not work under any other platform. Also it means that W97M/Marker.GO and other variations on its theme, which have a corrupt compressed code part and working *Office 97* p-code, will not work under any other platform either. However, the fact that they do not work on other platforms does not preclude those other platforms from acting as conduits for infection. Neither does it excuse the fact that not one of the descriptions of these viruses mentions these facts.

Though differences in code make it obvious to the analyst what should work on various applications/operating systems, descriptions rarely give this information, leaving the user in the dark. Worse than that, the user may think that something is being hidden from them and attempt to investigate for themselves. This kind of tinkering quite often leads to new variants.



do Visual Basic: Execute a Visual Basic script
do Visual Basic plain text -- The script to execute
[Result: anything] -- the result of the script

Figure 1: The scriptable items list provided by AppleScript.

The seemingly extreme apathy from Mac users, combined with the general apathy for virus/security news from computer users as a whole, makes it very difficult for AV vendors to get their message across. (The message, of course, being to practise secure computing.)

Cause for Concern

If you look at the Scriptable Items list (see figure 1) of any *Office* application you will find ample cause for concern. Virus writers were relatively quick to see the advantages of multi-application *Office* infectors and multi-partite *Office* infectors, as well as using *Office* documents as droppers of other malware on Wintel. The potential exists to do all these things on the Mac, with the added advantage for the virus writers of being in a format that would not normally be understood, even by the products produced by those AV companies that cater for the Mac.

It is not all doom and gloom, however; *Office 2001* was the first in the product family that allowed VBA *not* to be installed (although there is one macro virus on the WildList that this will not prevent from replicating).

VBA is not merely a macro language – it is far too powerful for that. When VBS is combined with the power of AppleScript, the consequences will probably be disastrous.

Conclusions

For once, user education is not the only issue at stake. Even if your product developer does not have a Mac product, their mail gateway product should at least have the ability to detect Mac malware and their server products should be able to detect viruses on Macintosh shares. W97M.Melissa.W caught out too many developers – in fact only one product could detect the virus in the 4K sector size OLE 2 document. And can your mail product detect AplS.Simpsons.A?

The AV community must have the foresight to do the research and be prepared with a fix before these problems occur. I believe that, even if it does not happen next week, we will certainly see more malware utilising the Mac in the future.

OPINION 1

Pragmatic Anti-Virus Testing

Joe Wells
AVTestLab.com, USA

If our industry is to follow the pattern of the rest of the computing world, AV companies will soon become application service providers and, before long, the service capabilities of AV vendors will become the most important aspect of the industry. It is important that anti-virus test methodologies are revised to reflect these changes in the use of AV products.

Mass-mailing threats have demonstrated the need to refocus testing. Once these threats appeared, anti-virus ceased to be simply ‘a product’ and customer service became a critical consideration. The AV company’s ability to respond rapidly to a time-critical threat has come to the fore.

Pragmatic testing addresses the more important issues relating to AV products, especially where time and financial constraints play a part. While I believe such testing should include the traditional issues, it should, in addition, be expanded to include the testing of anti-virus services.

Good Testing Requires Good Input

Companies and testing organizations can access two key resources for input on what needs to be tested.

We can talk to technical managers in large corporations who deal with AV problems every day. (Now, there’s a novel idea: ask users what they want to see tested.) This means it’s time to admit that these people know their jobs and know what they need. In the past, some AV ‘experts’ have interpreted user requests as ‘wants’ as opposed to ‘needs’. (‘We know better than the users. We’ll give them what they really need.’) This ideology is wrong. We do not know the users’ situation and environment better than they do. When they say they need something, they genuinely do need it. We must listen to them – recognizing them as the professionals they are. Taking their requests and suggestions into consideration will help us fulfil their needs.

There are resources available to us within our own industry. A testing organization can ask an AV company how their product should be tested: QA staff should be asked what they test and how they do it, and technical support staff should be asked what ‘really’ needs to be tested in a product, based on their experience of the problems they have encountered.

Good Testing Requires Good Testers

What makes for a qualified tester? As in any other field, testing requires knowledge, experience and meticulous methods. In addition, there are issues unique to the testing of anti-virus products and services that require a more specialized knowledge.

For example, one critical and often overlooked issue is the tendency to immediately suspect the AV product when a virus sample is missed. Given the historical quality of viruses and anti-virus products, it is preferable that the tester should suspect the virus sample immediately, rather than the product. It is far more likely that it is the sample which is bad. This leads us to some of the unique aspects of tester qualifications.

A qualified anti-virus tester must have, at the very least, a working knowledge of the following: how viruses work, how anti-virus products work, how to replicate a virus successfully, how to disassemble and analyse a virus, how to verify a virus sample is valid (not corrupted, hacked, and so on), how to verify a virus sample's identity and how to verify that a virus has been 'cleaned' correctly.

These items, in turn, require some knowledge of multiple programming languages (from x86 assembler to VBScript), multiple environments (DOS, Win32, OLE2, *MS Office* Applications, etc.) and multiple file formats (PE Executable, OLE2 stream, master boot code, and so on).

In addition, there are issues regarding anti-virus product testing environments. A qualified test lab must have dedicated test machines to test all types of viruses effectively in these different environments. Such systems must be continually infected and returned to a trusted state. Moreover, it is highly desirable that the overall processes be automated. In turn, the tester must have a good knowledge of these environments.

Good Testing Requires Good Focus

An anti-virus testing organization should be dedicated to, and highly qualified in, testing anti-virus-specific features. The following list of items should be tested:

- On-demand scanning.
- On-access scanning.
- Email scanning.
- Appropriate repair.
- System recovery.
- Unknown virus detection.
- False positives.
- Product scan speed.
- Update effectiveness.

At the same time, the following service aspects should be tested for accuracy, availability and response time: manual and in-product information, Web site information, newsletter information, email technical support, phone-based technical support and fax-based support.

Of course, in dealing with support services (including a manual, online help, a Web site, etc.), certain aspects cannot be tested objectively. Ease of use, friendliness of interface, and even the clarity of wording, are all subjective, and tests of such features will reflect the personal preferences of the tester.

