ISSN 0956-9979

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Oh brother where art thou?** Last month Peter Ferry investigated the inner workings of virus chart-topper W32/Klez. In this issue, he turns his attentions to what has been described as Klez's little brother: W32/Elkern. See p.8.

• **This message will self destruct …** Alex Czarnowski sets any budding secret agents what seems like an impossible mission: to deploy a totally secure web and ftp server based on *Windows NT*/*2000* with *IIS 5.0* using only vendor-supplied tools and open-source solutions. His clues for a successful mission start on p.10.

• **Arabian nights:** This is not the first time Eddy Willems' inquisitive nature has landed him in trouble in Saudi Arabia. Read the story on p.12.

• **NetWare and tear:** After his last two comparative reviews ventured onto new (to *VB*) platforms, Matt Ham returns to more familiar territory for this month's comparative. Find out how nine AV products fare on *NetWare* on p.17.

CONTENTS

COMMENT

“ *Sysadmins who rely on the AV industry are sometimes let down.* ”

Mines of Disinformation

In my copious free time between job title changes, I run a threat/vulnerability assessment service. This necessitates my logging in at frequent intervals to check email resources such as AV industry alerts and work-related information-sharing resources, peer networking resources such as AVIEN/EWS mailing lists, *SecurityFocus* lists, automatic alerts from virus-specific scanners and content filters, and enquiries forwarded via helpdesk resources, incident management personnel, and so on. Other resources include vendor and other AV-related websites, newsgroups and so on.

Do I need all these goodies? Well obviously there’s an enormous quantity of duplication. Some of the commercial offerings can be highly product-specific – sometimes they are little more than advertising. Sometimes they concern threats that will never be seen in the wild, although I appreciate that information needs to be available in case a virus remains obscure for a while and then gets lucky. Some will have limited impact on the organizations I am supposed to protect. (Other organizations with a less fragmented infrastructure may find them even less of a concern, since they are likely to have generic filtering and behaviour blocking software in place at suitable choke points on their networks, so that suspicious attachments are discarded at the gateway or on a server.)

I have to make a judgement call on which warnings need to be passed on, to whom (select lists of system administrators and helpdesk personnel, management, general customer bases), and in what form (on an intranet web page for short-life alerts, as a brief summary memo, or a full-blown advisory). There is no way I can forward everything. Even the system administrators I feed with information will baulk at the bulk, and more-or-less casual intranet browsers will quickly overdose and overdoze, so that the impact of the really important stuff is diluted drastically.

Even in the case of a threat that really needs widespread information sharing, it isn’t always enough (or even acceptable) to forward the email. Some mailing lists don’t allow unmassaged forwarding. Some, especially the peer network mailings, are likely to be of highly variable accuracy. Some will not include all the information that is most useful to my customer base. Vendor websites are often very accurate. That observation isn’t as obvious as it sounds – web pages aren’t necessarily assembled by top-flight researchers, and occasionally they include the most astonishing gaffes. However, even where they are accurate, they aren’t necessarily updated immediately to reflect a new or growing threat, and when they are they may not include all the necessary information.

Recently, I was contacted by a system manager who had the pleasure of removing a Javascript Trojan from a couple of PCs. Major AV sites she consulted (if they mentioned this particular Trojan family at all) simply noted that there was ‘No further information’ or advised that there was no payload and advocated removal of the file. No mention was made of the Registry hacking needed to find and remove the second copy of itself that this particular variant dropped elsewhere into the system, in order to stop it continuing to visit an unwanted website. Not the sort of incident that tends to make headlines or conference papers, but a certified nuisance nonetheless. This reminded me of happy days at the PC support coalface, restoring Registry settings on PCs left unusable after anti-virus software had completed half a disinfection, or more recent conversations with administrators who’d heard of Klez variants that forge the ‘From:’ header and needed to know how to identify the real sender (where possible) – a detail that virus information databases invariably omit.

I’ve heard arguments for omitting this sort of detail, usually centred around not giving the bad guys ideas or useful information. Unfortunately, too many genies have vacated their bottles in recent years to justify this sort of Security Through Obscurity. The bad guys can get this sort of information from anywhere: however, sysadmins who rely on the AV industry are sometimes let down. It seems that the industry is still having problems keeping up with the needs of its customers (especially, as always, the small accounts). Perhaps we need an independent but reliable source of information on new threats, like CERT or CIAC, but more focused on virus issues?

David Harley, Independent Researcher & Author, UK

NEWS

Retail Therapy

Symantec has been on a blow-out shopping spree. Perhaps it was its purchase of *Mountain Wave* earlier this year that put the company in the mood for splashing out – not content with just the one purchase, *Symantec* has snapped up three more companies in an impressive bout of spending.

Hot on the heels of reporting a 39 per cent growth in revenue in the first fiscal quarter, it was announced last month that *Symantec* was the proud new owner of: *Recourse Technologies*, whose main business is the provision of intrusion detection systems; managed security services provider *Riptech, Inc.*; and *SecurityFocus*, provider of threat alerts and host of the highly regarded *BugTraq* vulnerability mailing list. The company spent $135 million on *Recourse Technologies,* $145 million on *Riptech*, and *SecurityFocus* set the company back a mere $75 million.

In the AV world, response to the news has been varied, some voicing concern over the monopoly-forming tendencies displayed by *Symantec*, while many have predicted the demise of the *BugTraq* list (and coinciding with the announcement was the opening of a new full disclosure list, imaginatively named ‘Full Disclosure’). However, support for *SecurityFocus* co-founder Elias Levy and his team has not been in short supply, and sources at *Symantec* have claimed that *BugTraq* subscriptions have continued to rise since the acquisition.

While statements from the two companies have attempted to allay any fears of changes to the character of the *BugTraq* mailing list, we will have to wait until the dust has settled to find out whether such fears were well founded, as well as what impact *Symantec*’s purchases will have on the larger picture. With *NAI* having increased its exchange offer for outstanding publicly held shares of *McAfee.com* by 15.5%, could this be the start of an empire-expanding battle between the bigger players in the security market? ■

VB on your Web

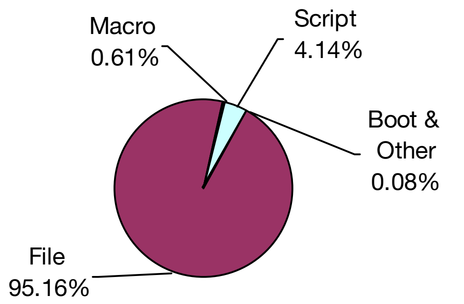
Times have moved on since ten years ago, when a mere 16 incidents of Form reported to *Virus Bulletin* meant that the virus qualified for the top position in *VB*’s virus prevalence table. While an archive of monthly prevalence tables dating back to 1995 can be found on the *VB* website, you can now add *VB*’s fully up-to-date virus prevalence information to your own website or intranet. *VB* provides a piece of JavaScript that can be copied and pasted to your web page where it can be tailored to achieve your preferred table layout. Alternatively, the prevalence information can be downloaded to your site as an RSS feed. For more information, including details about how the prevalence table is compiled and how you can become a contributor, see http://www.virusbtn.com/resources/prevalence/ ■

Prevalence Table – June 2002

Virus	Type	Incidents	Reports
Win32/Klez	File	6563	73.31%
Win32/SirCam	File	540	6.03%
Win32/Magistr	File	504	5.63%
Win32/Yaha	File	335	3.74%
VBSWG	Script	314	3.51%
Win32/BadTrans	File	186	2.08%
Win32/Hybris	File	71	0.79%
Win32/Frethem	File	55	0.61%
Win95/CIH	File	49	0.55%
Win32/Nimda	File	41	0.46%
Win32/Elkern	File	36	0.40%
Laroux	Macro	28	0.31%
Win32/Higuy	File	26	0.29%
Win32/Onamu	File	20	0.22%
Haptime	Script	19	0.21%
Kak	Script	18	0.20%
Win32/MTX	File	15	0.17%
Win32/Gibe	File	13	0.15%
LoveLetter	Script	10	0.11%
Win32/Fbound	File	10	0.11%
Win32/Aliz	File	9	0.10%
Win32/Ska	File	8	0.09%
Others [1]		82	0.92%
Total		8952	100%

[1] The Prevalence Table includes a total of 82 reports across 32 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

Low-Risk Linux

I believe that the threat of a virus for *Linux* is very, very low. To cause damage the virus needs to find a root exploit, and on most *Linux* systems, that is difficult. What this means is that a virus will not reach the critical infection ratio to spread. Even in two years time, when half the computers on the Internet will be *Linux*-based, viruses will not be a threat for *Linux*.

Shaun Savage
(Address not supplied)

No Comparison

While I appreciate Peter Morley's views on the AV industry (see *VB*, May 2002, p.16), I disagree with his comparison of *Linux* to DOS in terms of virus damage and his implication that the number of Trojans (and other malware) for *Linux* will equal that for *Windows*.

Linux is a vastly different OS from DOS and contains a lot of safeguards that DOS does not have – the greatest of all being the users and file permissions. A virus won't affect executable binaries on a computer because the binaries are owned by the root user and are not modifiable by general users.

A major reason why *Windows* is susceptible to viruses/Trojans/malware is that it is easily exploitable. I'm sure there are statistics that show how many viruses are written specifically for *Outlook*, *Outlook Express*, *MS Word Scripts*, etc. These problems would not exist if the programs concerned did not make it so easy to exploit them. This is another reason why *Linux* is a more secure platform.

Having programs that are GPL'ed and developed on an open basis enhances peer review such that the authors of these programs will *not* expose the same errors that many *Windows* programs have. In a sense, they're pressured into not making errors because everyone would cry foul and move on to different software if they did.

Of course, all of these defences go out the window when a *Linux* distribution with poor security is implemented (i.e. *Lindows* or any other distribution that uses root as its main user account). It is the users of these distributions that virus writers will target – and they will be successful.

Curtis H
(Address not supplied)

The Time is Nigh

It seems to be time for *Virus Bulletin* to have a more clearly-defined policy on updating software submitted for review. The current policy is (as I understand it) 'updates will be applied as of a certain date (the "submission deadline") and the resulting updated product is the one we'll test'.

This is all very well, and works wonderfully for products that either (a) in between releases, offer signature updates only; or (b) have a clearly-defined one-stop update process ('run this file to patch your installation', or 'press the UpdateMeNow button'). For the products in group (a), you download and apply the signature updates, either automatically or manually; for those in group (b), you download and run the file, or press the button. Easy.

Unfortunately, products exist for which updating is not (for one reason or another) a simple one-button or one-process deal – this is nicely demonstrated by *NAI*'s *VirusScan* in the *XP* comparative review (see original review, *VB* June 2002, and correction, *VB* July 2002).

Whilst it's true that the real question should be 'why are there multiple ways to update the same anti-virus product, some of which don't result in having "the latest version"?' (or even 'if the experts at *VB* made an error in updating the product, how are users expected to get it right?'), it's the job of the reviewer to define rules that enable him or her to justify his actions whilst performing a review.

Clearly, in the case of *NAI VirusScan*, something went awry and the reviewer's idea of how to update the product to the latest version was different from the supplier's idea of how to do that.

I would be delighted to hear suggestions as to a modified policy on updates that *VB* could use. Remember, however, that such a policy must be something that is easy to follow, hard to misinterpret, and applicable to the many different ways of updating AV products.

Ian Whalley
VB Consulting Editor, USA

Safe as Xboxes …

The suggestions made in the article 'Playing with Fire: Security on the Game' (see *VB*, July 2002, p.2) are infeasible at best, as there is very little that could be done to a games console.

Games consoles don't access the Net in quite the same way as a computer (except for the *Dreamcast*), and so would not be vulnerable to viruses in the same way as a computer. Not to mention the fact that the *PS2* has *no* internal memory. The *Xbox* can save to a hard drive, but only saves games, and cannot download extraneous data from the Net.

The only way in which a console could become infected would be by contact with infected burned copies of the game that had viruses added to them (which would be able to infiltrate the boot sector of the system). Since you don't have to install a game onto the console, that's probably not a feasible infection route either. It's not going to happen at any time in the near future.

Nate V.
Max's Game Corner

The Author Replies …

Microsoft's *Xbox Live* will be able to use an existing broadband connection. Therefore it seems highly likely that it will use TCP/IP or UDP, leaving me somewhat baffled as to how this is '[not] quite the same way' as a computer.

It's worth pointing out that, as a stateless protocol, UDP is trivial to spoof. Additionally, *Microsoft* has a history of producing TCP/IP sequence numbers that aren't very secure. If you can spoof packets, you can pass arbitrary data to an *Xbox*, and can, perhaps, cause unexpected behaviour.

The *PS2* may not have any internal memory, but it does have data persistence in the form of memory cards. However, that is a little beside the point because, to assume that the *Xbox* can save only game data to the hard drive would be to blindly believe that the system will work as intended – perhaps a little like saying that only BIOS updates can be written to the Flash BIOS – clearly a meltdown waiting to happen …

It does seem unlikely that one will be able to download saved games from the Internet, yet security patches, feature additions and other items of that ilk are likely to become available. Thus, the assumption that there will be no exploits that make games write arbitrary data to the hard drive is the kind of thing that I imagine keeps the imaginations of today's virus writers furtively exploring new levels of technological depravity.

Finally, many of the popular games magazines release demo CDs, but of course, *they* couldn't be infected, could they (cue dry laugh).

Pete Sergeant
Virus Bulletin, UK

CONFERENCE PREVIEW

Easy does it: VB2002

Helen Martin, Editor

In October 1990, the following announcement appeared in a slimline edition of *VB*: 'A *Virus Bulletin* conference will take place on 12–13 September 1991. The objectives of the conference are 1) *to present factual information about computer viruses*, 2) t*o demonstrate defensive procedures*, 3) *to discuss probable future virus developments and countermeasures* and 4) *to attempt to harmonise research efforts*.' Twelve years on, the objectives of the *VB* Conference remain unchanged (although some attendees might advocate the addition of objective number 5: to sample the local tipple and take full advantage of local bar facilities).

Since the inaugural *VB* conference in 1991, delegate numbers have risen from an initial audience of 150 to the bumper crop of more than 350 at VB2000 in Orlando. This year we have every reason to expect to exceed that number when VB2002 takes to the streets of New Orleans.

Over the years *VB*'s conferences have been eventful to say the least – some of the tales that spring to mind include a suspiciously high number of fire alarms (and a fire), a presentation by an ex-virus writer, delegates under hypnosis, a prominent member of the anti-virus community with his head in a working guillotine [*how did* that *happen? - Ed*] and countless jokes and japes (fuelled, I am (un?)reliably informed, by little more than local tap water and the sheer sense of the occasion).

Frivolity aside, *VB* conferences provide a focus for the AV industry, representing an opportunity for experts in the anti-virus arena to share their research interests, discuss methods and technologies and set new standards, as well as meet with – and learn from – those who put their technologies into practice in the real world. Delegates range from dedicated AV researchers to security experts from military organizations and large corporations worldwide.

VB2002 Programme

This year's programme is packed with the proverbial 'something for everyone'. Legal issues concerning the inadvertent transmission of viruses will be covered by Meiring de Villiers of Stanford University, while ex-*VB* editor Nick FitzGerald focuses on free AV techniques.

Speakers from the corporate sector include *IBM*'s Ed Hahn who will be looking at the evolution of managing viruses in a large corporation, John Alexander of *Wells Fargo* asking the question 'how squeaky are your wheels?' in a discussion of how the 'health' of a large user population might be measured, and *Microsoft*'s Randy Abrams explaining the corporation's automated virus-scanning system.

Robert Vibert, moderator of the Anti-Virus Information Exchange Network (AVIEN), will provide an update on how far AVIEN has come since it was an idea mooted over cocktails in the bar at VB2000.

On the more technical side, Kurt Natvig will present a follow-up to his VB2001 paper on Sandbox technology, while Sami Rautiainen looks at *Linux* backdoors and Markus Schmall probes the potential for malicious code on Java 2 ME. Anti-virus testing – a topic bound to result in some lively discussion – comes into the limelight with Andreas Marx's paper which focuses on retrospective testing of AV products. A *Panda Software* double act will present a paper on attacks on *.NET*, while *VB* old hands Eric Chien and Péter Ször discuss blended attacks.

The remaining highlights of the programme are too many to list here; we look forward to presentations of papers by some of the best experts in the field. The full details, including paper abstracts, can be found on the *VB* website (see http://www.virusbtn.com/conference/).

Work Hard … Play Harder?

The *Virus Bulletin* Conference has an impressive history of memorable entertainment, and it is a well known fact that much of the real, hardcore, cutting-edge 'work' that goes on at any conference is conducted not in meeting rooms, but in those informal 'breakout sessions' that take place in the bar.

This year *VB* will take full advantage of all the fun and frolics the non-stop party city New Orleans has to offer. Proceedings will kick off with a drinks reception with a difference as we take to the Mississippi on an authentic paddlewheeler. Without wanting to give too much (or anything!) away, it is with confidence that I predict that the Gala dinner will be spectacular and an unmissable event in itself.

Book Now!

Rooms at the Hyatt Regency will be held at a special conference rate until 27 August, so we advise you to register as early as possible. As usual, *VB* subscribers are entitled to a reduced conference registration rate. All that remains for me to say is I look forward to seeing you all in New Orleans. *Laissez les bons temps rouler*.

Conference:	VB2002, Hyatt Regency, New Orleans, LA, USA.
Dates:	26–27 September 2002.
Prices:	US$1595 non-subscribers; US$1395 *VB* subscribers.
Booking:	Telephone +44 1235 555139; email vb2002@virusbtn.com or download a booking form from http://www.virusbtn.com/conference/.

2002 NEW ORLEANS
www.virusbtn.com

VIRUS ANALYSIS 1

In the Spida's Web

Gabor Szappanos
VirusBuster, Hungary

There are two known variants of SQL/Spida. Both variants exploit the security hole described in *Microsoft* Knowledge Base article Q313418, which was published in January 2002: *MS SQL Server* versions 7.0 and 2000 install themselves with a blank default password for the SQL Server System Administrator (SA) account.

While variant A uses binary programs based on the SQLPoke utility for spreading, variant B uses Javascript for its main infection code. The following comments can be seen inside the second worm:

```
     "// sqlprocess v2.5"
     "// Greetings to whole Symantec anti-
  virus department."
```

The worm consists of a combination of binary files and Javascript files. These are stored in the system folder:

drivers/services.exe	port scanner
clemail.exe	SMTP mailer program
pwdump2.exe	SAM password dump utility
samdump.dll	DLL used by the password dumper
run.js	shell command execution wrapper
sqldir.js	tool to display database and table names
sqlexec.js	a command wrapper to execute SQL command on a remote computer by attaching to the DB provider and issuing commands using the xp_cmdshell command
sqlinstall.bat	batch file to install the worm
sqlprocess.js	the main worm spreading routine
timer.dll	an ActiveX DLL containing the necessary timer functions (Sleep)

When a new system is targeted for infection, the installation script sqlinstall.bat is executed. This spawns instances of the sqlexec.js script to execute commands via the bogus SA admin account. This script makes use of the extended, stored, procedures feature which is present in SQL servers and which enables functions to be called in DLLs outside the database. The worm uses the xp_cmdshell esp command to execute the shell commands passed as an argument.

The worm activates the guest account on the target PC, then adds it to the local administrators and the Domain Admins group. The latter is likely to fail if the local admin account is not a domain admin account. Next, the worm connects to the remote admin$ share. If it finds the file regedt32.exe in the *Windows* directory, the infection will be aborted. Note that, normally, this program is located in the system folder – the worm copies it into the *Windows* folder as an innocent-looking infection indicator.

Then the virus copies all of its files into the target system, and sets their attributes to hidden. Finally, the worm deactivates the guest account, removes it from the admin group, and sets the password of the SA account to a randomly generated four-letter string of lower case letters of the English alphabet. This prevents reinfection of the target.

Switching to the next phase, the main worm routine, sqlexec.js, is started on the compromised machine. First, this script checks whether its passed argument is 'init' (which is the case if the script is executed during the startup process on an infected computer). If the argument is 'init', the routine is executed again, this time with no argument. In this case the worm will only infect other computers.

If it was started with the 'init' switch, the virus registers itself for automatic startup as a system service with the key HKLM\System\CurrentControlSet\Services\NetDDE \ImagePath, and value '%COMSPEC% /c start netdde && sqlprocess init'. Note that a script cannot be executed as a system service during startup, but this way netdde.exe is loaded, and the '&&' argument will force it to execute the passed script.

If the PC is running *SQL Server* version 7, then the worm ensures the Winsock SQL connectivity by setting the HKLM\software\microsoft\mssqlserver\client\connectto \dsquery Registry key to 'dbmssocn'. Then it copies the file regedt32.exe from the *Windows* system folder to the *Windows* folder to mark the presence of the worm.

Next, the worm collects all sorts of information into the file send.txt. It runs ipconfig /all, and executes sqldir.js to gather information about the SQL database structure. Finally, it attempts to extract the domain user passwords using pwdump2.exe.

All of this data is emailed to ixltd@postone.com in a mail with the subject 'SystemData-' followed by the password of the SA account. The body of the message contains all the data that has been gathered.

Attack

Next comes the most interesting part of the worm – the attack against new SQL servers. The virus scans the network for possible targets in an endless loop. However, it does not pick the target in an entirely random fashion, but uses weighted random domain generation. It maintains an array of random class A domain addresses, and another one with the weights attributed to them. Then it fills a 1037-element array with the possible domain IP addresses – each with as many occurrences as its weight. This way, the domain 216 will appear 151 times, domain 64 appears 111 times, domain 211 appears 101 times and so on.

Then the virus generates a random number between 1 and 1235 and picks the appropriate element from the array. As the array contains only 1037 elements, there is a 16 per cent chance that the index will be invalid. In this case, the worm generates a random domain address between 1 and 223, then generates a random number between 0 and 255, appends it to the domain address, and calls services.exe to scan this subnet from *.1.1 to *.255.254 for possible targets.

This program attempts to connect to the 1433 port of the addresses within this range. If a vulnerable SQL server is identified, its IP address will be appended to the text file rdata.txt. After one port scanner cycle is terminated, sqlprocess.js reads all target addresses from this file and launches sqlinstall.bat to penetrate the target.

SQLSpida.B avoids the non-public IP ranges by skipping the IP domain if it is 10, 127, 172 or 192, which contain the internally usable IP subnets (and the loop-back address). It may not be the most efficient filtering system, as usable IP ranges are lost within these domains, but it is good enough for a virus. It is interesting, though, that the array contains the domain 192, which will be skipped anyway during this check. Maybe this was a minor oversight by the virus author, who may have generated the array and the weights from publicly available domain statistics.

IP Distribution Statistics

Earlier worms (including SQLSpida.A itself) used random target IP selection, which resulted in a huge number of wasted probes, targeting non-existing IP domains. An improvement is observable in CodeRed.B, which skewed the target range to include more addresses from the subnet of the infected PC. However, this is nothing compared to SQLSpida.B, which uses a very purposeful generation mechanism based on empirical data. But was it worth the effort? Did it increase the worm's chances of survival?