However, other aspects of support services can be tested in a scientific and repeatable manner. The various sources of information associated with a product (or specific update) can be checked for accuracy and consistency. If 'accuracy' is extended to exclude misleading and exaggerated information, then marketing claims can be tested as well. Response time testing should be applied to email, telephone, and fax support, and included in the test results.

Good Testing Requires Fixing Current Problems

Aside from additional service testing, future testing must deal with two problematic aspects of current testing: fair certification and heuristic testing.

Seldom do anti-virus product developers release ineffectual or flawed updates. Most have a mature update process. Yet, this stability is often hidden in simple pass-or-fail test reports. For example, if a developer regularly provides solid updates, they pass the test. A sloppy competitor may have to provide several updates before passing the test, yet the subsequent report states simply that both passed. Therefore, the information about the developer's process stability and trustworthiness is hidden. If a developer has a stable, trustworthy process, users should be made aware of this. If, on the other hand, a developer regularly releases ineffective updates, this fact should be made public too.

To solve this problem a testing entity should implement a rating system, which not only validates a developer's update, but also demonstrates the stability of their update process. By extension, this rating can be applied to the detection, repair, and recovery testing above.

For the rest of this section I will refer to this as a 'Vn' rating system.

A Vn rating has two aspects. The 'V' is simply the fact that the product was validated (i.e. certified) because it successfully handled all the threat samples. The 'n' portion is a number representing the update/test iteration that earned the product's 'V' rating. Thus, if the first update sent by the product vendor was successful, it receives a V1 rating, but if it took five attempts, the rating would be V5.

Such a system would provide users with information that is currently generally unavailable. In addition, it would allow anti-virus product developers to demonstrate the stability and trustworthiness of their updating process.

The process of validating each product may be repetitive. For example, if the first update sent by the vendor fails, the tester immediately alerts the vendor, who fixes the problem, and sends a new update for validation. Although an update is ultimately validated and distributed, this cycle of failure would be reflected in the Vn rating.

The purpose of this approach is to avoid hiding pertinent information from the user – namely, the integrity of the vendor's updating process and related possibility of ineffective updates being distributed.

An Approach to Heuristic Testing

Current testing needs to address a product's ability to deal with new and unknown threats. Products may detect such threats by various means: as variants by fuzzy detection or generically by heuristics, pattern recognition, behaviour, or automated integrity response.

One solution is safe, simple and does not involve using 'virus simulators' or creating new viruses or variants. Rather it uses the tools at hand. All that is needed are a few recent viruses and a current anti-virus product with an older update. For example, have a current version of VerminBlaster on hand, along with a signature database from May, and test it against new samples from the August WildList. This uses real viruses that are a real threat, which the product should not recognize.

This test represents a real-world situation. There really are customers out there who have products with older updates. They really are potential victims of newer viruses. However, even this approach will be problematic because there are several possible test results for each new sample. For this reason results of testing will have to be reported in more detail. Consider the potential results for any given sample:

- Not detected.
- Detected generically as a threat.
- Detected generically as a possible threat.
- Detected generically as suspicious.
- Detected and identified correctly.
- Detected and identified as being a variant of a known threat.
- Detected and identified as being a completely different threat.

The Future

The AV industry must move ahead, and testing will have to change to keep pace. As the world moves to the *Microsoft .NET* model and AV companies become application service providers, new issues will arise, which will require new testing models. This is clearly a challenge for testing organizations. Especially when one considers the fact that anti-virus products and services have evolved dramatically over the past decade, yet many testing organizations are still stuck back in the scan-age of the early 1990s.

OPINION 2

Sysadmins are Doing it for Themselves

David Harley
NHS Information Authority, UK

Who are the readers of *Virus Bulletin* and the other (less specialized) security magazines? People in the anti-virus industry, of course, though a certain group of technical specialists is probably better represented than some other groups (especially Sales in my somewhat depressing experience). Other readers include independent specialists, system administrators, network gurus, and some managers with a technical bias and/or accounts big enough to sway the industry's perception of 'What The Customer Wants'.

In the security industry, conference papers are usually presented by and for the industry itself, or, more rarely, its largest customers. Corporate briefings, seminars and training workshops, tend to be marketing exercises – which doesn't mean they can't be very useful, but they do usually represent a particular group of commercial interests, and give the impression of being intended for corporate customers (potential or actual), rather than individuals, small businesses, community projects and so on.

Vendor Web sites (and, indeed, their product ranges and licensing terms) tend to be polarized between large corporations and home users. As large concerns accept the need to allocate increased resources to security management and even anti-malware specialists, the anti-virus industry is waking up to the need to consider their requirements and complementary expertise in virus management.

Maintaining Standards

A major corporate body is likely to have in-house expertise, protocols, policies, and premium support agreements to fall back on. Furthermore, the voice of the (major corporate) customer is heard where AV vendors are seldom found.

BS7799, aka ISO17799, has its origins in a statement of 'best practice' as determined by major corporates, rather than as prescribed by the security establishment. The detailed protocols favoured by large institutions are likely to be based on those prescribed by special interest consumer and pressure groups such as the Information Security Forum. This reduces the likelihood of favoured vendors gaining undue influence, but may mean that initiatives are not always technically as well founded as they might be. For example, although it addresses the question of virus controls, BS7799 requires some interpretation to apply in the light of current real-life anti-virus technology. Nevertheless, God and consultancies look after the big battalions, on the whole.

AVIEN a Good Time

As Robert Vibert's article in the August 2001 issue of *Virus Bulletin* seems to suggest, administrators have grown tired of relying on the AV industry to supply them with comprehensive information, in an attempt to keep them informed and protected in a timely manner. After all, vendors don't usually get the latest fast burner before their customers. If those customers are security-literate and able to communicate directly (as they can through AVIEN's Early Warning System), a much faster spread of information is possible than that through the AV vendor channels. In fact, the vendors are somewhat handicapped in this respect. We expect a high standard of response to a new threat from a vendor (which doesn't mean we always get it).

By the time a new potential threat has been received, verified and processed by a vendor, passed on to other vendors, and the relevant information passed on via Web sites, mailing lists and the media, a considerable length of time may have passed. And, as we've learned many times in recent years, even an hour can be a long time in the age of the fast burner.