This question can be answered by analysing the data collected about the worm. The ten most prevalent target domains picked by the worm algorithm are as follows:

Domain	Prevalence
216	12.23%
64	8.99%
211	8.18%
209	5.02%
210	3.97%
212	3.64%
206	3.48%
61	3.24%
63	2.91%
202	2.91%

We also have (with acknowledgment to Roger Thompson for WormCatcher data and Costin Raiu for his Smallpot statistics) the observed source domain for 1433 port probes, which showed the following prevalence (with a total of 954 hits):

Domain	Prevalence
211	16.88%
210	7.23%
216	6.39%
61	6.08%
66	5.77%
202	5.14%
209	4.61%
64	4.3%
203	4.19%
207	4.09%

Similar data has been gathered by the SANS (System Administration, Networking and Security) Institute. Their data indicates a huge increase in port 1433 probes after about 19 May, and this port is still the second most probed port since then – which indicates that the number of infected computers has reached saturation level and almost every possible target is infected now.

The SANS domain prevalence list is not directly relevant to this analysis, as it does not provide per-port statistics, only accumulated statistics on all ports. However the fact that, on a selected day, 60 per cent of all port probes came on port 1433, could serve as an indicator. These statistics show similar distribution with the subnets targeted by the virus in the top quarter of the list.

It is important to note that the subnet statistics of the worm algorithm show the possible targets for infection, while the source statistics for the probes show the distribution of the infected computers. If the two sets of statistics show strong similarity, we can conclude that the strategy used by the virus is successful.

It is quite obvious from the WormCatcher/Smallpot statistics that the worm used a good strategy. Of the ten most frequently generated subnets seven are present on the probe list, with more-or-less conforming prevalence. This not only means that the worm was successful in finding populated subnets, but that it was also successful in finding vulnerable SQL servers within those subnets.

Conclusion

There is no question that worms are evolving. An improvement has been seen in many other worms in their target selection. In the future we can expect to see new algorithms, based on available domain population statistics, being used, thus avoiding the large number of wasted probes and increasing the worms' chances of propagation.

VIRUS ANALYSIS 2

Un combate con el Kerñado

Peter Ferrie
Symantec Security Response, Australia

W32/Elkern could be considered the 'little brother' of W32/Klez. Even though Klez carries the Elkern virus and runs it on the machines that Klez infects, it is Klez that has received all the attention. Little mention is ever made of Elkern, and some of the details of its behaviour have remained unexplained. They are described here.

There are three variants of Elkern. The first, which is 3326 bytes long, is carried by Klez variants A to D, F and G; the second Elkern variant, which is 3587 bytes long, is carried by Klez.E, and the third, which is 4926 bytes long, is carried by Klez variants H to L.

Elkern.3326 and Elkern.3587

Both Elkern.3326 and Elkern.3587 can exist in two formats: as a DLL or as an executable file. When the viral code gains control for the first time, if it is loaded as an executable file, it will always run, but if it is loaded as a DLL the viral code will run only during the DLL_PROCESS_ATTACH event.

Windows N(o)T

If the code is run, Elkern will search memory for kernel32.dll and get the addresses of the APIs that it requires. The first major bug in the virus occurs here: the API names are converted to a 32-bit CRC value, but Elkern compares only the lower 16 bits of this value. This results in the retrieval of the wrong API addresses under *Windows NT*, where several of the calculated values differ only in the upper 16 bits.

This mistake has been made repeatedly by virus authors, including the author of W32/Kriz, and is likely to continue as the majority of computer users (including virus authors) skip *Windows NT* in favour of *Windows 2000* and *XP*.

If a debugger seems to be running, Elkern will stop running at this time.

Must Run, Back Soon

If Elkern was not loaded as a DLL, it will copy itself to %system% and alter the Registry. Under *Windows 9x/ME*, the filename will be 'wqk.exe' and the Registry entry 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run' will contain a value called 'WQK', which points to %system%\wqk.exe.

Elkern will also call the RegisterServiceProcess() API, if it exists, in order to remove the Elkern process from the task list. Under *Windows 2000*/*XP*, the filename will be 'wqk.dll' and the Registry entry 'HKLM\Software\Microsoft\WindowsNT\CurrentVersion\Windows' will contain the value 'AppInit_DLLs', pointing to 'wqk.dll'. Any previous data for this value are lost.

The AppInit_DLLs is an interesting value. It exists in *Windows NT*/*2000*/*XP*, and the files in the value data are loaded into the process memory of all processes that run after the Registry change has been made.

Furthermore, if the computer is rebooted, these files will load into critical system processes, such as Winlogon. This poses a problem for anti-virus software that terminates processes containing viral code: terminating the Winlogon process will cause *Windows* to display the dreaded blue screen of death.

Elkern calls the routine repeatedly to copy the file and alter the Registry, at random intervals from one to seven seconds, requiring much speed (or luck) in order to disable it successfully.

What are my Chances?

At this point, Elkern enters the loop that searches repeatedly for files to infect. Before each location is searched, Elkern will check whether the payload should activate. The payload will always activate on 13 March and 13 September, but there is a small chance that the payload will be activated regardless of the date.

Though small in isolation, the chance of payload activation is increased greatly by the repeated checking process.

Elkern begins searching for files to infect in %system% and in the current directory. Next it will search on drive letters, beginning with a random letter and continuing until it reaches Z, before resuming from A. Under *Windows 2000*/*XP*, or if the WQK file was the one that launched this code, Elkern will also enumerate open shares on the local network to find files to infect.

Fuel Injection

During the file search, Elkern will open every file, regardless of extension. If the payload has been activated, Elkern will overwrite the entire file with zeros. If the payload has not been activated, Elkern will examine the file for its potential to be infected. Files will be infected if they are at least 8 KB PE files and are neither WinZip self-extractors nor DLLs.

Elkern.3587 also avoids RAR self-extractors, and files protected by the System File Protection. The infection method is very similar to that used by W95/CIH. The viral code is split into a linked list of blocks that are placed in the unused space at the end of sections in the file.

Since Elkern is so large, it will increase the size of the last section if there is insufficient unused space available elsewhere in the file. The entry point is altered to point directly to the Elkern code. Elkern.3587 will recalculate the checksum if one existed before.

Elkern.4926

If the previous Elkerns were a brick wall, then Elkern.4926 would be a rock wall constructed without mortar. It looks like a hurried work, unfinished and fatally buggy. It exists only as an executable file infector. It contains some of the same bugs that exist in the previous Elkerns (for example, the 16-bit comparison of the CRC32 value).

Whenever Elkern.4926 is run, it alters its appearance slightly. Elkern has many subroutines that are encrypted individually, and whose keys are altered each time the subroutines are used.

Additionally, Elkern has several routines for altering the code of several other routines, however these alterations are limited to register replacement and alternative encodings of some instructions.

Dude, where's my Code?

Elkern.4926 will inject its code into the memory of certain processes. If the process enumeration functions are found, Elkern will open all processes under *Windows 2000*/*XP*, and any process whose name contains '\explorer' under *Windows 9x*/*ME*.

If the enumeration functions are not found, Elkern will attempt to open any accessible process, by cycling through 20,000 different process IDs. Once Elkern has opened a process, it will read from a fixed image base value of 0x400000. This is unusual behaviour because the true image base of a process can be retrieved using the enumeration functions.

Elkern will then search the import table of the process for a reference to 'user'. If this is found, Elkern will search a random number (0–63) of imports for either the DispatchMessageA function or the DispatchMessageW function.

Regardless of the success of the search, Elkern will hook an import. If the search was successful, Elkern will hook the last import that was examined; otherwise, it will hook the second last import that was examined. This routine is executed repeatedly, with a small delay between each run.

Elkern begins searching for files to infect in the current directory. Then it searches on drive letters, beginning with a random letter and continuing until Z is reached, before resuming from A. It will also enumerate open shares on the local network to find files to infect.

The file search will skip directories that contain 'rary Inter' or 'tem32\dllcac'. A misfeature of the name comparison algorithm is that files and directories are also skipped if they begin with certain characters, such as the letter 'n'.

Additionally, files are skipped if they begin with any of the following: _avp, aler, amon, anti, nod3, npss, nres, nsch, n32s, avwi, scan, f-st, f-pr, avp, nav.

Considering the number of names in this list that begin with 'n', it appears that the virus author is unaware of the comparison bug. Files will be examined if their suffix is .exe or .scr, but there is a small chance that files with other extensions will be examined too.

Elkern considers a file to be infectable if it is a PE GUI or console application that is not a DLL, does not contain the text 'irus', is not protected by the System File Checker that is present in *Windows 98*/*ME*/*2000*/*XP*, and is neither a WinZip nor RAR self-extractor.

There is also a process to check whether the file is already infected but, due to a bug in the virus, this check always fails. The result is that files are reinfected repeatedly, eventually becoming too large to execute.

The file infection procedure for Elkern.4926 is identical to that of the previous variants: the viral code is split into a linked list of blocks that are placed in the unused space at the end of sections in the file, and the size of the last section will be increased if there is insufficient unused space available elsewhere in the file.

If the file contains relocations near the entry point, the entry point will be altered to point directly to the Elkern code. Otherwise, Elkern will place a jump at the original entry point that will point to the Elkern code. If the host contained a checksum, Elkern will recalculate it now.

Conclusion

W32/Elkern shows how even a buggy virus can become widespread, by being associated with a virus that is even more prolific.

Fortunately, Elkern does not stand well on its own. For the moment, at least, this battle is half over.

W32/Elkern

Type:	Memory-resident parasitic appender/inserter.
Infects:	*Windows* Portable Executable files.
Payload:	Elkern.3326, and .3587 overwrite all files on 13 March and 13 September. Elkern.4926 has no payload.
Removal:	Delete infected files and restore them from backup.

TUTORIAL

Mission Impossible – Part 1

Aleksander Czarnowski
AVET Information and Network Security

Mission Briefing

Your mission, should you choose to accept it, is to deploy a web and ftp server based on *Windows NT*/*2000* with *IIS 5.0* in such a way that it will be immune to any current and future worm attacks of the sort seen last year with the appearance of Nimda. During the mission you may use only vendor-supplied tools and open-source solutions. You have less than 24 hours, starting from now …

The Key: Careful Planning

This mission can be accomplished, no matter what you think about *Windows* and *IIS* security. However, you need to employ the right approach. The secret is careful planning.

First, consider what network services you really need to deploy, then add to this list native *Windows NT* services. Do you really need an ftp server (the same content can be served though HTTP)? What *IIS* authorization methods will you use? Do you need SSL? How will you administer the server remotely? Are you planning to install any additional servers such as *Exchange 2000* on this host?

Installation and Hardening

The process of installing *Windows NT*/*2000* in a secure manner has been described in great detail (see Stefan Norberg, *Securing Windows NT/2000 Servers for the Internet*, O'Reilly, November 2000, ISBN 1-56592-768-0), so I shall concentrate only on the parts that are important for the *IIS* server.

First, you should use only NTFS partition, as NTFS supports ACLs (Access Control Lists). This is a crucial security feature, and one you will be using extensively.

You will also need a number of disk partitions: never install everything on one partition. The rationale behind this is the ability to separate WWW and ftp directory structures (Inetpub) from system binaries.

As most attacks are not able to cross partition boundaries, an HTTP request such as '/scripts/..%5c../winnt/system32/cmd.exe' or 'msadc/..%c0%af../..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir' would fail even on unpatched servers. Of course, unpatched servers would still be vulnerable, but the risk of attack would be minimized.

So we need at least one partition for system files, one for the Inetpub directory structure and one for storing log files. If you are installing *IIS* as a requirement for *Exchange 2000*, you also need at least one additional partition for *Exchange* mailboxes.

If you look again at the aforementioned HTTP requests, you will see that both of them assume default names for the Windows and System directories. You can change those settings, together with the default system drive (C:), during *Windows* installation.

This will protect your server against attacks that rely on default configuration, but remember that if the attacker can run code of his choice on your system, nothing will stop him/her from using the GetSystemDirectory() and GetWindowsDirectory() Win32 API functions to retrieve true paths.

Internet servers should be on a server that is an isolated *Windows* domain. Do not define any trust relationships (in any direction) between *IIS* server and any other systems in your networks.

Unfortunately, this approach will not always work where *Exchange 2000* servers are concerned. Integrating *Exchange* with *Active Directory* is a wonderful idea, but it will not work very well on an isolated domain.

On the other hand, such a server should not be used as a web and ftp server. Installations like this are common on intranets, where users authenticate to the domain and this process grants them access, through *Outlook*, to *Exchange* resources including mailboxes, public folders and so on. In this case, you should disable OWA (Outlook Web Access). You can run *Exchange 2000* server (together with the required *IIS*), use public folders extensively and make the server secure.

For remote administration I would advise Terminal Service with a high encryption pack. This allows you to administer your *IIS* through encrypted communication in a very comfortable way.

Don't run telnet server. If you don't trust TS, you can install SSH server, or it is even possible to install *OpenSSH* which is a free, open-source SSH server implementation which supports both the SSHv1 and SSHv2 protocols.

If you install any additional components, remember their requirements and the consequences of such actions. For example, installation of *Certificate Server* will disallow any hostname changes.

During installation and hardening the server should not be connected to any network. During the configuration of TCP/IP settings the system might want to check whether the network interface is up. In this case, connect the system with null cable to another computer, or simply plug it into an empty hub.

Service Packs and Hot-Fixes

After successful installation apply all available Service Packs (at the time of writing SP2 was the latest) and hot-fixes. During this process it is not advisable to use hfnetchk or MSBA to check for missing patches, as both of these tools have some security problems (which will be described in the second part of this tutorial).

Services

Windows 2000 comes with many network services that should be disabled. Again, consult Stefan Norberg's *Securing Windows NT/2000 Servers for the Internet*, as well as *Microsoft*'s 'Windows 2000 Server Baseline Security Checklist' (see http://www.microsoft.com/technet/security/tools/chklist/w2ksvrcl.asp) to check which services you should disable, according to your requirements.

However, in all cases you should disable Simple TCP/IP services, and you won't need DHCP and SNMP on your Internet server (those who disabled their SNMP server a long time ago were able to sit back without any worries through one of the more recent Microsoft Security Bulletin scares).

The server starts many system services by default – disable as many of these as you can (usually this includes services such as Alerter or Spooler). *Microsoft*'s 'List of Services Needed to Run a Secure IIS Computer' (Microsoft Knowledge base article Q189271) provides a list of the minimal services required both by the system and *IIS 4.0*.

Log Files

You can change the default location of log files through the Registry entries. This, together with proper ACL settings, will protect the logs further. You should also reconfigure log file sizes and their storage option.

It is not possible to describe one proper log setting method for every configuration, so you will have to do it yourself. Basically, your log size should be large enough to hold at least 14–31 days' worth of events.

You should never overwrite logs automatically – always back them up first. It is possible to set up a server in such a manner that it will crash if logs are full. When setting maximum log file sizes, keep in mind the size of your log partition.

Null Sessions

You should limit null sessions. This can be done easily through MMC snap-in (which can be found under Security Policy; Security Options) or by editing the Registry key HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Control\LSA\RestrictAnonymous (always use regedt32 instead of regedit).

The recommended value for this Registry key is 1, although *Windows 2000* does accept setting the RestrictAnonymous value to 2. (A good description of the possible settings for the RestrictAnonymous value can be found in Timothy M. Mullen's article on the *SecurityFocus* website; see http://online.securityfocus.com/infocus/1352.)

Registry Access

The next step is to limit anonymous access to the Registry itself. This can be done by setting appropriate permission on the key: HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\Control\SecurePipeServers\winreg.

Administrators should be granted full control rights and any additional users and groups that have been granted access should be removed. Setting ACLs on the Registry is possible only by using regedt32.

If you just wanted to Exchange

If you installed *IIS* as a requirement for *Exchange 2000* and you do not need web access for mailboxes and public folders, you can protect *IIS* and *Exchange* very effectively using just a few mouse clicks.

All *IIS* web services need to be running if you deploy public folders (accessible through *Outlook*). But you don't need to allow everyone to connect to *IIS* web server, so you can limit access by setting IP restriction through IIS MMC snap-in: grant access only to 127.0.0.1 and to your server IP addresses.

This will enable *Exchange* features, while closing access to the most risky *IIS* services. In such a configuration only DoS attacks with spoofed IPs would be possible, but any remote penetration attempt would fail.

Milestone

Now we have *Windows 2000 Server* running *IIS*. We still need to harden *IIS* by changing its default settings and by adding several tools. Your *IIS* installation is far from being secure.

Don't go away with the idea that the installation of Service Packs and hot-fixes are the only ways of making *Microsoft* products secure. Certainly these are crucial for security, but they are by no means the only part of the securing process.

In the next part of this tutorial (in next month's issue of *VB*) we will look at the hfnetchk, urlscan and iislockdown tools, together with *IIS* and *ISA Server 2000* settings. Until then, do not connect your *IIS* to the network and if you have a spare moment, take a look at *Microsoft*'s IIS 5.0 Baseline Security Checklist, at http://www.microsoft.com/technet/security/tools/chklist/iis5cl.asp.

FEATURE 1

Virus Hunting in Saudi Arabia – Part 2

Eddy Willems
Data Alert International, Belgium

[*Last year (see* VB*, October 2001, p.10), Eddy Willems related the terrible tale of the computer virus horrors he witnessed at the hands of Saudi customs officials. Earlier this year he returned to Saudi, where his inquisitive nature led him into trouble of a different kind.*]

Whenever I visit a new city I like to spend a day wandering around, exploring and getting a feel of the place and its people. My experience is that, by night, many places have a completely different atmosphere. People have warned me that sometimes I venture too far and that I could be putting myself in danger. But, when they advise me not to visit certain areas of the town, I am intrigued as to why. I have visited some infamously dangerous areas in cities both in the US and Europe. Now, I have felt the same atmosphere in the Middle East.

Arabian Nights

In the Middle East it is very pleasant to go out at night because that is the only time when the climate is bearable for walking around. If you venture outside at noon you may find yourself being barbecued by the sun! In the summer the temperature can rise to about 45 ºC easily, whereas at night a slightly cooler temperature of between 20 ºC and 30 ºC is very pleasant.

All the shops in Saudi seem to stay open very late into the night. Usually when exploring a new city I try to find the electronics shops, so one evening I asked the hotel receptionist to direct me to the area of the town in which these were located. I followed his directions to a large square downtown.

The square was completely filled with small computer shops and some larger electronics shops. As I arrived, a man approached me. I assumed he was about to ask me for something and thought he must be a drug addict. But it seems that I had jumped to the wrong conclusion: ‘Hi, do you want some software, CDs, music or DVDs?’ he asked. A little puzzled, I replied, ‘No thank you’, and walked on.

About five metres further on another man approached me with the same question. This continued until I had been offered the chance to buy software or music CDs about fifty times.

Suddenly I had an idea: what if you really wanted some software … what if you asked for anti-virus software?

Place your Orders

I began walking around the square again. Nearly every man I spoke to seemed to have some kind of anti-virus software on his list (each of these ‘vendors’ has lists from which you can choose the software; once they have taken your ‘order’ they go away and return with everything you requested copied onto one CD).

Prices seem to vary from 10 to 40 Saudi Rials (approximately 0.3 Euro = 1 SR) for one product or program. After a while I started asking for corporate anti-virus software. This was not so easy to come by. Most of the men didn’t know what ‘corporate’ meant – the majority of them were ‘illegals’ (with no visa) and seemed to have no computer knowledge whatsoever.

After a while, I realized that one man was following me very closely. After the 25th man I passed, this guy approached me and said that he had heard what I was searching for. He asked me to follow him and led me to a narrow alley…

Everything under the Sun

After going through a small door we entered a building where we climbed two levels up some broken stairs, then he asked me to wait in a small dark room. After a while he returned and asked me which anti-virus product I wanted.

‘What do you have?’ I asked the man. ‘Everything!’, he exclaimed as he showed me into a room containing a PC. He inserted a DVD into the machine and asked me to make my selection.

I saw every latest version of nearly every AV package I could think of – even *NAI ePO server 2.5.0, Symantec System Center 7.5* and *Trend’s NeaTSuite* were on the DVD. The price was 30 SR.

I told the man that I was not really interested as I hadn’t found what I was looking for. It looked to me as if only one or two anti-virus packages were missing, so I told him I wanted a package (*eSafe*) that wasn’t on the DVD, since I was keen to leave as quickly as possible. A little upset, the man explained that he couldn’t have everything.

Surprise!

At that moment the man asked me something I was really not expecting: ‘Maybe I can help you with some computer viruses?’ he said, ‘What do you think?’.

Taken by surprise, I asked him what he could give me. The man left the room and, after a few minutes, he returned with another DVD.

He explained that this was the best virus DVD available. ‘More than 300,000 different viruses – even undetectables!’ he told me. ‘That’s impossible,’ I told him. When he realized I knew the field well, he conceded that there were about 32,000 viruses on the DVD. ‘Will you take it?’ he asked again.

I was horrified by this proposal, but I was quite intrigued. I asked the man how he had obtained this collection. He told me that he knew a man who wrote viruses and that this DVD had been the man’s own private collection. However, since having been married, the man was no longer interested in viruses and had given the collection away.

After hesitating a while, I repeated that I was not interested in buying the virus collection. This time the man became angry and asked me to pay him 100 SR (approximately 30 Euros). Again I stated that I was not interested, but the man started shouting at me and I felt very intimidated.

A little nervous because of the strange environment and very confused, I gave the man some money. He threw the DVD at me and asked me to leave immediately. Before I left he advised me to say nothing about this deal and to ‘forget’ him.

The Analysis

On my hasty return to the hotel I hoped that I had not been ripped off by this wretched deal. Once I reached my hotel room I very quickly booted my notebook and searched for the scanners I had brought with me. Only three of them were up to date.

First, I discovered that the DVD was at least readable, though not fully used. Nevertheless, there were 30,751 files on the disk. It seemed that the DVD was not a copy of some known CD on the Internet like the ‘old’ Digital Hackers’ Alliance virus CD or others and it didn’t look like a collection from an anti-virus vendor either.

The DVD contained a mixture of executables, zip and rar files, docs, xls and some html files. Within most of the (few) archived files, I found just one or two other files. This brought the total up to 31,657 different files.

I used *NAI VirusScan 4.5.1 SP 1* with 4160 Engine and Dat file 4205, as well as *AVP Pro 4.0* and *Symantec NAV CE 7.6*, both with the latest (May 2002) update. It appeared that every single file was, indeed, infected – although not each with a different virus.

I found exactly 31,655 different viruses. This indicated that the DVD had been prepared properly – otherwise I would have found many more uninfected files. This is a huge number for such a collection.

I did not find any new, undetected viruses, although some of those I found on the disk were relatively recent (e.g. W32/Yaha.c@MM). Nevertheless, I did find viruses which don’t appear frequently and are classed as Zoo viruses, such as V5M/unstable (a proof of concept virus written in VBA for *Visio 2000*). The viruses themselves were not always named or classified. In most cases the viruses were not even replicated.