Kindness to vendors isn't what I do best, but I will concede that when information does eventually emerge from vendor sources, it's generally of a pretty high standard. But, in some cases, it's too late for organizations (let alone individuals) who are relying on their vendor of choice to keep them informed, and are not able, for whatever reason, to apply stringent generic blocking of mail attachments (for instance). When AVIEN members are able to block a new email virus/worm while the vendor labs are still glued to their microscope eye-pieces, it is not because they already know all there is to know about the newcomer; it's because they have just enough information to take short-term measures, such as putting in a temporary filter.

Information gathering and risk assessment are ongoing processes, and as time progresses and more information becomes available there are often changes to the descriptions of the threat. In fact, it's a trade-off between being 'timely, but not necessarily correct in every detail' and 'obsessively accurate'.

From the alerting point of view, the value of the information service lies in the opportunity to take the earliest possible action. As a result of the cooperative, non-commercial culture of AVIEN, people tend to be tolerant of an occasional false alarm. If, on the other hand, a vendor were to go off half-cocked, the company might lose both face and competitive advantage. Having said this, false alarms have not been a big issue in AVIEN to date, not least since membership of the network presupposes a level of professional responsibility, and there are enough grizzled veterans to keep the less experienced on track.

SOHOs and Other Red Light Districts

Home users, SOHOs, and small organizations with a single LAN or a handful of subnets face many of the same threats, choices and responsibilities as the big companies, but usually don't have access to the same specialist resources and expertise. Smaller concerns are likely to find themselves having to rely on poorly informed media resources; inconsistent, unreliable AV vendor Web sites; volunteer newsletters, newsgroups and other Web sites, some of which are remarkable only for their grimly amateur, ultracrepidarian status.

This apparent disdain for the amateur may seem, at first, an odd position for someone who wormed (as it were) his way into the AV establishment by way of an assortment of Internet/Usenet FAQs. Sadly, over recent years I have become all too aware of the difficulties of reconciling the practicalities of maintaining unsponsored, volunteer resources with the need to maintain standards. To accept responsibility for an FAQ is, effectively, to say 'This is how it is'. However, 'how it is' quickly becomes 'how-it-was-but-ain't-anymore'. The somewhat bureaucratic nature of Internet FAQ administration militates against quick modifications. Leaving aside the esoteric complexities of Usenet administration, anyone can set up shop as an 'anti-virus expert' (no qualifications needed), and may end up giving disastrous advice even with the best of intentions. Of course, there are a number of useful amateur sites, just as there are a number of useless vendor sites.

ASPs and Outsourcery

Outsourcing scales badly to small organizations: unless the services provided are remarkably perfunctory, it can be cheaper to employ a full-time security person or, as is a more likely situation, get someone to squeeze a quart of security into a part-time pint pot. For home users, outsourcing cannot possibly scale at all according to the cost models used by the current major players.

There's a certain irony here. The perfect transparency promised by the idea of farming out your virus/anti-virus problems to a third party is, on the whole, *exactly* what the everyday home user wants. Hence, perhaps, the willingness to rely on anti-virus packages bundled with home systems (which may or may not be installed, updated or upgraded), Web-based scanners, evaluation copies or one of the dwindling band of free scanners. I can't help but wonder whether the home user market might be a more appropriate target for ASP evangelism than the major corporates. After all, these organizations seem to be becoming better and better at incorporating anti-virus with other security management. If ASP solutions work for telecommuters, it seems likely that they'd work for other home users.

Who will Protect the Munchkins?

Home users fare even worse than the small business: apart from being prey to the same unreliable resources as everyone else, the average home user's main point of contact with the security world may well be a salesman at the local electrical store. Or something/someone they stumble across on a *Google* search.

One of the side-effects of having my mail address listed on various Web sites is that I receive more requests for help than I can handle (along with multiple copies of Hybris, Magistr, and SirCam). Leaving aside the occasional attempt by a small business to blag some free consultancy from me, there seem to be a lot of confused people out there who can't find or make sense of the FAQs, haven't quite made the jump from real human being to Internaut, and who simply don't understand the technology that drives their desktop. (And why should they? I don't understand the technology that carries me to work, but that doesn't disqualitfy me from commuting.)

This is the group targeted by vendors offering the 'Swiss Army knife' type of software suite, combining personal firewalls, cut-down anti-virus software, intrusion detection systems, mail cryptography and so on. These can be cheap in terms of unit cost, but may not be cost-effective in maintenance terms, or even particularly efficient in security terms. This raises the question as to what we expect desktop software to be able to do in the absence of a battery of corporate solutions.

Somewhat unexpectedly, the CERT Coordination Center at Carnegie Mellon – usually thought of as a corporate resource – addressed this question recently in the context of home network security in response to the increasing availability of broadband connectivity. The document (see http://www.cert.org/tech_tips/home_networks.html) refers to several attacks, including viruses and Trojans, backdoors, DoS and DDoS attacks, mobile code, and packet sniffing.

Conclusion

The security establishment *should* worry about the home user. Even non-viral attacks on an individual machine may be a precursor to attacks on other machines, using the first machine as an intermediary. As types of threat converge and use multiple entry points, it is unrealistic to think of various classes of computer user as being somehow totally isolated from one another: we can only deal adequately with current malware problems by being responsible netizens.

However, the AV industry does not appear to be very interested in supporting home users. And understandably so: it's an expensive, time-intensive exercise to take support calls from someone who holds only a single licence. Received wisdom within the industry is that the average home user won't pay more than a bare minimum for AV services in any form. Which may explain why free resources are used so heavily – but will we ever be able to control the malware problem while home users remain so vulnerable, making them a potential staging post for attacks on corporate systems?

COMPARATIVE REVIEW

Surfing the NetWare

Matt Ham

It is exactly a year since the last *NetWare* Comparative and little has changed. On that occasion I bemoaned the fact that *NetWare* required a 240 MB patch in order to meet *Novell*'s minimum patch list. This time the patch size has increased to 280 MB and must be approaching the size of the operating system itself. The line-up of products has not changed much since a year ago either. There were eleven products on offer in the previous *NetWare* Comparative, all of which are represented again here, along with the additions of *GeCAD*'s *RAV*, which was a beta product last year, and *Trend Micro*'s *Server Protect for NetWare*.