Where the macro viruses were concerned, I easily located the real content inside the files. Most of the files seemed to have come from European corporates – I found it puzzling that these had appeared on this side of the world. I’m unsure whether the man was completely honest about the details of his virus-writing acquaintance.

The Lesson

One of the last questions I asked the man was whether there was a lot of demand for these CDs.

He told me that I was one of the first to have asked for anti-virus software. It seems that most people ask for some specific OS software like *Windows XP* or *Windows 2000*. He had come across the virus CD by coincidence. He told me that he had received only a few special requests for it, and explained that there was significantly more demand for ‘good’ hacker tools at the moment.

I advised the man not to sell any more of this kind of CD or software because of the trouble he could get into with the authorities. He told me that there had already been several raids that attempted to put a stop to this illegal dealing. However, he explained that virus writing is very easy to get away with in this region, because of the lack of effective laws relating to computer crime.

I think that I underestimated this man when I stormed out of that dark alley at around midnight that night. He knew exactly how many viruses were on the DVD, and he gave me a detailed explanation of the laws concerning computer crime.

It seems that I had stumbled across a man who was not typical of these illegal software vendors. It occurred to me that he was the only one who seemed to have any knowledge about computers.

Later on, I contacted the local police about these practices (which was an adventure in itself!). On my next visit to the city I found out that a raid had been carried out by the police and most of the men had been arrested.

Dangerous Corners

Again I have been surprised in a land that I didn’t know very well. I do not condone the sort of practice I experienced, but I couldn’t prevent it – would the man have stayed calm had I not agreed to the deal?

I have learned that drug dealing, software dealing and even virus dealing lie in the same dangerous corners of our society. I hope I never have to write part three of this series – but my work will bring me back to the same region later this year, so watch this space!

FEATURE 2

Abacus, EFI and Anti-Virus

Oleg Petrovsky
Computer Associates, USA

The very first computer did not have to boot. The abacus, a piece of hardware invented back in 3 AD, was ready to operate as soon as you looked at it. Furthermore, it was as secure as a rock – sometimes literally.

One can debate *ad nauseum* about whether the abacus was the first prototype of modern computers, but the fact remains that the complexity of computer devices and their applications has made a gigantic leap since then.

To Boot or Not to Boot …

Nowadays we find ourselves surrounded by computers without even realizing it. The production of embedded and general-purpose computers increases steadily each year. The complexity of integrated circuits has increased by a factor of eight over the past six years, as has the complexity of the software used by computer systems.

In 1971, the year the first microprocessor was manufactured, the computer market was dominated by mainframes. The processors were a long way from being micro. Usually a processor block would be confined in a fridge-sized area and a strong belief persisted that they required the fridge itself to keep their temperature down.

The boot process of such a computer involved executing a proprietary code stored permanently in the Read Only Memory (ROM) of the computer system. One of the major functions of that code was to initialize the pre-boot environment devices supported by the computer, find a medium that had an operating system (OS) on it, load it into conventional memory and transfer an execution control to the OS kernel.

That particular piece of code was bound to the hardware it was supporting and in most cases it required a major rewrite once an operating system changed or a new device, which had to be supported in the pre-boot environment, was added to the mainframe.

It is ironic that in present times, when the computer industry is flooded with hundreds of different types of microprocessor, systems still rely on the hardware-specific proprietary code stored in a computer system's ROM as an interface to the computer's hardware during the boot-up sequence. Since all the system's device-supporting code is stored in the ROM, and is largely undocumented and written in assembly language, it is extremely difficult to add new hardware devices to the system or to upgrade legacy device driver code for the pre-boot environment.

EFI to the Rescue

The Extensible Firmware Interface (EFI) standard emerged as a logical step to provide flexibility and extensibility to boot sequence processes, enabling the complete abstraction of a system's BIOS interface from the system's hardware. In doing so, this provided the means of standardizing a boot-up sequence, extending device drivers and boot time applications' portability to non PC-AT-based architectures, including embedded systems like Internet appliances, TV Internet set-top boxes and 64-bit *Itanium* platforms.

The Big Picture

Figure 1 (below) shows a simplified representation of the major EFI layers on the right as they compare to the legacy-booting environment on the left. For *Itanium*-based systems EFI implementation will usually consist of EFI boot manager, EFI defined system partition (FAT 32), the set of EFI applications used in the EFI shell mode, the set of EFI device drivers and finally the OS loader.

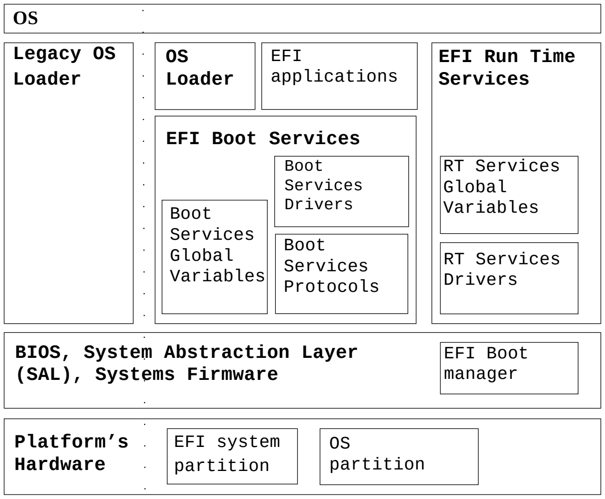


Figure 1. EFI structural block diagram.

The EFI boot manager would normally be placed in Non Volatile Random Access Memory (NVRAM). The NVRAM is protected and needs to be unlocked before information can be written to it. This is done at the software level to simplify the EFI boot manager upgrade procedure.

During the system power up the boot manager is set to be the first piece of code to receive execution control. However, on the PC-AT BIOS-aware systems, which currently persist on the hardware market, the boot-up sequence might still go through the initial BIOS initialization procedures before transferring control to the EFI boot manager.

In cases where the BIOS receives control first during a boot-up sequence, the EFI boot manager code needs to be placed on the system's hard drive or on some other media, to which the BIOS will transfer control after the Power On Self Test (POST) routine.

The boot manager's main function is to load up the system drivers needed during the boot-up sequence and to pass control to an EFI application. An EFI application is a binary file stored in PE32+ format on media accessible by the boot manager.

The range of the EFI applications would consist of the following choices: a first stage Operating System Loader, EFI shell environment, the shell's external commands or an EFI boot maintenance program. The boot manager is controlled by the set of data structures defined by the EFI specification. These structures can be administered through the boot maintenance program, allowing the boot manager to contain a menu pointing to all available boot options, including all installed operating systems and other EFI applications.

EFI Services

All services which are available under the EFI boot environment can generally be divided into two groups, namely, Boot services and Run Time services. Boot services are available only during the boot-up sequence and are terminated by the ExitBootServices() call made from within the first stage OS loader just before passing control to the entry point of the OS kernel.

The list of functions provided by Boot services is well documented in the EFI specification and is beyond the scope of this article; suffice it to say that Boot services provide access to Simple Network Protocol, File System Protocol, Preboot Execution Environment (PXE) Base Code and PXE Base Call Back protocols.

Boot services are also available to the EFI Shell application and to all its internal and external commands. The Run Time services are available after the OS is loaded and can be accessed by the OS kernel or OS applications. Run Time services export interfaces to EFI system variables and the computer system clock, and also make it possible to reset the entire platform by invoking the ResetSystem() function.

EFI System Partition

One of the booting options supported by EFI, and in which we are most interested, is booting from an arbitrary block device containing an EFI system partition.

A block device is usually a hard drive with MBR or Global Unique Identifier Partition Table (GUIDPT or GPT) schemes, or removable media such as LS120 or ZIP type drives. The GPT exists as a self-identifying structure and does not require the EFI environment to function.

All information that is required for the GPT scheme to operate is stored in well-documented, specified locations on the physical media on which GPT resides. It is worth noting that GPT disks may contain 2^64 blocks and have a secondary partition for the CRC32 integrity check of contained data. The system partition defined by EFI does not have to have its first sector modified in order to support the EFI booting sequence. This feature allows a transition stage where legacy AT-PC and EFI booting environments can co-exist on the same system platform. The system partition contains EFI applications, EFI drivers and EFI boot loaders.

EFI Shell and OS Loader

As noted earlier, EFI applications can represent external commands, which are available in the EFI shell mode. The EFI shell itself is an application that can be invoked from the boot manager. The EFI shell presents a convenient environment for testing EFI device drivers and is also a substitute for the DOS environment on *Itanium*-based systems.

The EFI boot loader is a special type of application which does not transfer control to the EFI environment on completion. Instead, it allocates system memory for the OS kernel, loads it up from one of the pre-defined locations, frees system resources allocated by the EFI environment and passes control to the entry point of the OS kernel image. The predefined location of the OS kernel is determined by EFI system variables set earlier by the boot maintenance program or, alternatively, by the default boot sequence hard coded inside the boot manager according to the EFI specification.

Someone is Outside the EFI Door …

Upon analysis of the security implications of the EFI *Itanium* implementation, it was noted that it is possible to modify files that reside on the EFI system partition. In doing so, it is possible to substitute an OS loader with an arbitrary EFI application performing some additional functions before OS kernel loading and invocation.

The accessibility of the Boot services allows the arbitrary EFI application to access the TCP/IP stack built on top of the Simple Network Protocol and implement SMTP, ftp, POP3 or other TCP/IP-based protocols. It also allows the arbitrary EFI application to modify files accessible locally through the EFI File System Protocol.

As noted previously, all EFI-style applications are stored in PE32+ format with the special signature in the header which distinguishes EFI images from other PE32 executables. The '+' indicates that some of the PE header fields are extended from four bytes to eight bytes long to support a 64-bit address space. The PE32+ binary file format provides an opportunity to add sections of arbitrary code and modify the PE header to provide execution control to the additional code.

As mentioned earlier, it is possible, by means of information stored in the set of EFI system variables, to control the boot sequence and the path variable that determines the location of images that have to be loaded as OS kernel binaries.

The global variables can be modified using the functionality exported by the Run Time services. It is also possible to register a Run Time EFI driver that will be active in memory after the boot-time sequence when the OS is fully operational, as well as modify or even substitute the OS kernel image. This can be achieved by one of the EFI applications or the modified OS loader. Furthermore, it is possible by invoking the PXE mechanism to boot an alternative kernel from an arbitrary remote system.

The EFI Byte Code (EBC) virtual machine, which is accessible as a part of Boot services, provides a means of writing EFI applications using byte code instructions. Much like Java's implementation, the EBC protocol is handled by the internal interpreter and allows multi-platform portability, as well as a significant reduction in the size of EFI applications and EFI drivers.

Does the Door have Good Locks?

Even though the EFI provides a useful set of functions that could be used in security attacks, making them work is not an easy task. First, this would require in-depth knowledge of the EFI specification, which is still in its draft form, and secondly there is a provision in the EFI specification for the use of Authentication Protocol Interface (API).

Due to the inherent extensibility of the EFI, authorization and authentication services can be added to the working EFI implementation by utilizing the well-defined API. The API makes it possible to provide a mechanism of key and ID pair management as well as a validation of the pairs against previously created and stored credentials.

The authentication mechanism enables the user to establish a chain of trust for EFI applications. In other words, applications which share a particular security policy are capable of running only those applications that participate in the same security policy. Subsequently, the range of actions a hostile application is able to undertake may be restricted once engaged under an environment supervised by the security policy.