Issues that arose last time fell into two main categories. The first was the age-old favourite of ACG.A and ACG.B in the polymorphic sets, both viruses having caused problems to a wide variety of products over the years. These have, however, become less of a problem with more recent incarnations of software on other platforms and the question is whether this improvement will transfer across to the *NetWare* products on test.

Second was the ever-present bogeyman of extension list problems, one which centred on the lack of scanning where extensionless files were concerned. Since O97M/Tristate is represented in the WildList still, this could prove to be a problem if developers have been tardy. Since the last Comparative there has been yet another new entry as far as extensions are concerned, the .LNK extension which is used by W32/SirCam.A as a method of pretending to be an unadulterated version of the infected and emailed file. This might be expected to prove a pitfall for at least one of the products on offer, if past experience is anything to go by. Past experience also predicts that the victim could be any of those scanners not scanning all files by default, though to discover if this was a problem you will have to read on.

Testing Procedures and Test Sets

By way of a little variation from the previous *NetWare* Comparative, the client platform this time was *Windows 98* with *Novell*'s *Client for Windows 95/98/ME v3.30.00.0 SP 3* running on W98. *NetWare* itself was version 5.1 patched to Service Pack 3. This patch level adheres to the *Novell* minimum patch list for the week of product submission. Products were submitted no later than 6 August, and the July WildList (the most recent available at that time) was used as a basis for the construction of test sets for the In the Wild (ItW) set.

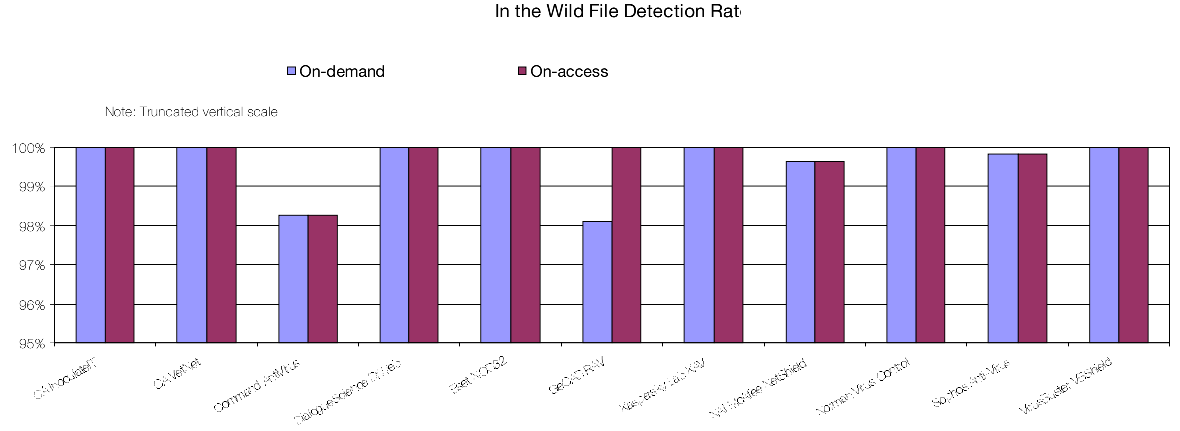
Scanning was performed on the server with the virus test sets and speed test sets both being located on the SYS volume. While this avoids scanning speeds being dictated by the network speeds under the test conditions used, it may give higher throughput rates than might be encountered when scanning files across a network.

For the on-access scanning test the files in the viral test set were opened from a client machine in order to trigger detections. Due to the nature of server scanning on *NetWare* the checking of boot sector viruses was not performed.

There were a few additions to the ItW test set. The most notable newcomer was the aforementioned W32/SirCam.A with its wide selection of double extensions and the usual addition of 32-bit *Windows* infectors, script worms and macro viruses making up the unexceptional remainder.

Symantec

The offering from *Symantec* showed early promise but was soon discovered to be virtually untestable in the defined test environment. The NLM-based portion of the product can readily be installed and updated, though the latter involves some shenanigans, from a *Windows 98* Client. At this point there is a *Norton AntiVirus* available on the server, but how is it controlled? The answer is in the use of *Symantec*'s



On-demand tests	ItW File		Macro		Polymorphic		Standard	
	Number Missed	%	Number Missed	%	Number Missed	%	Number Missed	%
CA InoculateIT	0	100.00%	0	100.00%	9	98.90%	0	100.00%
CA VetNet	0	100.00%	16	99.71%	1	99.99%	0	100.00%
Command AntiVirus	7	98.27%	0	100.00%	1	99.99%	6	99.42%
DialogueScience DrWeb	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	1	99.98%
GeCAD RAV	8	98.10%	0	100.00%	0	100.00%	17	99.13%
Kaspersky Lab KAV	0	100.00%	0	100.00%	0	100.00%	0	100.00%
NAI McAfee NetShield	6	99.65%	3	99.97%	1	99.88%	8	99.77%
Norman Virus Control	0	100.00%	0	100.00%	17	97.92%	14	99.58%
Sophos Anti-Virus	3	99.82%	13	99.67%	191	95.36%	37	99.15%
VirusBuster VBShield	0	100.00%	39	98.97%	28	95.71%	17	99.37%

proprietary management interface, which works solely via *NT*. With a *Windows 98* Client no control could be exerted upon the server software and testing was abandoned.

Trend Micro

Trend's offering too declared itself to require *NT* as an administration platform, though the claim here was that after installation from an *NT* box, the server side software could be controlled through *Windows 98*. Several hours later, having set up a number of information servers in order to deploy the *Trend* product, the situation was much the same – a loaded, but singularly uncontrollable NLM on the server. Again, testing was abandoned.