Furthermore, there are mechanisms for checking the integrity of PXE boot images provided by the Boot Integrity Services (BIS) APIs. Adding BIS to the EFI implementation minimizes the opportunity for unauthorized modification of boot service images.

All the protocols and functionalities necessary for making an EFI implementation more robust and secure are already in the EFI specification. It is up to software developers to start using them in future EFI implementations. And let's not forget that, since the EFI system partition is visible to the hosting OS in most circumstances, should a malicious code happen to affect any EFI applications stored on that partition, a good old friend, the anti-virus scanner, should be able to clean and keep the infections away.

ERRATUM

Windows XP Comparative Review: McAfee VirusScan

Unfortunately an error occurred in *Virus Bulletin*'s *Windows XP* comparative review (see *VB* June 2002, p.21): the results for *Network Associates' McAfee VirusScan* were replaced by those for *NAI VirusScan*. The correct results for *McAfee VirusScan* are reproduced in the table below.

The samples missed by *VirusScan* were mainly in the polymorphic set, where the offending items were Sepultura, W32/CTX and W32/Fosforo. The .TMP file dropped by W32/Nimda.A was undetected both in the *XP* review and in this month's *NetWare* tests. The file is included in the standard set as something of a curiosity file since, although it contains Nimda's code and is dropped by Nimda, this file is not a threat under any normal circumstances.

The results reported in the review for clean set scanning and false positives were correct. In light of the fact that no false positives were encountered and all In the Wild scans resulted in full detection, *McAfee VirusScan* is rightfully awarded a VB 100 % award for its performance. *VB* offers its apologies to *Network Associates* and to readers for the confusion.

McAfee VirusScan		On Demand	On Access
ITW File	number missed	0	0
	% detection	100.00%	100.00%
ITW Boot	number missed	0	0
	% detection	100.00%	100.00%
ITW Overall	number missed	0	0
	% detection	100.00%	100.00%
Macro Virus	number missed	0	0
	% detection	100.00%	100.00%
Polymorphic	number missed	8	8
	% detection	99.86%	99.86%
Standard	number missed	1	2
	% detection	99.98%	99.87%

COMPARATIVE REVIEW

NetWare and Tear

Matt Ham

The annual *NetWare* comparative has arrived once more and, as is usually the case, a new version of *NetWare* is in order; this year *NetWare 6* replaces *NetWare 5*.

The GUI that was introduced in *NetWare 5* has been retained in *NetWare 6*, although this is of limited relevance since the majority of products in this review are console-based. The minimal need to use the interface came as something of a relief, since *Novell*'s style gurus have opted for an interface which depicts a number of people in irritatingly unnatural poses who seem to have been attached to *Novell*'s trademark red 'N' by cut-and-paste jobs of varying degrees of competence.

Platform Scares

It seems that the choice of *NetWare 6* as a test platform scared off some vendors, who did not feel that their products had been adequately tested on the operating system to allow them to be subjected to the full *VB* testing process.

Special mention on this front goes to *Symantec*'s *Norton AntiVirus*. Originally this was submitted for testing in its 7.60 Corporate Edition version. However, it soon became apparent that there were some problems with the product's on-access scanning.

A discussion with *Symantec*'s engineers revealed that the product had been submitted under the misunderstanding that the test would take place on *NetWare 5*. Since the 7.60 version of *NAV* is not designed for *NetWare 6*, the product was withdrawn from the review. Unfortunately, version 8 of *NAV*, which *is* designed for *NetWare 6*, is not yet commercially available and so could not be included in the test.

Test Sets

Changes in the test sets for this comparative included the addition of W32/Simile (aka W32/Etap) in order to bolster the ranks of the polymorphic set. Since polymorphics and extensions were the root of some problematic issues in the previous two *NetWare* reviews, these were of particular interest on this occasion.

The last *NetWare* comparative review (see *VB*, September 2001, p.17) predicted that this year's review would prove to be much the same as ever – in that improvement would be seen in the general behaviour of the products, but that idiosyncrasies would remain to torment the unlucky user (and cause them to damn *Novell* and its assembled developers unto the seventh generation).

Since the proof of this metaphorical pudding is in the eating, it is now time to tuck into the offerings on the table, and judge them as sweet, savoury or downright sickening.

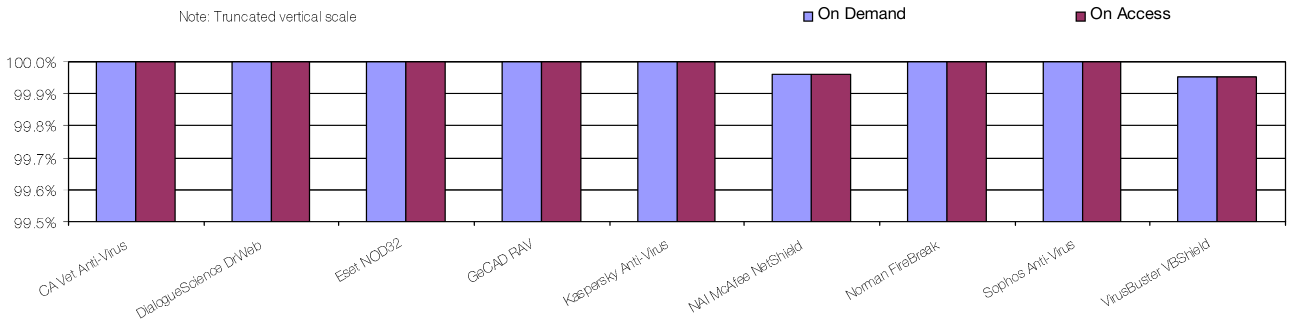
Test Environment

The test equipment has changed considerably since the last *NetWare* review in terms of both hardware and software. The configuration chosen was a *NetWare* server with an *NT* client. (In the last comparative several products demonstrated an incompatibility with a *Windows 98* client and as a result *NT* or, more likely, *XP* client is likely to be used in future reviews.)

While on-demand scans were selected to be performed entirely on the server where possible, control of this scanning was initiated by client-side utilities in cases where these were provided. Wherever possible, results are obtained by the parsing of log files – only one product in this review required different treatment.

On-access scanning was tested using file access from the client to files located on the server. This access was

In the Wild File Detection Rates



On-demand tests	ItW File		Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%
CA Vet Anti-Virus	0	100.00%	16	99.71%	13	99.31%	1	99.94%
DialogueScience DrWeb	0	100.00%	34	99.20%	1	99.96%	1	99.98%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%
GeCAD RAV	0	100.00%	0	100.00%	78	95.29%	6	99.67%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	2	99.84%	0	100.00%
NAI McAfee NetShield	1	99.96%	3	99.97%	1	99.92%	2	99.88%
Norman FireBreak	0	100.00%	0	100.00%	149	91.25%	15	99.32%
Sophos Anti-Virus	0	100.00%	9	99.77%	93	93.31%	17	99.43%
VirusBuster VBShield	1	99.95%	0	100.00%	658	86.87%	11	99.56%

triggered by a custom utility which performs file opens on every file in the virus test sets. Products were logged as able to detect a virus on access if, when configured to do so on viral detection, the files were blocked from being accessed.

Logging for on-access scanners is still less well implemented than for on-demand scanners and thus this method has been chosen as being more universally applicable to the products on test. Again, there was one product that could not be tested in this way, instead detection was judged by deletion of infected files.

Try, Try and Try Again

Where results were unobtainable due to software failure or displays of particularly strange behaviour of the software, the testing procedure was repeated up to three times so as to determine whether the defect was reproducible or simply a one-off glitch.

Despite the fact that the images used for these new installations are identical in every way, this process of repetition will often change the results obtained. Products which remain untestable after three retries are noted as such. Although, in the past, more than three attempts have been required to coerce a product into correct operation, this cut-off point has been introduced due to the time constraints imposed by publication deadlines.

The server operating system was *NetWare 6* with service pack 1 installed, linked by a 100 Mbit ethernet connection to an *NT 4 SP 6* workstation. The client software used on the workstation was *Novell Client 4.83*. Both the workstation and the server were fully re-imaged between changes of product, ensuring that each product had an identical configuration for installation. A further *Windows XP Professional* workstation was attached to the server for use in storing results data. Hardware specifications are provided at the end of the review.

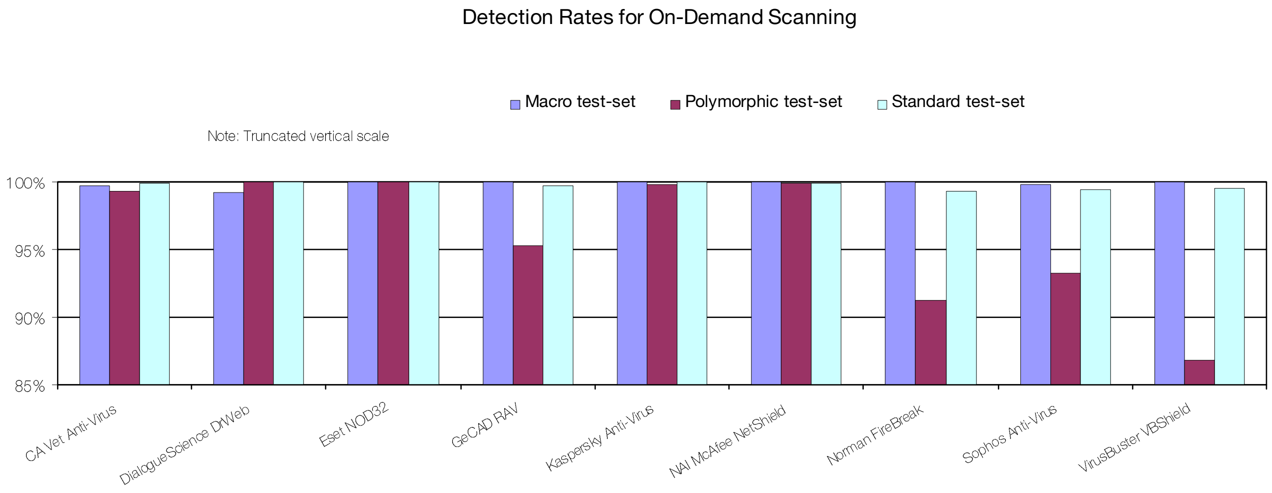
The method of control varied considerably between the products reviewed, although the majority were controlled directly through the NLM on the server. This method of control should be assumed throughout the review unless otherwise stated. Where required, NWAdmin version 5.1.9f was installed for administrative purposes.

False Positives and Archives

Testing of false positives was performed on the usual *Virus Bulletin* clean set, consisting of 5500 clean executables and a selection of OLE files embedded with varying numbers of macros and other OLE streams.

For the testing of archive handling, subsets of the aforementioned test libraries were used, zipped into multiple archives with one level of compression applied. Figures for scanning throughput on the archived file sets are given for the uncompressed content size of the archive.

In products which are speed-limited by disk access times, throughput may be higher on archived files than on the same files when unarchived. This is due to the fact that the time taken to read an archive plus perform calculations to decompress the archives in memory can be faster than reading a much larger file from the hard drive.



Computer Associates Vet Anti-Virus 10.4.9 v 2160

ItW File	100.00%	Macro	99.71%
ItW File (o/a)	100.00%	Macro (o/a)	99.71%
Standard	99.94%	Polymorphic	99.31%

Vet is usually among the first products to be described in the writeup of any comparative, and on this occasion it was also the first product to undergo the testing process. The first test always sets the tone for a review, since although certain products may be uniformly easy or difficult to review, the operating system in use can be gauged fairly quickly for quirks and oddities. As mentioned above, this was a pleasant experience with *NetWare*, allowing the products themselves to claim the rightful centre of attention.

Installation of *Vet* was straightforward, and updating was a simple matter of copying across new files into the installation directory.

Leaving aside the mention of Aardvarks in the manual, *Vet for NetWare* has no major distinguishing features, its interface being a single central NLM with a classic *NetWare* look. Irritatingly, the status of a scan cannot be viewed from this interface – the only information available is the fact that the scan is in progress. Since the log files are locked during scanning this leaves an air of mystery surrounding any scan. This obfuscation also applied to some of the options within the program where, for example, the default state of archive scanning could be discovered only by scanning.

Despite these complaints, *Vet*'s performance was good – scans were fast and false-positive-free on the clean set and no misses of virus samples In the Wild gains the product a VB 100% award. Where weaknesses did occur in detection they were isolated rather than general – with the polymorphic viruses in both polymorphic and macro test sets containing some files which presented difficulties.

DialogueScience DrWeb 4.28

ItW File	100.00%	Macro	99.20%
ItW File (o/a)	100.00%	Macro (o/a)	99.20%
Standard	99.98%	Polymorphic	99.96%

Also sporting a classic *NetWare* look, *DrWeb* emphasises its retro style by using a green colour scheme for the interface. The most unusual feature of the product is its total lack of an on-demand scanner. This is not the fatal flaw that might be anticipated, since scheduled scans may be used as a replacement for this functionality. However, the process of on-demand scanning is rendered somewhat clumsy by this design. The scheduled and on-access scanning portions of the program are both controlled from a single NLM.

Scanning of the clean test sets was at the faster end of the spectrum, with the usual 16 suspicious files being produced. With full detection of files In the Wild, *DrWeb* earns the second VB 100% award of this comparative. The newer polymorphics were a particularly strong area for *DrWeb*, with only one sample missed in this category. Slightly more surprising was a weakness in older Excel macro viruses.

Eset NOD32 1.280 20020708

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

When discussing *NOD32* in the past, faults have been few and far between, but on this occasion the matter was somewhat different. The normally delightful *NOD32* log file has, in some bizarre fashion, been converted to a festering mass of corruption designed to attract dire imprecations.

First, the file names in the log were changed to 8+3 format, making it extremely difficult in some cases to determine

On-access tests	ItW File		Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%
CA Vet Anti-Virus	0	100.00%	16	99.71%	13	99.31%	3	99.81%
DialogueScience DrWeb	0	100.00%	34	99.20%	1	99.96%	1	99.98%
Eset NOD32	0	100.00%	0	100.00%	0	100.00%	0	100.00%
GeCAD RAV	0	100.00%	0	100.00%	78	95.29%	8	99.55%
Kaspersky Anti-Virus	0	100.00%	0	100.00%	2	99.84%	2	99.87%
NAI McAfee NetShield	1	99.96%	3	99.97%	1	99.92%	4	99.76%
Norman FireBreak	0	100.00%	0	100.00%	150	91.25%	15	99.32%
Sophos Anti-Virus	0	100.00%	13	99.67%	93	93.31%	18	99.41%
VirusBuster VBShield	1	99.95%	0	100.00%	664	86.74%	13	99.44%

exactly which files had been missed. As if that cardinal sin were not enough, the path delimiting ‘\’ symbols were all converted to ‘/’ and all file names converted to lower case. While the changing of path delimiters may be excusable for some arcane *NetWare*-specific reason, it seems pointless to change file names in two respects when referring to those files in a log.

Returning to the product, *NOD32* comes as two NLMs – amon and nod32 – handling on-access and on-demand scanning respectively. Installation and update were both simple matters of copying the files to the correct location. The nod32 NLM is loaded and unloaded each time an on-demand scan is initiated and, as such, does not support scheduled scans directly.

As far as detection and scan speeds were concerned, *NOD32* retained its impressive performance history, detecting all files in all test sets. This, combined with no false positive detections, gains *NOD32* yet another VB 100% award. It is to be hoped that the new-found log file problems remain less of an ongoing feature than the product’s impressively high detection rates.

GeCAD RAV AntiVirus v.8 1.07

ItW File	100.00%	Macro	100.00%
ItW File (o/a)	100.00%	Macro (o/a)	100.00%
Standard	99.67%	Polymorphic	95.29%

RAV is the first of the products described so far to have a *Windows*-based installer for its product. An automatic update function is supported, though for full automation it seems that the *Windows* product must also be installed. The product itself is split into separate components which are loaded as different NLMs for each function.

The scan of the clean sets was notably slower on the executable files than the OLE files in the test set, and resulted in one false positive. The rate of scanning on clean files was also significantly slower than that on infected files – which would suggest that *RAV* is using quite a large quantity of heuristics.

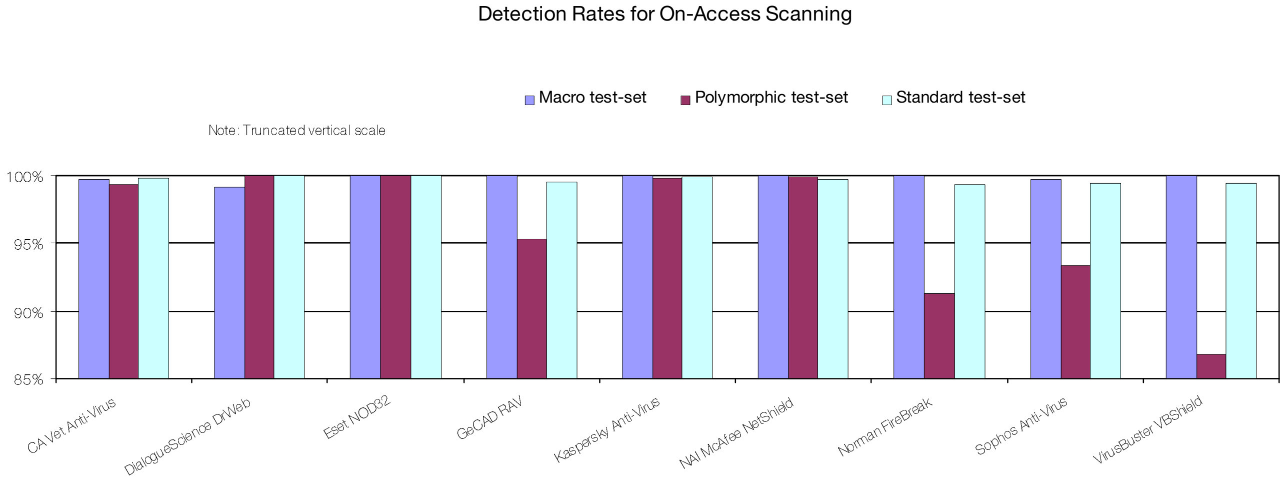
The single false positive will be irritating for *GeCAD*, since the detection statistics for *RAV* were good. Misses did occur on the polymorphics in both the polymorphic and standard test sets, but samples in the macro and ItW sets were fully detected.

The two main misses were the newer polymorphics of W32/Etap and W32/Zmist.D. This pair is rapidly assuming the mantle long held by the ACG and SP variants in the category of ‘difficult-to-detect’ polymorphics.

The matter of log files reared its ugly head again when analysing *RAV*’s results, the path names having been converted to 8 + 3 format in the log.

Kaspersky Anti-Virus 4.00.01

ItW File	100.00%	Macro	100.00%
ItW File (o/a)	100.00%	Macro (o/a)	99.97%
Standard	99.09%	Polymorphic	98.10%



Kaspersky Anti-Virus is the first of those products tested which does not rely on being controlled directly through the NLM or a command line interface. During installation it installs snapins for both NWAdmin and ConsoleOne and requires that all administration be performed through these.

In this case, NWAdmin was used for control of scans. For this method of administration there are both pros and cons. On the negative side, there is the need for communication between the client and server during scans, which might be expected to lead to slower scan speeds. In practice, however, the scans were not noticeably slower than those performed by other products, so this is a niggle of minor concern. On the more positive side, the use of a real GUI rather than a *NetWare*-style console interface makes both administration and scans substantially easier to perform.

Scanning performance was flawless in the In the Wild and macro test sets which, combined with a lack of false positives, results in a VB 100% award for *Kaspersky* after a considerable drought. There were misses in the standard and polymorphic sets, which were, oddly enough, confined to samples whose file names begin with the letter N.

This odd behaviour was apparent in both on-access and on-demand tests, but further examination of the results showed that the phenomenon was not exhibited on the same files in the two. Reinstallation of the product and repeats of the tests could not reproduce this odd behaviour, which thus enters the 'unexplained mysteries' file. The misses following the subsequent tests left *KAV* with very close to full detection in all test sets.

NAI McAfee NetShield 4.60 4.160 4.0.4210

ItW File	99.96%	Macro	99.97%
ItW File (o/a)	99.96%	Macro (o/a)	100.00%
Standard	99.88%	Polymorphic	99.92%

NetShield is another product which uses a client-based interface in order to implement changes on the server-based portion of the product. In this case the *NetShield* console is a *Windows*-style application on the client, which attempts to contact the server-based portion of the software whenever it is run and requires a login and server selection on every execution. This requires slightly more rigmarole than the *Kaspersky* control method described above, and requires that the Java runtime environment be present on the client machine before the *NetWare* portion of the product can be installed.

With Java's future on *Microsoft* platforms being uncertain, it remains to be seen what changes will be made to *NAI*'s reliance on the runtime environment in future releases. On a positive note, users familiar with any other *NAI* product will find that the interface here is so similar to that found in others from the same manufacturer that there will be no difficulty in using the *NetWare* software.

The scanning speeds exhibited by *NetShield* were at the slower end of the table, though it was difficult to tell how much of this was due to trans-network interaction since scan speed is often relatively slow for *NAI* products.

Unfortunately *NAI*'s *NetShield* does not become the fifth product to receive a VB100 in this review. Despite having laid to rest the ghost of extension-based misses on most of their platforms, the *NetWare* product failed to detect any of those samples which were extensionless, including one, O97M/Tristate.C, In the Wild. With detection rates elsewhere being close to perfect and no false positives, the misses of these samples may leave a particularly nasty taste in *NAI*'s corporate maw.

Norman FireBreak 4.10.2047 5.00.42

ItW File	100.00%	Macro	100.00%
ItW File (o/a)	100.00%	Macro (o/a)	100.00%
Standard	99.32%	Polymorphic	91.25%

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files	
	Time (s)	Throughput (MB/s)	FPs [susp]	Time(s)	Throughput (MB/s)	FPs [susp]	Time (s)	Throughput (MB/s)	Time(s)	Throughput (MB/s)
CA Vet Anti-Virus	140	3906.7		11	7212.2		86	1853.7	11	6782.5
DialogueScience DrWeb	165	3314.7	[16]	13	6102.6		73	2183.8	13	5739.0
Eset NOD32	65	8414.3		7	11333.4		22	7246.2	4	18651.9
GeCAD RAV	565	968.0		9	8814.9		85	6434.5	11	7212.2
Kaspersky Anti-Virus	230	2378.0		18	4407.4		136	1172.2	32	2331.5
NAI McAfee NetShield	450	1215.4		27	2938.3		165	966.2	37	2016.4
Norman FireBreak	2040	268.1		10	7933.4		20	7970.8	4	18651.9
Sophos Anti-Virus	146	3746.1		20	3966.7		44	3623.1	10	7460.7
VirusBuster VBShield	279	1960.3	1	98	809.5		133	1198.6	40	1865.2

Norman's *FireBreak* returns to the NWAdmin method of control, though it also offers direct control over the single NLM-based server portion. This proved fortuitous because the NWAdmin portion of the application refused to function properly. The method of control used, therefore, was that of interaction directly with the NLM interface. Control on the server was hindered somewhat by the less than intuitive choice of selection keys (for example F5 to select an object for scanning), which are not mentioned on-screen. The readme files do contain this information, though it is buried sufficiently deeply that a casual reader will be very lucky to spot it.