Computer Associates InoculateIT v 4.5 engine 26.04

ItW File	100.00%	Macro	100.00%
ItW File (o/a)	100.00%	Macro (o/a)	100.00%
Standard	100.00%	Polymorphic	98.90%

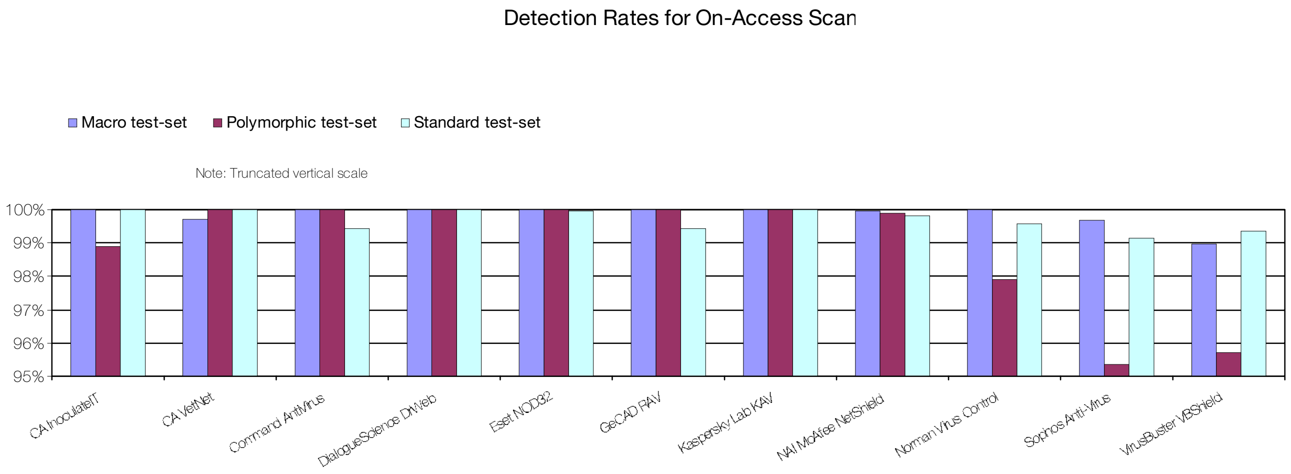
Despite having the same version number as in the last Comparative, *InoculateIT* has seen changes in many parts of its operation. Installation uses the same CD as the last test and, as is customary with *InoculateIT*, there was a patch to be added before operation could begin. These processes performed with admirable ease, though updating virus signatures was more basic and labour-intensive.

Amusement was to be had when loading and unloading the software with the use of the surely made-up word 'endingizing' during one particular session. Notification of infection messages with hex descriptors commencing 0baddeed was also sufficient to enliven the testing procedure a little. If faults were to be found, these would be in the fact that the test procedure took an inordinately long time – delays were noted during the clean set at every point where the directory to be scanned was altered.

On the detection results front, however, *Computer Associates* will be pleased again with only nine samples of the polymorphic W95/SK.8044 being missed out of the full test set. As expected from past performances there were no false positives in the speed testing, so *InoculateIT* is once more possessor of a VB 100% award.

Computer Associates Vet NetWare Anti-Virus 10.3.4

ItW File	100.00%	Macro	99.71%
ItW File (o/a)	100.00%	Macro (o/a)	99.71%
Standard	100.00%	Polymorphic	99.99%



In the last review, the *VetNet* program was considered one of the simplest for installation. There was a little confusion as to the name of the program; although referred to almost exclusively as *VetNet*, the actual NLM goes by the name of *Vet_Net* – something which, thankfully, was explained in the HTML installation file provided. After this the program proved easy enough to configure, and the scans were rapid enough that any installation delay could be easily forgiven.

Configuration changes are implemented at the console rather than at an external point such as the workstation, and therefore did not incur a delay in registering, making this a pleasant affair as far as direct hands-on use is concerned (though, perhaps, less desirable to a remote administrator).

As far as detection was concerned, *Vet* missed identical files on access and on demand, one of which was a single specimen of the polymorphic virus ACG.A. The remainder of misses lay in the macro set, where the majority of misses were samples of the polymorphic X97M/Soldier.A. The misses were very similar, in fact, to those in the DOS Comparative two months ago, and were a vast improvement over the detection rate noted in the September 2000 *NetWare* comparative. A good improvement since last year and retaining full detection of In the Wild files gains *Vet NetWare* a further VB 100% award.

Command AntiVirus for NetWare v 4.61I

ItW File	98.27%	Macro	100.00%
ItW File (o/a)	98.27%	Macro (o/a)	100.00%
Standard	99.42%	Polymorphic	99.99%

Unusually, *Command* is the sole representative of the *F-Prot* stable represented in this review and the absence of its usual pair of running mates seemed to have put it off its stride. The main problem lay in detection, which was far from *F-prot*'s usual outstanding performance on other platforms and had reverted to the singularly inept manner in which it performed in the last *NetWare* Comparative. On that occasion a VB 100% award was missed by the absence of extensionless files on the list of those to be scanned. Rather than learn from this experience, *Command* now fails to scan .HTM, .PIF and .LNK extensions by default. This combination saw many samples of JS/Kak missed, a scattering of ignored VBS viruses with HTM portions and a failure to detect some of those files infected by W32/SirCam.A.

On the administration front, *Command* scored some negative marks by having a far too vigorous scheduled scan which seemed difficult to be rid of, and which interfered with several test procedures. Since scans are still somewhat tricky to spot as being in progress this was not noted at the time of scanning. The scanning speeds were also very much on the slow side and *Command* will, no doubt, be somewhat disappointed with their overall performance.

The cynical, and highly controversial hypothesis that the lack of any other *F-Prot* products in this test might be due to the other developers being well aware of its failings and preferring to be kept out of the public eye is, of course, totally unsubstantiated since neither product was inspected in any way.

DialogueScience DrWeb for NetWare v 4.25

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

The *DrWeb for NetWare* installation process is one of the simpler of those on offer. Simply unzipping the files that make up the product into an appropriate directory enables activation – though setting up a path to that directory in addition will make running the program simpler in the long run. Scanning was the simplest and speediest in completion of any of the products examined up to this point. Although others may find snap-ins and the availability of an aesthetic interface important, to a jaded reviewer's eyes speed and

On-access tests	ItW File		Macro		Polymorphic		Standard	
	Number Missed	%	Number Missed	%	Number Missed	%	Number Missed	%
CA InoculateIT	0	100.00%	0	100.00%	9	98.90%	0	100.00%
CA VetNet	0	100.00%	16	99.71%	1	99.99%	0	100.00%
Command AntiVirus	7	98.27%	0	100.00%	1	99.99%	6	99.42%
DialogueScience DrWeb	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	1	99.98%
GeCAD RAV	0	100.00%	0	100.00%	0	100.00%	13	99.42%
Kaspersky Lab KAV	0	100.00%	0	100.00%	0	100.00%	0	100.00%
NAI McAfee NetShield	6	99.65%	3	99.97%	1	99.88%	4	99.84%
Norman Virus Control	0	100.00%	0	100.00%	17	97.92%	14	99.58%
Sophos Anti-Virus	3	99.82%	13	99.67%	191	95.36%	37	99.15%
VirusBuster VBShield	0	100.00%	39	98.97%	28	95.71%	17	99.37%

simplicity bring greater pleasure. There will, of course, be a need for greater administrative ability in a large organisation (though with *NetWare* products this may well be less of an issue than for non-server operating systems, since the smaller number of such machines makes home-made scripts much more of a feasible deployment method).

The suspicious file problem, consistently the only fly in *DialogueScience*'s ointment, is still present but not alarming. As for detection, this was once again at the 100% level in all test sets and as such can not be faulted. *DrWeb* rightly earns a further VB 100% award to add to its collection.

Eset NOD32 v 1.99

ItW File	100.00%	Macro	100.00%
ItW File (o/a)	100.00%	Macro (o/a)	100.00%
Standard	99.98%	Polymorphic	100.00%

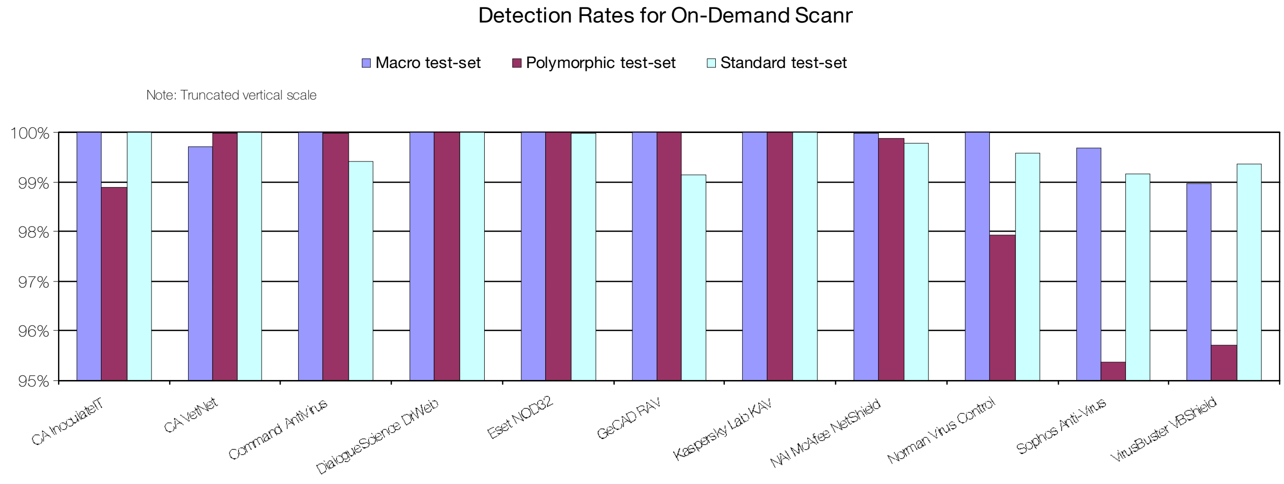
NOD32 has similarities to *DrWeb*, and not only in the length of its product name. The product is another which has a more basic than average interface – in this case consisting of a command line-invoked scanner for both on-access and on-demand duties. These share the same virus database information and lack any form of aesthetic adornment. However, these functional lines conceal what is by far the fastest of the scanning engines in this review. The fastest of the other products was more than twice as slow as *NOD32* over the clean set of executables, while the slowest was over 40 times as tardy.

This was remarkably similar to *NOD32's* performance in the review a year ago, as was the number of files missed. On that occasion one file was missed in the standard set, and on this occasion the standard set saw another solitary miss – though, admittedly, on a different virus. No false positives were registered and thus *Eset*'s product is the happy recipient of a VB 100% award.

GeCAD RAV Antivirus v 8 1.00

ItW File	98.10%	Macro	100.00%
ItW File (o/a)	100.00%	Macro (o/a)	100.00%
Standard	99.13%	Polymorphic	100.00%

Differences between on-access and on-demand scanning were rare in this review, though, unlike the last *NetWare* review, not non-existent. The greatest extent to which this was seen was in *GeCAD*'s product. A very good performance in on-access scanning was somewhat let down by several misses on demand In the Wild. These were all found in VBS worms, both in the .VBS and .HTM parts of these samples.



Unfortunately the log file produced by *RAV* was unusable for parsing attempts and thus detection on demand was completed by deletion. On-access scanning, on the other hand, was performed by denial of access – though it seems unlikely that this might be the cause of such a difference in performance. A difference in behaviour between on-access and on-demand scanning is perhaps not that surprising however, since this is another product which has two applications, one for on-demand and another for on-access scanning. These both operate as console-style GUIs on the server and clearly this has led to slightly differing configurations between the two.

In the last *NetWare* test this version of *RAV* was only just out of beta and failed to install, so these results are a pleasant surprise in comparison. Since the In the Wild on-demand misses are clearly reparable by dint of being absent on access, the future looks promising for the product. The only possible problem lies in the speed of scanning, which was somewhat tardy on the clean executable set, though this is balanced by much superior speed on the OLE set.

Kaspersky Anti-Virus for Novell NetWare 3.06.04

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

Kaspersky Anti-Virus performed well and was easy to install in the last *NetWare* review, making its behaviour in this review all the more mystifying. The NWAdmin portion of the program was able to load the NLM onto the target server, but failed to realise that it had done so. Many hours of parameter and protocol adjustment succeeded in tracking down the problem – AVP being the only product that requires TCP/IP communication to be successful, and being very fussy about the port it performs this over. This problem overcome, the product was one of the simpler to operate, with a large degree of control available in a rather simpler manner than with most other products. The scans proceeded speedily both on access and on demand, though results were at first somewhat confusing. There was a clean sweep on all files, but installing an optional upgrade removed detection of the W32/SirCam samples in the WildList. Thankfully for *Kaspersky Lab*, this was mentioned nowhere in the documentation and was thus not considered to be a default option.