The primary problem for *FireBreak* came with the scanning of the executable clean set. On these files the scanning rate slowed to a snail's pace, becoming increasingly languorous as the test continued. In the past, slow scanning speeds for *Norman* products have been a result of delaying the scan engine deliberately so as not to overload the server, though on this occasion server load reached 100% for considerable lengths of time. However, the other scan speeds were very good and no false positives were detected.

With full detection rates in the ItW and macro test sets, *Norman FireBreak* qualifies for another VB 100%. Weaknesses in detection were, fairly predictably, centred around the newer polymorphics, W32/Etap, W32/Zmist.D and W32/Fosforo. On a slightly more negative note, in log file parsing it was noted that some portions of the path had had their case converted when displayed in the log file, in addition to alteration of '\' to '/' in path descriptions.

Sophos Anti-Virus remains unique in its method of installation, consisting of only a single NLM. When executed this acts in much the same way as a self-extracting executable, creating directories and the files to fill them.

Updates are managed automatically by placing further releases of the NLM into a specified directory, from where the components are extracted. All the functions of the product are controlled through one main NLM installed in this process.

Traditionally, *Sophos* products have been set up with the scanning of compressed files turned off by default, so it came as a surprise to note that the opposite was true in this product. This brings *SAV* in line with most other products in this review, though sadly it also shares with most of those products the lack of a means to browse to targets. Another feature in common with several other products in the review is *SAV*'s habit of mangling log file entries – in this case the crimes were addition of entries for some worms, conversion to 8+3 format and conversion of '\' to '/'.

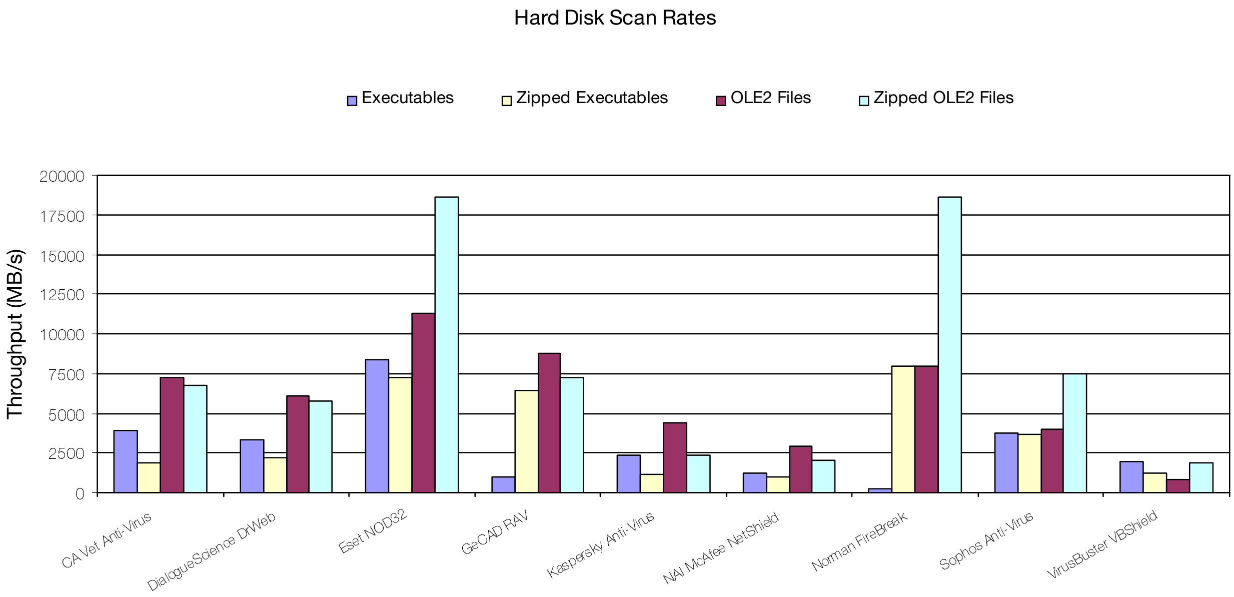
Despite these complaints (which are, by and large, directable towards the majority of the products on offer), *Sophos AntiVirus* performed speedily and with good detection rates. As usual, the samples in the test set that are potentially slow to scan were undetected by choice. This includes the various Access viruses present in the set, mid-infectors such as Positron and DLL-based threats such as Navrhar. Since none of these reside in the ItW set, however, *Sophos Anti-Virus* earns another VB 100 % award.

Sophos Anti-Virus 3.59

ItW File	100.00%	Macro	99.77%
ItW File (o/a)	100.00%	Macro (o/a)	99.67%
Standard	99.43%	Polymorphic	93.31%

VirusBuster VBShield v 1.14.000 7.456

ItW File	99.95%	Macro	100.00%
ItW File (o/a)	99.95%	Macro (o/a)	100.00%
Standard	99.56%	Polymorphic	86.87%



In the previous two *NetWare* reviews, *VBShield* was notable for the fact that its on-demand log files were unusable. It seems that some things never change since this was the case once again, making it necessary for results to be gained by deletion of infected files. Other products featured unusable log files on access, but *VirusBuster* was the only product to do so on demand. Since the problem is simply that the log file splits reports for one file arbitrarily over more than one line if they are over a certain number of characters, this would seem to be an easy and worthwhile fix to implement.

In the previous *NetWare* review, *VirusBuster*'s product suffered the majority of its problems with the polymorphic viruses. This was the case again. Almost all misses for *VBShield* were in the polymorphic test sets, with one of the polymorphic W32/CTX samples being missed in the ItW test set. This was sufficient to deny *VBShield* a VB 100 % award. There were a large number of misses not only amongst the newer but also amonst some of the older polymorphic files. Happily, the comment made in the last review that a significant improvement in detection rates had been seen in *VirusBuster*'s products over the preceding year, can be repeated, although this may make the narrow miss of a VB 100 % all the more disappointing for *VirusBuster*'s developers.

Conclusions

The review finishes on a product for which the comments made in last year's review still ring true, but what is surprising is that the rest of the products reviewed show fewer similarities with their previous incarnations and that my general dislike of *NetWare* has been somewhat mollified over the course of this latest comparative.

In general, the detection rates and ease of use of the products have improved rather more than I dared to hope at the end of the last *NetWare* review. With poorly chosen extension listings for *NAI*, one false positive for *GeCAD* and one missed sample for *VirusBuster* being the three factors preventing a clean sweep of VB 100% awards, this is among the more impressive comparative reviews in terms of product performance. This is deserving of congratulations to all concerned – though tempered with the knowledge that some of the results were let down by such small failings.

NetWare 6 is clearly *Novell*'s customer product of choice at the moment. It is somewhat disturbing that so many companies do not yet have enough confidence in their products on *NetWare 6* to submit them for testing – or have no current product that is usable on *NetWare 6*.

That the market for *NetWare* has suffered considerably during the last half-decade is undeniable, yet the installed user base remains as a market. One feels that, while some companies are active in their development of new features and management tools on *NetWare*, a number of others consider it to be an unpleasant chore to update.

For my prediction I will state boldly that this will not be the year of the *NetWare* virus. With the anti-virus developers reluctant to support *NetWare* when being paid for their expertise, what hope for inspiring virus writers to produce malware for such an operating system? With this thought in mind, *NetWare* looks more appetizing at every turn.

Technical Details

Test environment: Server: 1.6 GHz Intel Pentium 4 workstation with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy, running *NetWare 6 Service Pack 1*.

Workstation: 1.6 GHz Intel Pentium 4 workstation with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy, l running *Windows NT 4 Service Pack 6*.

Network: 100 Mbit ethernet.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/NetWare/2002/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 9731266, Fax (781) 9731267

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

END NOTES AND NEWS

The Information Systems Audit and Control Association's **Network Security Conference takes place 12–14 August 2002 in Las Vegas, USA and 18–20 November 2002 in Munich, Germany**. For more information see http://www.isaca.org/.

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

The Fourth Annual NTBugtraq Retreat will be held at NTBugtraq Headquarters in Lindsay, Ontario, Canada, 20–23 August 2002. The event will consist of three days of discussions focused around *NT*/*W2K*/*XP* and security issues. The event is designed to encourage interaction between participants to leverage knowledge gained, share concerns and common questions, and help form consensus on how to approach securing *Windows NT*/*2000*/*XP*. Registration is restricted to 50 people. See http://ntbugtraq.ntadvice.com/conference.asp.

The 9th International Computer Security Symposium, COSAC 2002, takes place 8–12 September 2002 at Killashee Hotel, County Kildare, Ireland. Cost of registration includes your choice of 40 symposium sessions, five full-day master classes, and the COSAC International Peer Group meeting, in addition to full-board accommodation and meals. Register at http://www.cosac.net/.

The 12th International Virus Bulletin Conference takes place at the Hyatt Regency, New Orleans, LA, USA from 26–27 September 2002. Register now and take advantage of special *VB* subscriber rates. Contact us for more information: tel +44 1235 555139, or email VB2002@virusbtn.com. See the *VB* website for full conference programme details: http://www.virusbtn.com/.

Black Hat Asia 2002 takes place at the Marina Mandarin Hotel, Singapore, 1–4 October 2002. Five training courses take place 1–2 October, with two tracks of presentations at the Briefings, 3–4 October. For further information see http://www.blackhat.com/.

Information Security Systems Europe 2002 will be held in Disneyland, Paris, from 2–4 October 2002. Presentations cover technology, infrastructure, applications, legal/political issues and threats and responses. For more details see http://www.isse.org/.

The Third Annual RSA Conference 2002, Europe is to take place 7–10 October 2002 at Le Palais des Congrès de Paris, France. As well as keynote presentations there will be more than 85 individual breakout sessions on topics ranging from enterprise security to hacking and intrusion forensics. See http://www.rsaconference.com/.

COMPSEC 2002 takes place on 30 October and 1 November 2002 in London, UK. Presentations and interactive workshops are arranged within four streams, covering management concerns, infrastructure, law and ethics, technical issues and case studies. Register by 15 July for reduced rates. See http://www.compsec2002.com/.

The CSI 29th Annual Computer Security Conference and Exhibition will be held 11–13 November 2002 in Chicago, IL, USA. The conference is aimed at anyone with responsibility for or interest in information and network security. For more information email csi@cmp.com or see http://www.gocsi.com/.

The 5th Anti-Virus Asia Researchers (AVAR) Conference takes place 21-22 November 2002 in Seoul, Korea. Topics covered will include information on how the AV community works together globally, the latest virus and AV technologies, and reports on virus prevalence in various countries in Asia. The conference will be hosted by *Ahnlab, Inc.* For more information see http://www.aavar.org/.

Infosecurity 2002 conference and exhibition will be held 10–12 December 2002 at the Jacob K. Javits Center, New York, USA. For further details, including information on exhibiting and conference registration, see http://www.infosecurityevent.com/.

A call for papers has been issued for the RSA 2003 conference. Submissions must be received by 16 September 2002. Details of how to submit proposals can be found at http://www.rsaconference.net/.

NetIQ Corporation* and *Sybari Software, Inc.* have announced a new performance and availability management module for *Antigen. The *NetIQ* module provides centralized management and diagnostics of *Antigen* technology through automated problem detection and correction. See http://www.sybari.com/.

AV-Test.org has published the results of detection and disinfection tests on *Windows ME* and *XP*. See http://www.av-test.org/.