So, despite these various odd features thrown by fate into the path of testing, *Kaspersky Anti-Virus* earns a VB 100% award. As for speed testing the product falls in the middle of the pack – though faster than average for a product controlled from the client rather than the server.

Network Associates McAfee NetShield v 4.50

ItW File	99.65%	Macro	99.97%
ItW File (o/a)	99.65%	Macro (o/a)	99.97%
Standard	99.77%	Polymorphic	99.88%

The installation of *NetShield* proved one of the more taxing, in that it seemed to crash without respite whenever installation was selected. Thankfully it turned out that the installer was simply excruciatingly slow, to an extent not seen with any other product.

The *McAfee* interface on the *NetWare* machine is among the most cluttered of all those on test – combining results for both on-demand and on-access scanning on one standard-sized page. This does not particularly hinder control, but does leave the user somewhat cross-eyed. This is mitigated to a certain extent by the presence of the client-based program, which allows for control over the scanning operations.

On the other hand, this client-based program is apparently in constant contact with the server, resulting in slow scanning speeds if viruses are detected. Admittedly the information seems to be bundled up – since information about infections is incremented in steps rather than on a file-by-file basis, on both server and client.

Hard Disk Scan Rate	Executables			OLE Files		
	Time (s)	Throughput (kB/s)	FPs [susp]	Time(s)	Throughput (kB/s)	FPs [susp]
CA InoculateIT	1181	463109.4	0	69	1149764.7	0
CA VetNet	275	1988844.3	0	25	3173350.7	0
Command AntiVirus	1725	317062.1	0	103	770230.7	0
DialogueScience DrWeb	354	1545006.1	[16]	26	3051298.7	0
Eset NOD32	102	5362080.1	0	14	5666697.6	0
GeCAD RAV	1841	297084.3	1 [1]	18	4407431.5	0
Kaspersky Lab KAV	509	1074522.9	0	40	1983344.2	0
NAI McAfee NetShield	922	593201.9	0	48	1652786.8	0
Norman Virus Control	4414	123908.5	0	20	3966688.4	0
Sophos Anti-Virus	325	1682868.2	0	37	2144155.9	0
VirusBuster VBShield	707	773595.7	0	92	862323.6	0

With regard to detections, however, *NetWork Associates* have once again managed to be caught out by the pesky problem of scanned extensions. The fact that relatively new entries .PIF and .LNK files went unscanned came as no great surprise, but a weary sigh is all that can be mustered upon noting that extensionless files were not subjected to examination. Since the files used in this test were those most recently downloaded from the *NAI* Web site, not even the excuse of the use of old media can be claimed in defence of the guilty parties.

NetShield is of note as a rather obvious sign of the lack of change in some of the programs evaluated here. All the notable problems seen in this review were similarly noted in the previous review – a year may be a vast aeon in politics, but in *NetWare* anti-virus it can sometimes seem like a fleeting second.

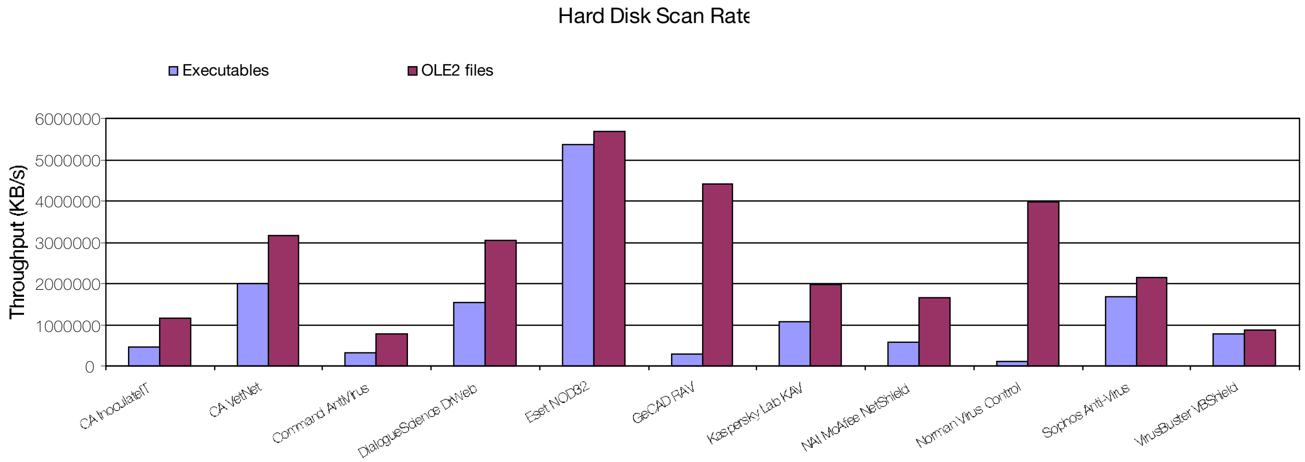
Norman Virus Control v 4.05

ItW File	100.00%	Macro	100.00%
ItW File (o/a)	100.00%	Macro (o/a)	100.00%
Standard	99.58%	Polymorphic	97.92%

Norman Virus Control suffers from some identity problems, being referred to alternately as *FireBreak* and *NVC for NetWare* in the documentation and installation programs. Commencing with installation the user is directed to NWAdmn32 which is now the place in which configuration is performed. This has the odd side effect of making it impossible to alter settings for *Norman Virus Control* from the program itself – all such commands must be issued from this adminstration program. Part of the installation process had to be performed manually from NWAdmn32 but overall the process was not too complex.

As far as the false positives test was concerned, there was one major glitch in that the scan process froze repeatedly on several of the files in the clean set. Time did turn out to be the great healer in this matter, but scan times were markedly increased as a result, producing the slowest scanning of executables in the clean set by quite a margin. The OLE file scanning was not afflicted by this problem, neither was the viral test set to any noticeable degree.

Norman's polymorphic detection rates were well up on last year's performance, in accordance with other platforms for



the *Norman* product range where engine overhauls have been made across the board. These improvements are certainly good to see and result in a VB 100% award.

Sophos Anti-Virus v 3.48

ItW File	99.82%	Macro	99.67%
ItW File (o/a)	99.82%	Macro (o/a)	99.67%
Standard	99.15%	Polymorphic	95.36%

The *Sophos Anti-Virus* NLM retained the idiosyncrasies that make it somewhat less than pleasant to review, the most irksome of which are the small maximum size of log file and an inability to select sets of files easily for scanning using the installed list of program extensions. This list of extensions also proved to be the program's weak point as far as detections were concerned, since the .LNK and .BAT versions of W32/SirCam.A went undetected. The requirement for extra extensions to be added to the list has been added to the information in the IDE virus definition file compilations on the *Sophos* Web site, though unfortunately this innovation came too late to save company pride on this occasion.

Other than these rather problematic misses detection was elsewhere somewhat hindered by extension-related misses and those files not detected due to the overheads involved. One area where the *Sophos NLM*, and other *Sophos* products in general, still have problems due to detection-related issues is the polymorphic set where ACG.A still remains undetected.

VirusBuster VBShield for NetWare v 1.09

ItW File	100.00%	Macro	98.97%
ItW File (o/a)	100.00%	Macro (o/a)	98.97%
Standard	99.37%	Polymorphic	95.71%

And so to the last of the products on review this month. This was a 'hiccupy' newcomer in the last *NetWare* review, so its behaviour comes under careful scrutiny. The readability of report files seemed to have improved when displayed on-screen, though this initial improvement proved to be short-lived and detection was again performed by deletion.

The detection rate was where the majority of improvements lay, and these were vast indeed. None of the percentage detections in any category were above 96% in the previous review, with polymorphic viruses coming in at a lowly 77% rate. On this occasion the polymorphic detection rate is vastly improved with marked increases in the ItW test set – sufficiently improved, in fact, to warrant a VB 100% award. This increase in detection rates is certainly not a one-off occurrence either: it was noted in last year's *NetWare* review, and if it continues into the future more VB 100% awards are almost certain to follow.

Conclusions

As this test draws to a close, I ponder the comments made at the end of the last *NetWare* review. My conclusion is brief: the situation has not remained as dire as it was at the end of the last *NetWare* review. Improvements have been made by many products. Then again, there remain some odd behavioural traits in products which veer towards the sadistic. I suspect I shall be able to say exactly the same next year.

Technical Details

Server: 500 MHz AMD Athlon server with 6 GB HD, 64 MB RAM, CD-ROM and 3.5-inch floppy running *Novell NetWare* 5.11 with Service Pack 3.

Workstation: 750 MHz AMD Duron workstation with 128 MB Ram, 8 and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, running *Windows 98* with *Novell Client for Windows 95/98* version 3.30.00.0 Service Pack 3.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/NetWare/2001/07test_sets.html.

A full description of the results calculations 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, 50 Sth Audubon Road, Wakefield, MA 01880, USA

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

Don't miss out on VB2001! The Eleventh International Virus Bulletin Conference & Exhibition (VB2001) takes place on 27 and 28 September 2001 at the Hilton Prague. Reserve your place now: contact Bernadette Disborough; tel +44 1235 544034 or visit the *Virus Bulletin* Web site http://www.virusbtn.com/vb2001/ for a booking form and more details.

ISSE 2001 Information Security Solutions Europe takes place 26–28 September 2001 at the QEII Conference Centre, London. For further information email isse@eema.org, tel +44 1386 793028 or visit the Web site http://www.eema.org/isse/.

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

COMPSEC 2001 takes place 17–19 October 2001 at the Queen Elizabeth Conference Centre, London, UK. For more details visit the Web site http://www.compsec2001.com/ or contact Tracy Collier: tel +44 1865 843297; email t.collier@elsevier.co.uk.

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

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

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

Abstracts for EICAR 2002 must be submitted by 1 December 2001. Papers pertaining to malicious codes and unwanted side-effects or malfunction, information age, warfare and society, cryptography and the protection of privacy, new media and e-commerce are of interest. For more information about the conference visit the EICAR Web site at http://Conference.EICAR.org/.

***Viruses Revealed*, by David Harley, Urs Gattiker and Robert Slade** is a detailed guide offering full-scale analysis of the origin, structure and technology behind computer viruses and addressing the latest methods of virus detection and prevention. The book is published by *Osborne/McGraw-Hill* and is due out 7 September 2001.

***Panda Software*'s *Panda Antivirus for Linux* is on offer as freeware.** The program can be downloaded from the company's Web site http://www.pandasoftware.com/com/linux/linux.asp.

***F-Secure* and Finnish ISP *Sonera Plaza* have announced an** agreement with *F-Secure Online Solutions* to bring *F-Secure*'s automatic updating of information security to *Sonera*'s Internet clients. *Sonera Plaza Information Security* will offer anti-virus and firewall solutions based on *F-Secure*'s products. For further details visit http://www.europe.f-secure.com/.

***Kaspersky Labs* has announced the beta release of *Kaspersky Anti-Virus* for SMTP gateways**. The new program offers customers the opportunity of embedding a centralized anti-virus scanning for email independent of the type of server being used. The current version of *Kaspersky Anti-Virus* can be used on SMTP gateways running under the *Linux* operating system, and future versions will support FreeBSD, OpenBSD and Solaris (Intel/Sparc). The fully functioning product is due for release October 2001. See http://www.kaspersky.com/.

McAfee* has embarked on a development effort with DDoS solutions providers *Arbor Networks*, *Asta Networks* and *Mazu Networks to create a solution for DDoS attacks. The new threat management solution will not only monitor for anomalous traffic entering the network, but also detect the presence of Zombies within the network. For more details see http://www.nai.com/.

A beta version of *F-Prot Antivirus for Linux* has been released. Download a copy from http://www.frisk.is/f-prot/products/fplin.html.

Central Command* has released *AntiVirus eXpert Professional 6.0. In addition to on-access file and email protection, *AntiVirus eXpert* monitors behaviour and will block applications attempting to overwrite or modify system files, *Windows* registry or those trying to access the Internet without the user's consent. The software can be downloaded from http://www.centralcommand.com/products.html.